

# Rotation,Nutation,andPrecessionModel

## 5.0

Generated by Doxygen 1.8.5

Wed Jun 1 2022 12:07:54



# 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>11</b>
6.1	Models . . . . .	11
6.1.1	Detailed Description . . . . .	11
6.2	Environment . . . . .	12
6.2.1	Detailed Description . . . . .	12
6.3	RNP . . . . .	13
6.3.1	Detailed Description . . . . .	13
6.4	GenericRNP . . . . .	14
6.4.1	Detailed Description . . . . .	14
6.4.2	Macro Definition Documentation . . . . .	14
6.4.2.1	PATH . . . . .	14
6.5	RNPJ2000 . . . . .	15
6.5.1	Detailed Description . . . . .	15
6.6	RNPMars . . . . .	16
6.6.1	Detailed Description . . . . .	16
<b>7</b>	<b>Namespace Documentation</b>	<b>17</b>
7.1	jeod Namespace Reference . . . . .	17
7.1.1	Detailed Description . . . . .	18

<b>8</b>	<b>Data Structure Documentation</b>	<b>19</b>
8.1	jeod::NutationJ2000 Class Reference	19
8.1.1	Detailed Description	20
8.1.2	Constructor & Destructor Documentation	21
8.1.2.1	NutationJ2000	21
8.1.2.2	~NutationJ2000	21
8.1.2.3	NutationJ2000	21
8.1.3	Member Function Documentation	21
8.1.3.1	initialize	21
8.1.3.2	operator=	21
8.1.3.3	update_rotation	21
8.1.4	Friends And Related Function Documentation	21
8.1.4.1	init_attrjeod__NutationJ2000	21
8.1.4.2	InputProcessor	22
8.1.5	Field Documentation	22
8.1.5.1	D	22
8.1.5.2	D_coeffs	22
8.1.5.3	epsilon_bar	22
8.1.5.4	equa_of_equi	22
8.1.5.5	F	22
8.1.5.6	F_coeffs	22
8.1.5.7	L	23
8.1.5.8	L_coeffs	23
8.1.5.9	long_coeffs	23
8.1.5.10	long_t_coeffs	23
8.1.5.11	M	23
8.1.5.12	M_coeffs	23
8.1.5.13	num_coeffs	24
8.1.5.14	nutaton_in_longitude	24
8.1.5.15	nutaton_in_obliquity	24
8.1.5.16	obliq_coeffs	24
8.1.5.17	obliq_t_coeffs	24
8.1.5.18	omega	24
8.1.5.19	omega_coeffs	24
8.2	jeod::NutationJ2000Init Class Reference	25
8.2.1	Detailed Description	26
8.2.2	Constructor & Destructor Documentation	26
8.2.2.1	NutationJ2000Init	26
8.2.2.2	~NutationJ2000Init	26
8.2.2.3	NutationJ2000Init	26

8.2.3	Member Function Documentation	26
8.2.3.1	operator=	26
8.2.4	Friends And Related Function Documentation	26
8.2.4.1	init_attrjeod__NutationJ2000Init	26
8.2.4.2	InputProcessor	26
8.2.5	Field Documentation	26
8.2.5.1	D_coeffs	26
8.2.5.2	F_coeffs	27
8.2.5.3	L_coeffs	27
8.2.5.4	long_coeffs	27
8.2.5.5	long_t_coeffs	27
8.2.5.6	M_coeffs	27
8.2.5.7	num_coeffs	28
8.2.5.8	obliq_coeffs	28
8.2.5.9	obliq_t_coeffs	28
8.2.5.10	omega_coeffs	28
8.3	jeod::NutationJ2000Init_nutation_j2000_default_data Class Reference	28
8.3.1	Detailed Description	28
8.3.2	Member Function Documentation	29
8.3.2.1	initialize	29
8.4	jeod::NutationMars Class Reference	29
8.4.1	Detailed Description	30
8.4.2	Constructor & Destructor Documentation	30
8.4.2.1	NutationMars	30
8.4.2.2	~NutationMars	30
8.4.2.3	NutationMars	30
8.4.3	Member Function Documentation	30
8.4.3.1	operator=	30
8.4.3.2	update_rotation	31
8.4.4	Friends And Related Function Documentation	31
8.4.4.1	init_attrjeod__NutationMars	31
8.4.4.2	InputProcessor	31
8.4.5	Field Documentation	31
8.4.5.1	I_at_j2000	31
8.4.5.2	I_dot	31
8.4.5.3	I_m_orig	31
8.4.5.4	int_to_double	31
8.4.5.5	mean_anomaly_j2000	32
8.4.5.6	mean_motion	32
8.4.5.7	nutration_in_longitude	32

8.4.5.8	<a href="#">nutation_in_obliquity</a>	32
8.4.5.9	<a href="#">obliquity_angle</a>	32
8.4.5.10	<a href="#">psi_m_orig</a>	32
8.4.5.11	<a href="#">q_angle_j2000</a>	33
8.5	<a href="#">jeod::PlanetOrientation Class Reference</a>	33
8.5.1	<a href="#">Detailed Description</a>	34
8.5.2	<a href="#">Constructor &amp; Destructor Documentation</a>	34
8.5.2.1	<a href="#">PlanetOrientation</a>	34
8.5.2.2	<a href="#">~PlanetOrientation</a>	34
8.5.2.3	<a href="#">PlanetOrientation</a>	35
8.5.3	<a href="#">Member Function Documentation</a>	35
8.5.3.1	<a href="#">activate</a>	35
8.5.3.2	<a href="#">deactivate</a>	35
8.5.3.3	<a href="#">ephem_activate</a>	35
8.5.3.4	<a href="#">ephem_build_tree</a>	35
8.5.3.5	<a href="#">ephem_initialize</a>	35
8.5.3.6	<a href="#">get_name</a>	36
8.5.3.7	<a href="#">initialize</a>	36
8.5.3.8	<a href="#">operator=</a>	36
8.5.3.9	<a href="#">set_name</a>	36
8.5.4	<a href="#">Friends And Related Function Documentation</a>	36
8.5.4.1	<a href="#">init_attrjeod__PlanetOrientation</a>	36
8.5.4.2	<a href="#">InputProcessor</a>	36
8.5.5	<a href="#">Field Documentation</a>	36
8.5.5.1	<a href="#">active</a>	36
8.5.5.2	<a href="#">name</a>	37
8.5.5.3	<a href="#">orient_interface</a>	37
8.5.5.4	<a href="#">planet</a>	37
8.5.5.5	<a href="#">planet_omega</a>	37
8.5.5.6	<a href="#">planet_rot_state</a>	37
8.6	<a href="#">jeod::PlanetRNP Class Reference</a>	38
8.6.1	<a href="#">Detailed Description</a>	39
8.6.2	<a href="#">Member Enumeration Documentation</a>	39
8.6.2.1	<a href="#">RNPFidelity</a>	39
8.6.3	<a href="#">Constructor &amp; Destructor Documentation</a>	39
8.6.3.1	<a href="#">PlanetRNP</a>	39
8.6.3.2	<a href="#">~PlanetRNP</a>	40
8.6.3.3	<a href="#">PlanetRNP</a>	40
8.6.4	<a href="#">Member Function Documentation</a>	40
8.6.4.1	<a href="#">get_name</a>	40

8.6.4.2	<code>operator=</code>	40
8.6.4.3	<code>propagate_rnp</code>	40
8.6.4.4	<code>update_axial_rotation</code>	40
8.6.4.5	<code>update_rnp</code>	40
8.6.5	Friends And Related Function Documentation	41
8.6.5.1	<code>init_attrjeod__PlanetRNP</code>	41
8.6.5.2	<code>InputProcessor</code>	41
8.6.6	Field Documentation	41
8.6.6.1	<code>enable_polar</code>	41
8.6.6.2	<code>NP_matrix</code>	41
8.6.6.3	<code>nutaton</code>	41
8.6.6.4	<code>polar_motion</code>	41
8.6.6.5	<code>precession</code>	42
8.6.6.6	<code>rnp_type</code>	42
8.6.6.7	<code>rotation</code>	42
8.6.6.8	<code>scratch_matrix</code>	42
8.7	<code>jeod::PlanetRotation</code> Class Reference	42
8.7.1	Detailed Description	44
8.7.2	Constructor & Destructor Documentation	44
8.7.2.1	<code>PlanetRotation</code>	44
8.7.2.2	<code>~PlanetRotation</code>	44
8.7.2.3	<code>PlanetRotation</code>	44
8.7.3	Member Function Documentation	44
8.7.3.1	<code>get_rotation</code>	44
8.7.3.2	<code>get_rotation_transpose</code>	44
8.7.3.3	<code>initialize</code>	44
8.7.3.4	<code>operator=</code>	45
8.7.3.5	<code>update_rotation</code>	45
8.7.3.6	<code>update_time</code>	45
8.7.4	Friends And Related Function Documentation	45
8.7.4.1	<code>init_attrjeod__PlanetRotation</code>	45
8.7.4.2	<code>InputProcessor</code>	45
8.7.5	Field Documentation	45
8.7.5.1	<code>current_time</code>	45
8.7.5.2	<code>DAYTOJULIANCENT</code>	46
8.7.5.3	<code>DEGTORAD</code>	46
8.7.5.4	<code>DEGTOSEC</code>	46
8.7.5.5	<code>JULIANCENTTODAY</code>	46
8.7.5.6	<code>JULIANDAYTOSEC</code>	46
8.7.5.7	<code>RADTODEG</code>	46

8.7.5.8	rotation	47
8.7.5.9	SECTODEG	47
8.7.5.10	SECTOJULIANDAY	47
8.8	jeod::PlanetRotationInit Class Reference	47
8.8.1	Detailed Description	48
8.8.2	Constructor & Destructor Documentation	48
8.8.2.1	PlanetRotationInit	48
8.8.2.2	~PlanetRotationInit	48
8.8.2.3	PlanetRotationInit	48
8.8.3	Member Function Documentation	48
8.8.3.1	operator=	48
8.8.4	Friends And Related Function Documentation	48
8.8.4.1	init_attrjeod__PlanetRotationInit	48
8.8.4.2	InputProcessor	48
8.9	jeod::PolarMotionJ2000 Class Reference	48
8.9.1	Detailed Description	50
8.9.2	Constructor & Destructor Documentation	50
8.9.2.1	PolarMotionJ2000	50
8.9.2.2	~PolarMotionJ2000	50
8.9.2.3	PolarMotionJ2000	50
8.9.3	Member Function Documentation	50
8.9.3.1	initialize	50
8.9.3.2	operator=	50
8.9.3.3	update_rotation	50
8.9.4	Friends And Related Function Documentation	51
8.9.4.1	init_attrjeod__PolarMotionJ2000	51
8.9.4.2	InputProcessor	51
8.9.5	Field Documentation	51
8.9.5.1	last_table_index	51
8.9.5.2	override_table	51
8.9.5.3	polar_mjd	51
8.9.5.4	warn_table	51
8.9.5.5	xp	51
8.9.5.6	xp_tbl	52
8.9.5.7	yp	52
8.9.5.8	yp_tbl	52
8.10	jeod::PolarMotionJ2000Init Class Reference	52
8.10.1	Detailed Description	53
8.10.2	Constructor & Destructor Documentation	53
8.10.2.1	PolarMotionJ2000Init	53



8.10.2.2	<a href="#">~PolarMotionJ2000Init</a>	53
8.10.2.3	<a href="#">PolarMotionJ2000Init</a>	54
8.10.3	<a href="#">Member Function Documentation</a>	54
8.10.3.1	<a href="#">operator=</a>	54
8.10.4	<a href="#">Friends And Related Function Documentation</a>	54
8.10.4.1	<a href="#">init_attrjeod__PolarMotionJ2000Init</a>	54
8.10.4.2	<a href="#">InputProcessor</a>	54
8.10.5	<a href="#">Field Documentation</a>	54
8.10.5.1	<a href="#">last_table_index</a>	54
8.10.5.2	<a href="#">override_table</a>	54
8.10.5.3	<a href="#">polar_mjd</a>	54
8.10.5.4	<a href="#">xp</a>	54
8.10.5.5	<a href="#">xp_tbl</a>	55
8.10.5.6	<a href="#">yp</a>	55
8.10.5.7	<a href="#">yp_tbl</a>	55
8.11	<a href="#">jeod::PolarMotionJ2000Init_xpyp_daily_default_data Class Reference</a>	55
8.11.1	<a href="#">Detailed Description</a>	55
8.11.2	<a href="#">Member Function Documentation</a>	55
8.11.2.1	<a href="#">initialize</a>	55
8.12	<a href="#">jeod::PolarMotionJ2000Init_xpyp_monthly_default_data Class Reference</a>	56
8.12.1	<a href="#">Detailed Description</a>	56
8.12.2	<a href="#">Member Function Documentation</a>	56
8.12.2.1	<a href="#">initialize</a>	56
8.13	<a href="#">jeod::PrecessionJ2000 Class Reference</a>	56
8.13.1	<a href="#">Detailed Description</a>	57
8.13.2	<a href="#">Constructor &amp; Destructor Documentation</a>	57
8.13.2.1	<a href="#">PrecessionJ2000</a>	57
8.13.2.2	<a href="#">~PrecessionJ2000</a>	57
8.13.2.3	<a href="#">PrecessionJ2000</a>	57
8.13.3	<a href="#">Member Function Documentation</a>	57
8.13.3.1	<a href="#">operator=</a>	57
8.13.3.2	<a href="#">update_rotation</a>	57
8.13.4	<a href="#">Friends And Related Function Documentation</a>	58
8.13.4.1	<a href="#">init_attrjeod__PrecessionJ2000</a>	58
8.13.4.2	<a href="#">InputProcessor</a>	58
8.14	<a href="#">jeod::PrecessionMars Class Reference</a>	58
8.14.1	<a href="#">Detailed Description</a>	59
8.14.2	<a href="#">Constructor &amp; Destructor Documentation</a>	59
8.14.2.1	<a href="#">PrecessionMars</a>	59
8.14.2.2	<a href="#">~PrecessionMars</a>	59

8.14.2.3	PrecessionMars	59
8.14.3	Member Function Documentation	59
8.14.3.1	compute_fixed_matrices	59
8.14.3.2	operator=	60
8.14.3.3	update_rotation	60
8.14.4	Friends And Related Function Documentation	60
8.14.4.1	init_attrjeod__PrecessionMars	60
8.14.4.2	InputProcessor	60
8.14.5	Field Documentation	60
8.14.5.1	J	60
8.14.5.2	N	60
8.14.5.3	NJ_matrix	60
8.14.5.4	nutaton	60
8.14.5.5	psi_at_j2000	61
8.14.5.6	psi_dot	61
8.14.5.7	psi_precess	61
8.15	jeod::RNPJ2000 Class Reference	61
8.15.1	Detailed Description	63
8.15.2	Constructor & Destructor Documentation	63
8.15.2.1	RNPJ2000	63
8.15.2.2	~RNPJ2000	63
8.15.2.3	RNPJ2000	63
8.15.3	Member Function Documentation	63
8.15.3.1	ephem_update	63
8.15.3.2	get_dyn_time_ptr	63
8.15.3.3	get_name	63
8.15.3.4	initialize	64
8.15.3.5	operator=	64
8.15.3.6	timestamp	64
8.15.3.7	update_axial_rotation	64
8.15.3.8	update_rnp	64
8.15.4	Friends And Related Function Documentation	65
8.15.4.1	init_attrjeod__RNPJ2000	65
8.15.4.2	InputProcessor	65
8.15.5	Field Documentation	65
8.15.5.1	gmst_ptr	65
8.15.5.2	internal_name	65
8.15.5.3	last_updated_time_full	65
8.15.5.4	last_updated_time_rotational	65
8.15.5.5	never_updated_full	66

8.15.5.6	<a href="#">never_updated_rotational</a>	66
8.15.5.7	<a href="#">NJ2000</a>	66
8.15.5.8	<a href="#">PJ2000</a>	66
8.15.5.9	<a href="#">PMJ2000</a>	66
8.15.5.10	<a href="#">RJ2000</a>	66
8.15.5.11	<a href="#">time_dyn_ptr</a>	67
8.16	<a href="#">jeod::RNPJ2000_rnp_j2000_default_data Class Reference</a>	67
8.16.1	<a href="#">Detailed Description</a>	67
8.16.2	<a href="#">Member Function Documentation</a>	67
8.16.2.1	<a href="#">initialize</a>	67
8.17	<a href="#">jeod::RNPMars Class Reference</a>	67
8.17.1	<a href="#">Detailed Description</a>	69
8.17.2	<a href="#">Constructor &amp; Destructor Documentation</a>	69
8.17.2.1	<a href="#">RNPMars</a>	69
8.17.2.2	<a href="#">~RNPMars</a>	69
8.17.2.3	<a href="#">RNPMars</a>	69
8.17.3	<a href="#">Member Function Documentation</a>	69
8.17.3.1	<a href="#">ephem_update</a>	69
8.17.3.2	<a href="#">get_dyn_time_ptr</a>	70
8.17.3.3	<a href="#">get_name</a>	70
8.17.3.4	<a href="#">initialize</a>	70
8.17.3.5	<a href="#">operator=</a>	70
8.17.3.6	<a href="#">timestamp</a>	70
8.17.3.7	<a href="#">update_axial_rotation</a>	70
8.17.3.8	<a href="#">update_rnp</a>	71
8.17.4	<a href="#">Friends And Related Function Documentation</a>	71
8.17.4.1	<a href="#">init_attrjeod__RNPMars</a>	71
8.17.4.2	<a href="#">InputProcessor</a>	71
8.17.5	<a href="#">Field Documentation</a>	71
8.17.5.1	<a href="#">internal_name</a>	71
8.17.5.2	<a href="#">last_updated_time_full</a>	71
8.17.5.3	<a href="#">last_updated_time_rotational</a>	71
8.17.5.4	<a href="#">never_updated_full</a>	72
8.17.5.5	<a href="#">never_updated_rotational</a>	72
8.17.5.6	<a href="#">NMars</a>	72
8.17.5.7	<a href="#">PMars</a>	72
8.17.5.8	<a href="#">RMars</a>	72
8.17.5.9	<a href="#">time_dyn_ptr</a>	72
8.17.5.10	<a href="#">tt_ptr</a>	73
8.18	<a href="#">jeod::RNPMars_rnp_mars_default_data Class Reference</a>	73

8.18.1 Detailed Description . . . . .	73
8.18.2 Member Function Documentation . . . . .	73
8.18.2.1 initialize . . . . .	73
8.19 jeod::RNPMessages Class Reference . . . . .	73
8.19.1 Detailed Description . . . . .	74
8.19.2 Constructor & Destructor Documentation . . . . .	74
8.19.2.1 RNPMessages . . . . .	74
8.19.2.2 RNPMessages . . . . .	74
8.19.3 Member Function Documentation . . . . .	74
8.19.3.1 operator= . . . . .	74
8.19.4 Friends And Related Function Documentation . . . . .	74
8.19.4.1 init_attrjeod__RNPMessages . . . . .	74
8.19.4.2 InputProcessor . . . . .	74
8.19.5 Field Documentation . . . . .	74
8.19.5.1 fidelity_error . . . . .	74
8.19.5.2 initialization_error . . . . .	75
8.19.5.3 polar_motion_table_warning . . . . .	75
8.19.5.4 setup_error . . . . .	75
8.20 jeod::RotationJ2000 Class Reference . . . . .	76
8.20.1 Detailed Description . . . . .	76
8.20.2 Constructor & Destructor Documentation . . . . .	77
8.20.2.1 RotationJ2000 . . . . .	77
8.20.2.2 ~RotationJ2000 . . . . .	77
8.20.2.3 RotationJ2000 . . . . .	77
8.20.3 Member Function Documentation . . . . .	77
8.20.3.1 operator= . . . . .	77
8.20.3.2 update_rotation . . . . .	77
8.20.4 Friends And Related Function Documentation . . . . .	77
8.20.4.1 init_attrjeod__RotationJ2000 . . . . .	77
8.20.4.2 InputProcessor . . . . .	77
8.20.5 Field Documentation . . . . .	77
8.20.5.1 GMST . . . . .	77
8.20.5.2 nutation . . . . .	77
8.20.5.3 planet_rotational_velocity . . . . .	78
8.20.5.4 theta_gast . . . . .	78
8.20.5.5 use_full_rnp . . . . .	78
8.21 jeod::RotationMars Class Reference . . . . .	78
8.21.1 Detailed Description . . . . .	79
8.21.2 Constructor & Destructor Documentation . . . . .	79
8.21.2.1 RotationMars . . . . .	79

8.21.2.2	<a href="#">~RotationMars</a>	79
8.21.2.3	<a href="#">RotationMars</a>	80
8.21.3	<a href="#">Member Function Documentation</a>	80
8.21.3.1	<a href="#">operator=</a>	80
8.21.3.2	<a href="#">update_rotation</a>	80
8.21.4	<a href="#">Friends And Related Function Documentation</a>	80
8.21.4.1	<a href="#">init_attrjeod__RotationMars</a>	80
8.21.4.2	<a href="#">InputProcessor</a>	80
8.21.5	<a href="#">Field Documentation</a>	80
8.21.5.1	<a href="#">nutation</a>	80
8.21.5.2	<a href="#">phi_at_j2000</a>	80
8.21.5.3	<a href="#">phi_spin</a>	80
8.21.5.4	<a href="#">planet_rotational_velocity</a>	81
8.21.5.5	<a href="#">use_full_rnp</a>	81
<b>9</b>	<b><a href="#">File Documentation</a></b>	<b>83</b>
9.1	<a href="#">class_declarations.hh File Reference</a>	83
9.1.1	<a href="#">Detailed Description</a>	83
9.2	<a href="#">class_declarations.hh File Reference</a>	83
9.2.1	<a href="#">Detailed Description</a>	83
9.3	<a href="#">data_nutation_j2000.cc File Reference</a>	83
9.3.1	<a href="#">Macro Definition Documentation</a>	84
9.3.1.1	<a href="#">JEOD_FRIEND_CLASS</a>	84
9.4	<a href="#">data_rnp_j2000.cc File Reference</a>	84
9.4.1	<a href="#">Macro Definition Documentation</a>	84
9.4.1.1	<a href="#">JEOD_FRIEND_CLASS</a>	84
9.5	<a href="#">data_rnp_mars.cc File Reference</a>	85
9.5.1	<a href="#">Macro Definition Documentation</a>	85
9.5.1.1	<a href="#">JEOD_FRIEND_CLASS</a>	85
9.6	<a href="#">nutation_j2000.cc File Reference</a>	85
9.6.1	<a href="#">Detailed Description</a>	85
9.7	<a href="#">nutation_j2000.hh File Reference</a>	86
9.8	<a href="#">nutation_j2000.hh File Reference</a>	86
9.8.1	<a href="#">Detailed Description</a>	86
9.9	<a href="#">nutation_j2000_init.cc File Reference</a>	86
9.9.1	<a href="#">Detailed Description</a>	87
9.10	<a href="#">nutation_j2000_init.hh File Reference</a>	87
9.10.1	<a href="#">Detailed Description</a>	87
9.11	<a href="#">nutation_mars.cc File Reference</a>	87
9.11.1	<a href="#">Detailed Description</a>	87

9.12	nutaton_mars.hh File Reference	88
9.12.1	Detailed Description	88
9.13	planet_orientation.cc File Reference	88
9.13.1	Detailed Description	88
9.14	planet_orientation.hh File Reference	88
9.14.1	Detailed Description	89
9.15	planet_rnp.cc File Reference	89
9.15.1	Detailed Description	89
9.16	planet_rnp.hh File Reference	90
9.16.1	Detailed Description	90
9.17	planet_rotation.cc File Reference	90
9.17.1	Detailed Description	90
9.18	planet_rotation.hh File Reference	90
9.18.1	Detailed Description	91
9.19	planet_rotation_init.cc File Reference	91
9.19.1	Detailed Description	91
9.20	planet_rotation_init.hh File Reference	91
9.20.1	Detailed Description	92
9.21	polar_motion_j2000.cc File Reference	92
9.21.1	Detailed Description	92
9.22	polar_motion_j2000.hh File Reference	92
9.22.1	Detailed Description	92
9.23	polar_motion_j2000_init.cc File Reference	93
9.23.1	Detailed Description	93
9.24	polar_motion_j2000_init.hh File Reference	93
9.24.1	Detailed Description	93
9.25	precession_j2000.cc File Reference	93
9.25.1	Detailed Description	94
9.26	precession_j2000.hh File Reference	94
9.26.1	Detailed Description	94
9.27	precession_mars.cc File Reference	94
9.27.1	Detailed Description	95
9.28	precession_mars.hh File Reference	95
9.28.1	Detailed Description	95
9.29	rnp_j2000.cc File Reference	95
9.29.1	Detailed Description	96
9.30	rnp_j2000.hh File Reference	96
9.31	rnp_j2000.hh File Reference	96
9.31.1	Detailed Description	96
9.32	rnp_mars.cc File Reference	96

9.32.1 Detailed Description . . . . .	97
9.33 rnp_mars.hh File Reference . . . . .	97
9.34 rnp_mars.hh File Reference . . . . .	97
9.34.1 Detailed Description . . . . .	98
9.35 RNP_messages.cc File Reference . . . . .	98
9.35.1 Detailed Description . . . . .	98
9.36 RNP_messages.hh File Reference . . . . .	98
9.36.1 Detailed Description . . . . .	99
9.37 rotation_j2000.cc File Reference . . . . .	99
9.37.1 Detailed Description . . . . .	99
9.38 rotation_j2000.hh File Reference . . . . .	99
9.38.1 Detailed Description . . . . .	99
9.39 rotation_mars.cc File Reference . . . . .	100
9.39.1 Detailed Description . . . . .	100
9.40 rotation_mars.hh File Reference . . . . .	100
9.40.1 Detailed Description . . . . .	100
9.41 xpyp_daily.cc File Reference . . . . .	100
9.41.1 Macro Definition Documentation . . . . .	101
9.41.1.1 JEOD_FRIEND_CLASS . . . . .	101
9.42 xpyp_daily.hh File Reference . . . . .	101
9.43 xpyp_monthly.cc File Reference . . . . .	101
9.43.1 Macro Definition Documentation . . . . .	102
9.43.1.1 JEOD_FRIEND_CLASS . . . . .	102
9.44 xpyp_monthly.hh File Reference . . . . .	102

## Index

103





# Chapter 1

## Module Index

### 1.1 Modules

Here is a list of all modules:

Models . . . . .	11
Environment . . . . .	12
RNP . . . . .	13
GenericRNP . . . . .	14
RNPJ2000 . . . . .	15
RNPMars . . . . .	16



## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">jeod</a>	Namespace jeod . . . . .	17
----------------------	--------------------------	----



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

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

EphemerisInterface	
jeod::PlanetOrientation . . . . .	33
jeod::PlanetRNP . . . . .	38
jeod::RNPJ2000 . . . . .	61
jeod::RNPMars . . . . .	67
jeod::NutationJ2000Init_nutation_j2000_default_data . . . . .	28
jeod::PlanetRotation . . . . .	42
jeod::NutationJ2000 . . . . .	19
jeod::NutationMars . . . . .	29
jeod::PolarMotionJ2000 . . . . .	48
jeod::PrecessionJ2000 . . . . .	56
jeod::PrecessionMars . . . . .	58
jeod::RotationJ2000 . . . . .	76
jeod::RotationMars . . . . .	78
jeod::PlanetRotationInit . . . . .	47
jeod::NutationJ2000Init . . . . .	25
jeod::PolarMotionJ2000Init . . . . .	52
jeod::PolarMotionJ2000Init_xpyp_daily_default_data . . . . .	55
jeod::PolarMotionJ2000Init_xpyp_monthly_default_data . . . . .	56
RefFrameOwner	
jeod::PlanetOrientation . . . . .	33
jeod::RNPJ2000_rnp_j2000_default_data . . . . .	67
jeod::RNPMars_rnp_mars_default_data . . . . .	73
jeod::RNPMessages . . . . .	73



## Chapter 4

# Data Structure Index

### 4.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">jeod::NutationJ2000</a>	Implements the nutation portion of the J2000 RNP . . . . .	19
<a href="#">jeod::NutationJ2000Init</a>	The <a href="#">NutationJ2000Init</a> contains coefficients and other data used to initialize a <a href="#">NutationJ2000</a> object . . . . .	25
<a href="#">jeod::NutationJ2000Init_nutation_j2000_default_data</a>		28
<a href="#">jeod::NutationMars</a>	Implements the nutation portion of the "Pathfinder" Mars RNP model . . . . .	29
<a href="#">jeod::PlanetOrientation</a>	The generic framework for orientation models to interact with a DynManager object . . . . .	33
<a href="#">jeod::PlanetRNP</a>	The generic framework for orientation models based on the RNP paradigm . . . . .	38
<a href="#">jeod::PlanetRotation</a>	The generic base class for all planet transformations . . . . .	42
<a href="#">jeod::PlanetRotationInit</a>	The generic base class for all initializing classes for <a href="#">PlanetRotation</a> derived classes . . . . .	47
<a href="#">jeod::PolarMotionJ2000</a>	Implements the polar motion portion of the J2000 RNP . . . . .	48
<a href="#">jeod::PolarMotionJ2000Init</a>	Initializes the <a href="#">PolarMotionJ2000</a> object . . . . .	52
<a href="#">jeod::PolarMotionJ2000Init_xpyp_daily_default_data</a>		55
<a href="#">jeod::PolarMotionJ2000Init_xpyp_monthly_default_data</a>		56
<a href="#">jeod::PrecessionJ2000</a>	Implements the precession portion of the J2000 RNP . . . . .	56
<a href="#">jeod::PrecessionMars</a>	Implements the axial rotation portion of the "Pathfinder" Mars RNP model . . . . .	58
<a href="#">jeod::RNPJ2000</a>	Implements the J2000 RNP model using the generic RNP framework . . . . .	61
<a href="#">jeod::RNPJ2000_rnp_j2000_default_data</a>		67
<a href="#">jeod::RNPMars</a>	Implements the "Pathfinder" Mars RNP model using the generic RNP framework . . . . .	67
<a href="#">jeod::RNPMars_rnp_mars_default_data</a>		73
<a href="#">jeod::RNPMessages</a>	Describes messages used in the RNP model . . . . .	73
<a href="#">jeod::RotationJ2000</a>	Implements the axial rotation portion of J2000 RNP . . . . .	76
<a href="#">jeod::RotationMars</a>	Implements the axial rotation portion of the "Pathfinder" Mars RNP model . . . . .	78





## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

<a href="#">GenericRNP/include/class_declarations.hh</a>	Forward declarations of classes defined for JEOD 2.0 Generic RNP . . . . .	83
<a href="#">RNPJ2000/include/class_declarations.hh</a>	Forward declarations of classes defined for JEOD 2.0 J2000 RNP . . . . .	83
<a href="#">data_nutation_j2000.cc</a>		83
<a href="#">data_rnp_j2000.cc</a>		84
<a href="#">data_rnp_mars.cc</a>		85
<a href="#">nutation_j2000.cc</a>	Implementation for the NutationJ2000 class . . . . .	85
<a href="#">data/include/nutation_j2000.hh</a>		86
<a href="#">include/nutation_j2000.hh</a>	Model the nutation portion of the RNP routine for the Standard Epoch J2000 This is form of the IAU-76/FK5 RNP model, as implemented in Jeod 1.52 . . . . .	86
<a href="#">nutation_j2000_init.cc</a>	Implementation of NutationJ2000Init . . . . .	86
<a href="#">nutation_j2000_init.hh</a>	Used to load data and initialize the NutationJ2000 class through NutationJ2000::initialize . . .	87
<a href="#">nutation_mars.cc</a>	Implementation for the NutationMars class . . . . .	87
<a href="#">nutation_mars.hh</a>	Represent the nutation portion of the JPL-developed "Pathfinder" Mars orientation model . . .	88
<a href="#">planet_orientation.cc</a>	Implement PlanetOrientation . . . . .	88
<a href="#">planet_orientation.hh</a>	Establish a pure virtual framework for interfacing with the DynManager to set a planet orientation, meaning the transformation information from a planet's inertial frame to it's planet fixed frame .	88
<a href="#">planet_rnp.cc</a>	Implement PlanetRNP . . . . .	89
<a href="#">planet_rnp.hh</a>	Establish a framework for rotation-nutation-precession style planet attitude models . . . . .	90
<a href="#">planet_rotation.cc</a>	Implemenation for PlanetRotation . . . . .	90
<a href="#">planet_rotation.hh</a>	Pure virtual polymorphic base class for all forms of planet rotation in th RNP model including precession, nutation, polar motion and axial rotation . . . . .	90
<a href="#">planet_rotation_init.cc</a>	Implementation for PlanetRotationInit . . . . .	91

<a href="#">planet_rotation_init.hh</a>	Pure virtual polymorphic base class meant to be used by PlanetRotation::initialize when a large amount of data that could possibly change must be used for rotation calculation . . . . .	91
<a href="#">polar_motion_j2000.cc</a>	Implementation of PolarMotionJ2000 . . . . .	92
<a href="#">polar_motion_j2000.hh</a>	Model the polar motion portion of the RNP routine for the Standard Epoch J2000 . . . . .	92
<a href="#">polar_motion_j2000_init.cc</a>	Implementation for PolarMotionJ2000Init . . . . .	93
<a href="#">polar_motion_j2000_init.hh</a>	Used to load data and initialize the PolarMotionJ2000 class through PolarMotionJ2000::initialize	93
<a href="#">precession_j2000.cc</a>	Implementation for PrecessionJ2000 . . . . .	93
<a href="#">precession_j2000.hh</a>	Model the precession for the RNP routine for the Standard Epoch J2000 This is a form of the IAU-76/FK5 RNP model, as implemented in Jeod 1.52 . . . . .	94
<a href="#">precession_mars.cc</a>	Implementation of PrecessionMars . . . . .	94
<a href="#">precession_mars.hh</a>	Represent the precession portion of the JPL-developed "Pathfinder" Mars orientation model . .	95
<a href="#">rnp_j2000.cc</a>	Implementation for RNPJ2000 . . . . .	95
<a href="#">data/include/rnp_j2000.hh</a>		96
<a href="#">include/rnp_j2000.hh</a>	A specific implementation of PlanetRNP, for Earth . . . . .	96
<a href="#">rnp_mars.cc</a>	Implementation of RNPMars . . . . .	96
<a href="#">data/include/rnp_mars.hh</a>		97
<a href="#">include/rnp_mars.hh</a>	A specific implementation of PlanetRNP, for Mars . . . . .	97
<a href="#">RNP_messages.cc</a>	Implement RNP_messages . . . . .	98
<a href="#">RNP_messages.hh</a>	Implement RNP_messages . . . . .	98
<a href="#">rotation_j2000.cc</a>	Implementation of RotationJ2000 . . . . .	99
<a href="#">rotation_j2000.hh</a>	Model the axial rotation portion of the RNP routine for the Standard Epoch J2000 . . . . .	99
<a href="#">rotation_mars.cc</a>	Implementation of RotationMars . . . . .	100
<a href="#">rotation_mars.hh</a>	Represent the axial rotation portion of the JPL-developed "Pathfinder" Mars orientation model .	100
<a href="#">xpyp_daily.cc</a>		100
<a href="#">xpyp_daily.hh</a>		101
<a href="#">xpyp_monthly.cc</a>		101
<a href="#">xpyp_monthly.hh</a>		102

## Chapter 6

# Module Documentation

### 6.1 Models

#### Modules

- [Environment](#)

#### 6.1.1 Detailed Description

## 6.2 Environment

### Modules

- [RNP](#)

### 6.2.1 Detailed Description

## 6.3 RNP

### Modules

- [GenericRNP](#)
- [RNPJ2000](#)
- [RNPMars](#)

### 6.3.1 Detailed Description

## 6.4 GenericRNP

### Files

- file [GenericRNP/include/class\\_declarations.hh](#)  
*Forward declarations of classes defined for JEOD 2.0 Generic RNP.*
- file [planet\\_orientation.hh](#)  
*Establish a pure virtual framework for interfacing with the DynManager to set a planet orientation, meaning the transformation information from a planet's inertial frame to it's planet fixed frame.*
- file [planet\\_rnp.hh](#)  
*Establish a framework for rotation-nutation-precession style planet attitude models.*
- file [planet\\_rotation.hh](#)  
*Pure virtual polymorphic base class for all forms of planet rotation in th RNP model including precession, nutation, polar motion and axial rotation.*
- file [planet\\_rotation\\_init.hh](#)  
*Pure virtual polymorphic base class meant to be used by PlanetRotation::initialize when a large amount of data that could possibly change must be used for rotation calculation.*
- file [RNP\\_messages.hh](#)  
*Implement RNP\_messages.*
- file [planet\\_orientation.cc](#)  
*Implement PlanetOrientation.*
- file [planet\\_rnp.cc](#)  
*Implement PlanetRNP.*
- file [planet\\_rotation.cc](#)  
*Implemenation for PlanetRotation.*
- file [planet\\_rotation\\_init.cc](#)  
*Implementation for PlanetRotationInit.*
- file [RNP\\_messages.cc](#)  
*Implement RNP\_messages.*

### Namespaces

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

### Macros

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

#### 6.4.1 Detailed Description

#### 6.4.2 Macro Definition Documentation

##### 6.4.2.1 `#define PATH "environment/RNP/"`

Definition at line 41 of file RNP\_messages.cc.

## 6.5 RNPJ2000

### Files

- file [RNPJ2000/include/class\\_declarations.hh](#)  
*Forward declarations of classes defined for JEOD 2.0 J2000 RNP.*
- file [include/nutation\\_j2000.hh](#)  
*Model the nutation portion of the RNP routine for the Standard Epoch J2000 This is form of the IAU-76/FK5 RNP model, as implemented in Jeod 1.52.*
- file [nutation\\_j2000\\_init.hh](#)  
*Used to load data and initialize the NutationJ2000 class through NutationJ2000::initialize.*
- file [polar\\_motion\\_j2000.hh](#)  
*Model the polar motion portion of the RNP routine for the Standard Epoch J2000.*
- file [polar\\_motion\\_j2000\\_init.hh](#)  
*Used to load data and initialize the PolarMotionJ2000 class through PolarMotionJ2000::initialize.*
- file [precession\\_j2000.hh](#)  
*Model the precession for the RNP routine for the Standard Epoch J2000 This is a form of the IAU-76/FK5 RNP model, as implemented in Jeod 1.52.*
- file [include/rnp\\_j2000.hh](#)  
*A specific implementation of PlanetRNP, for Earth.*
- file [rotation\\_j2000.hh](#)  
*Model the axial rotation portion of the RNP routine for the Standard Epoch J2000.*
- file [nutation\\_j2000.cc](#)  
*Implementation for the NutationJ2000 class.*
- file [nutation\\_j2000\\_init.cc](#)  
*Implementation of NutationJ2000Init.*
- file [polar\\_motion\\_j2000.cc](#)  
*Implementation of PolarMotionJ2000.*
- file [polar\\_motion\\_j2000\\_init.cc](#)  
*Implementation for PolarMotionJ2000Init.*
- file [precession\\_j2000.cc](#)  
*Implementation for PrecessionJ2000.*
- file [rnp\\_j2000.cc](#)  
*Implementation for RNPJ2000.*
- file [rotation\\_j2000.cc](#)  
*Implementation of RotationJ2000.*

### Namespaces

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

#### 6.5.1 Detailed Description

## 6.6 RNPMars

### Files

- file [nutation\\_mars.hh](#)  
*Represent the nutation portion of the JPL-developed "Pathfinder" Mars orientation model.*
- file [precession\\_mars.hh](#)  
*Represent the precession portion of the JPL-developed "Pathfinder" Mars orientation model.*
- file [include/rnp\\_mars.hh](#)  
*A specific implementation of PlanetRNP, for Mars.*
- file [rotation\\_mars.hh](#)  
*Represent the axial rotation portion of the JPL-developed "Pathfinder" Mars orientation model.*
- file [nutation\\_mars.cc](#)  
*Implementation for the NutationMars class.*
- file [precession\\_mars.cc](#)  
*Implementation of PrecessionMars.*
- file [rnp\\_mars.cc](#)  
*Implementation of RNPMars.*
- file [rotation\\_mars.cc](#)  
*Implementation of RotationMars.*

### Namespaces

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

#### 6.6.1 Detailed Description



## Chapter 7

# Namespace Documentation

### 7.1 jeod Namespace Reference

Namespace jeod.

#### Data Structures

- class [PlanetOrientation](#)  
*The generic framework for orientation models to interact with a DynManager object.*
- class [PlanetRNP](#)  
*The generic framework for orientation models based on the RNP paradigm.*
- class [PlanetRotation](#)  
*The generic base class for all planet transformations.*
- class [PlanetRotationInit](#)  
*The generic base class for all initializing classes for [PlanetRotation](#) derived classes.*
- class [RNPMessages](#)  
*Describes messages used in the RNP model.*
- class [NutationJ2000Init\\_nutation\\_j2000\\_default\\_data](#)
- class [RNPJ2000\\_rnp\\_j2000\\_default\\_data](#)
- class [PolarMotionJ2000Init\\_xpyp\\_daily\\_default\\_data](#)
- class [PolarMotionJ2000Init\\_xpyp\\_monthly\\_default\\_data](#)
- class [NutationJ2000](#)  
*Implements the nutation portion of the J2000 RNP.*
- class [NutationJ2000Init](#)  
*The [NutationJ2000Init](#) contains coefficients and other data used to initialize a [NutationJ2000](#) object.*
- class [PolarMotionJ2000](#)  
*Implements the polar motion portion of the J2000 RNP.*
- class [PolarMotionJ2000Init](#)  
*Initializes the [PolarMotionJ2000](#) object.*
- class [PrecessionJ2000](#)  
*Implements the precession portion of the J2000 RNP.*
- class [RNPJ2000](#)  
*Implements the J2000 RNP model using the generic RNP framework.*
- class [RotationJ2000](#)  
*Implements the axial rotation portion of J2000 RNP.*
- class [RNPMars\\_rnp\\_mars\\_default\\_data](#)
- class [NutationMars](#)

- Implements the nutation portion of the "Pathfinder" Mars RNP model.*
- class [PrecessionMars](#)
  - Implements the axial rotation portion of the "Pathfinder" Mars RNP model.*
- class [RNPMars](#)
  - Implements the "Pathfinder" Mars RNP model using the generic RNP framework.*
- class [RotationMars](#)
  - Implements the axial rotation portion of the "Pathfinder" Mars RNP model.*

### 7.1.1 Detailed Description

Namespace jeod.

## Chapter 8

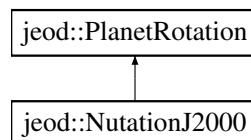
# Data Structure Documentation

### 8.1 jeod::NutationJ2000 Class Reference

Implements the nutation portion of the J2000 RNP.

```
#include <nutation_j2000.hh>
```

Inheritance diagram for jeod::NutationJ2000:



#### Public Member Functions

- [NutationJ2000](#) ()  
*Constructor.*
- virtual [~NutationJ2000](#) ()  
*destructor*
- virtual void [update\\_rotation](#) ()  
*Specific implementation of update\_rotation, from the polymorphic pure virtual base class [PlanetRotation](#).*
- virtual void [initialize](#) ([PlanetRotationInit](#) \*init)  
*Initialize the various coefficients needed for the calculation of nutation.*

#### Data Fields

- unsigned int [num\\_coeffs](#)  
*The number of coefficients in the 9 arrays directly following this declaration.*
- double \* [L\\_coeffs](#)  
*The coefficients to calculate mean anomaly of the moon.*
- double \* [M\\_coeffs](#)  
*The coefficients to calculate mean anomaly of the sun.*
- double \* [F\\_coeffs](#)  
*The coefficients to calculate mean argument of latitude of the moon.*
- double \* [D\\_coeffs](#)  
*The coefficients to calculate mean elongation from the sun.*

- double \* [omega\\_coeffs](#)  
*The coefficients to calculate the right ascension of the ascending node of the mean mean lunar orbit.*
- double \* [long\\_coeffs](#)  
*Zero order (in time) coefficients for calculating the nutation in longitude.*
- double \* [long\\_t\\_coeffs](#)  
*The first order (in time) coefficients for calculating the nutation in longitude.*
- double \* [obliq\\_coeffs](#)  
*Zero order coefficients for calculating the nutation in obliquity.*
- double \* [obliq\\_t\\_coeffs](#)  
*First order coefficients for calculating the nutation in obliquity.*
- double [nutation\\_in\\_longitude](#)  
*The nutation in longitude from the last call of update.*
- double [nutation\\_in\\_obliquity](#)  
*The nutation in obliquity from the last call of update.*
- double [L](#)  
*Last calculated mean anomaly of the moon.*
- double [M](#)  
*Last calculated mean anomaly of the sun.*
- double [F](#)  
*Last calculated mean argument of latitude of the moon.*
- double [D](#)  
*Last calculated mean elongation from the sun.*
- double [omega](#)  
*Last calculated ascension of the ascending node of the mean lunar orbit.*
- double [epsilon\\_bar](#)  
*Last calculated mean obliquity of the ecliptic.*
- double [equa\\_of\\_equi](#)  
*Last calculated equations of the equinox.*

## Private Member Functions

- [NutationJ2000](#) & [operator=](#) (const [NutationJ2000](#) &rhs)
- [NutationJ2000](#) (const [NutationJ2000](#) &rhs)

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_NutationJ2000](#) ()

## Additional Inherited Members

### 8.1.1 Detailed Description

Implements the nutation portion of the J2000 RNP.

Definition at line 63 of file `include/nutation_j2000.hh`.

## 8.1.2 Constructor & Destructor Documentation

### 8.1.2.1 jeod::NutationJ2000::NutationJ2000 ( void )

Constructor.

Initialize all in class data

Definition at line 73 of file nutation\_j2000.cc.

### 8.1.2.2 jeod::NutationJ2000::~~NutationJ2000 ( void ) [virtual]

destructor

Definition at line 103 of file nutation\_j2000.cc.

References `D_coeffs`, `F_coeffs`, `L_coeffs`, `long_coeffs`, `long_t_coeffs`, `M_coeffs`, `obliq_coeffs`, `obliq_t_coeffs`, and `omega_coeffs`.

### 8.1.2.3 jeod::NutationJ2000::NutationJ2000 ( const NutationJ2000 & rhs ) [private]

## 8.1.3 Member Function Documentation

### 8.1.3.1 void jeod::NutationJ2000::initialize ( PlanetRotationInit \* init ) [virtual]

Initialize the various coefficients needed for the calculation of nutation.

init must be of type [NutationJ2000Init](#) or a fail message will occur

Parameters

in	init	<a href="#">NutationJ2000Init</a> object with needed coefficients
----	------	---

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

Definition at line 276 of file nutation\_j2000.cc.

References `jeod::NutationJ2000Init::D_coeffs`, `D_coeffs`, `jeod::NutationJ2000Init::F_coeffs`, `F_coeffs`, `jeod::RN-PMessages::initialization_error`, `jeod::NutationJ2000Init::L_coeffs`, `L_coeffs`, `jeod::NutationJ2000Init::long_coeffs`, `long_coeffs`, `jeod::NutationJ2000Init::long_t_coeffs`, `long_t_coeffs`, `jeod::NutationJ2000Init::M_coeffs`, `M_coeffs`, `jeod::NutationJ2000Init::num_coeffs`, `num_coeffs`, `jeod::NutationJ2000Init::obliq_coeffs`, `obliq_coeffs`, `jeod::NutationJ2000Init::obliq_t_coeffs`, `obliq_t_coeffs`, `jeod::NutationJ2000Init::omega_coeffs`, and `omega_coeffs`.

### 8.1.3.2 NutationJ2000& jeod::NutationJ2000::operator= ( const NutationJ2000 & rhs ) [private]

### 8.1.3.3 void jeod::NutationJ2000::update\_rotation ( void ) [virtual]

Specific implementation of `update_rotation`, from the polymorphic pure virtual base class [PlanetRotation](#).

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

Definition at line 151 of file nutation\_j2000.cc.

References `jeod::PlanetRotation::current_time`, `D`, `D_coeffs`, `jeod::PlanetRotation::DEGTORAD`, `epsilon_bar`, `equa_of_equi`, `F`, `F_coeffs`, `L`, `L_coeffs`, `long_coeffs`, `long_t_coeffs`, `M`, `M_coeffs`, `num_coeffs`, `nutation_in_longitude`, `nutation_in_obliquity`, `obliq_coeffs`, `obliq_t_coeffs`, `omega`, `omega_coeffs`, and `jeod::PlanetRotation::rotation`.

## 8.1.4 Friends And Related Function Documentation

### 8.1.4.1 void init\_attrjeod\_\_NutationJ2000 ( ) [friend]

#### 8.1.4.2 friend class InputProcessor [friend]

Definition at line 65 of file include/nutation\_j2000.hh.

### 8.1.5 Field Documentation

#### 8.1.5.1 double jeod::NutationJ2000::D

Last calculated mean elongation from the sun.

trick\_units(—)

Definition at line 146 of file include/nutation\_j2000.hh.

Referenced by update\_rotation().

#### 8.1.5.2 double\* jeod::NutationJ2000::D\_coeffs

The coefficients to calculate mean elongation from the sun.

trick\_units(—)

Definition at line 91 of file include/nutation\_j2000.hh.

Referenced by initialize(), update\_rotation(), and ~NutationJ2000().

#### 8.1.5.3 double jeod::NutationJ2000::epsilon\_bar

Last calculated mean obliquity of the ecliptic.

trick\_units(—)

Definition at line 156 of file include/nutation\_j2000.hh.

Referenced by update\_rotation().

#### 8.1.5.4 double jeod::NutationJ2000::equa\_of\_equi

Last calculated equations of the equinox.

trick\_units(—)

Definition at line 161 of file include/nutation\_j2000.hh.

Referenced by jeod::RotationJ2000::update\_rotation(), and update\_rotation().

#### 8.1.5.5 double jeod::NutationJ2000::F

Last calculated mean argument of latitude of the moon.

trick\_units(—)

Definition at line 142 of file include/nutation\_j2000.hh.

Referenced by update\_rotation().

#### 8.1.5.6 double\* jeod::NutationJ2000::F\_coeffs

The coefficients to calculate mean argument of latitude of the moon.

trick\_units(—)

Definition at line 87 of file include/nutation\_j2000.hh.

Referenced by initialize(), update\_rotation(), and ~NutationJ2000().

#### 8.1.5.7 double jeod::NutationJ2000::L

Last calculated mean anomaly of the moon.

trick\_units(–)

Definition at line 134 of file include/nutation\_j2000.hh.

Referenced by update\_rotation().

#### 8.1.5.8 double\* jeod::NutationJ2000::L\_coeffs

The coefficients to calculate mean anomaly of the moon.

trick\_units(–)

Definition at line 79 of file include/nutation\_j2000.hh.

Referenced by initialize(), update\_rotation(), and ~NutationJ2000().

#### 8.1.5.9 double\* jeod::NutationJ2000::long\_coeffs

Zero order (in time) coefficients for calculating the nutation in longitude.

trick\_units(–)

Definition at line 103 of file include/nutation\_j2000.hh.

Referenced by initialize(), update\_rotation(), and ~NutationJ2000().

#### 8.1.5.10 double\* jeod::NutationJ2000::long\_t\_coeffs

The first order (in time) coefficients for calculating the nutation in longitude.

trick\_units(–)

Definition at line 108 of file include/nutation\_j2000.hh.

Referenced by initialize(), update\_rotation(), and ~NutationJ2000().

#### 8.1.5.11 double jeod::NutationJ2000::M

Last calculated mean anomaly of the sun.

trick\_units(–)

Definition at line 138 of file include/nutation\_j2000.hh.

Referenced by update\_rotation().

#### 8.1.5.12 double\* jeod::NutationJ2000::M\_coeffs

The coefficients to calculate mean anomaly of the sun.

trick\_units(–)

Definition at line 83 of file include/nutation\_j2000.hh.

Referenced by initialize(), update\_rotation(), and ~NutationJ2000().

**8.1.5.13 unsigned int jeod::NutationJ2000::num\_coeffs**

The number of coefficients in the 9 arrays directly following this declaration.

trick\_units(count)

Definition at line 74 of file include/nutation\_j2000.hh.

Referenced by initialize(), and update\_rotation().

**8.1.5.14 double jeod::NutationJ2000::nututation\_in\_longitude**

The nutation in longitude from the last call of update.

trick\_units(—)

Definition at line 124 of file include/nutation\_j2000.hh.

Referenced by update\_rotation().

**8.1.5.15 double jeod::NutationJ2000::nututation\_in\_obliquity**

The nutation in obliquity from the last call of update.

trick\_units(—)

Definition at line 129 of file include/nutation\_j2000.hh.

Referenced by update\_rotation().

**8.1.5.16 double\* jeod::NutationJ2000::obliq\_coeffs**

Zero order coefficients for calculating the nutation in obliquity.

trick\_units(—)

Definition at line 113 of file include/nutation\_j2000.hh.

Referenced by initialize(), update\_rotation(), and ~NutationJ2000().

**8.1.5.17 double\* jeod::NutationJ2000::obliq\_t\_coeffs**

First order coefficients for calculating the nutation in obliquity.

trick\_units(—)

Definition at line 118 of file include/nutation\_j2000.hh.

Referenced by initialize(), update\_rotation(), and ~NutationJ2000().

**8.1.5.18 double jeod::NutationJ2000::omega**

Last calculated ascension of the ascending node of the mean lunar orbit.

trick\_units(—)

Definition at line 151 of file include/nutation\_j2000.hh.

Referenced by update\_rotation().

**8.1.5.19 double\* jeod::NutationJ2000::omega\_coeffs**

The coefficients to calculate the right ascension of the ascending node of the mean mean lunar orbit.



trick\_units(-)

Definition at line 97 of file include/nutation\_j2000.hh.

Referenced by initialize(), update\_rotation(), and ~NutationJ2000().

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

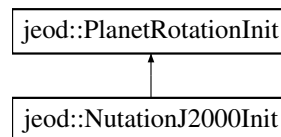
- [include/nutation\\_j2000.hh](#)
- [nututation\\_j2000.cc](#)

## 8.2 jeod::NutationJ2000Init Class Reference

The [NutationJ2000Init](#) contains coefficients and other data used to initialize a [NutationJ2000](#) object.

```
#include <nututation_j2000_init.hh>
```

Inheritance diagram for jeod::NutationJ2000Init:



### Public Member Functions

- [NutationJ2000Init](#) ()  
*constructor.*
- virtual [~NutationJ2000Init](#) ()  
*Destructor.*

### Data Fields

- int [num\\_coeffs](#)  
*The number of coefficients in the 9 arrays directly following this declaration.*
- double \* [L\\_coeffs](#)  
*The coefficients to calculate mean anomaly of the moon.*
- double \* [M\\_coeffs](#)  
*The coefficients to calculate mean anomaly of the sun.*
- double \* [F\\_coeffs](#)  
*The coefficients to calculate mean argument of latitude of the moon.*
- double \* [D\\_coeffs](#)  
*The coefficients to calculate mean elongation from the sun.*
- double \* [omega\\_coeffs](#)  
*The coefficients to calculate the right ascension of the ascending node of the mean mean lunar orbit.*
- double \* [long\\_coeffs](#)  
*Zero order (in time) coefficients for calculating the nutation in longitude.*
- double \* [long\\_t\\_coeffs](#)  
*The first order (in time) coefficients for calculating the nutation in longitude.*
- double \* [obliq\\_coeffs](#)  
*Zero order coefficients for calculating the nutation in obliquity.*
- double \* [obliq\\_t\\_coeffs](#)  
*First order coefficients for calculating the nutation in obliquity.*

## Private Member Functions

- [NutationJ2000Init](#) & `operator=` (const [NutationJ2000Init](#) &rhs)
- [NutationJ2000Init](#) (const [NutationJ2000Init](#) &rhs)

## Friends

- class [InputProcessor](#)
- void `init_attrjeod__NutationJ2000Init` ()

### 8.2.1 Detailed Description

The [NutationJ2000Init](#) contains coefficients and other data used to initialize a [NutationJ2000](#) object.

Definition at line 63 of file `nutations_j2000_init.hh`.

### 8.2.2 Constructor & Destructor Documentation

#### 8.2.2.1 `jeod::NutationJ2000Init::NutationJ2000Init ( void )`

constructor.

initialize low level data

Definition at line 63 of file `nutations_j2000_init.cc`.

#### 8.2.2.2 `jeod::NutationJ2000Init::~~NutationJ2000Init ( void )` `[virtual]`

Destructor.

Definition at line 83 of file `nutations_j2000_init.cc`.

References `D_coeffs`, `F_coeffs`, `L_coeffs`, `long_coeffs`, `long_t_coeffs`, `M_coeffs`, `obliq_coeffs`, `obliq_t_coeffs`, and `omega_coeffs`.

#### 8.2.2.3 `jeod::NutationJ2000Init::NutationJ2000Init ( const NutationJ2000Init & rhs )` `[private]`

### 8.2.3 Member Function Documentation

#### 8.2.3.1 `NutationJ2000Init& jeod::NutationJ2000Init::operator= ( const NutationJ2000Init & rhs )` `[private]`

### 8.2.4 Friends And Related Function Documentation

#### 8.2.4.1 `void init_attrjeod__NutationJ2000Init ( )` `[friend]`

#### 8.2.4.2 `friend class InputProcessor` `[friend]`

Definition at line 65 of file `nutations_j2000_init.hh`.

### 8.2.5 Field Documentation

#### 8.2.5.1 `double* jeod::NutationJ2000Init::D_coeffs`

The coefficients to calculate mean elongation from the sun.

trick\_units(-)

Definition at line 90 of file nutation\_j2000\_init.hh.

Referenced by jeod::NutationJ2000Init\_nutation\_j2000\_default\_data::initialize(), jeod::NutationJ2000::initialize(), and ~NutationJ2000Init().

#### 8.2.5.2 double\* jeod::NutationJ2000Init::F\_coeffs

The coefficients to calculate mean argument of latitude of the moon.

trick\_units(-)

Definition at line 86 of file nutation\_j2000\_init.hh.

Referenced by jeod::NutationJ2000Init\_nutation\_j2000\_default\_data::initialize(), jeod::NutationJ2000::initialize(), and ~NutationJ2000Init().

#### 8.2.5.3 double\* jeod::NutationJ2000Init::L\_coeffs

The coefficients to calculate mean anomaly of the moon.

trick\_units(-)

Definition at line 78 of file nutation\_j2000\_init.hh.

Referenced by jeod::NutationJ2000Init\_nutation\_j2000\_default\_data::initialize(), jeod::NutationJ2000::initialize(), and ~NutationJ2000Init().

#### 8.2.5.4 double\* jeod::NutationJ2000Init::long\_coeffs

Zero order (in time) coefficients for calculating the nutation in longitude.

trick\_units(-)

Definition at line 102 of file nutation\_j2000\_init.hh.

Referenced by jeod::NutationJ2000Init\_nutation\_j2000\_default\_data::initialize(), jeod::NutationJ2000::initialize(), and ~NutationJ2000Init().

#### 8.2.5.5 double\* jeod::NutationJ2000Init::long\_t\_coeffs

The first order (in time) coefficients for calculating the nutation in longitude.

trick\_units(-)

Definition at line 107 of file nutation\_j2000\_init.hh.

Referenced by jeod::NutationJ2000Init\_nutation\_j2000\_default\_data::initialize(), jeod::NutationJ2000::initialize(), and ~NutationJ2000Init().

#### 8.2.5.6 double\* jeod::NutationJ2000Init::M\_coeffs

The coefficients to calculate mean anomaly of the sun.

trick\_units(-)

Definition at line 82 of file nutation\_j2000\_init.hh.

Referenced by jeod::NutationJ2000Init\_nutation\_j2000\_default\_data::initialize(), jeod::NutationJ2000::initialize(), and ~NutationJ2000Init().

#### 8.2.5.7 `int jeod::NutationJ2000Init::num_coeffs`

The number of coefficients in the 9 arrays directly following this declaration.

`trick_units(count)`

Definition at line 73 of file `nututation_j2000_init.hh`.

Referenced by `jeod::NutationJ2000Init_nutation_j2000_default_data::initialize()`, and `jeod::NutationJ2000::initialize()`.

#### 8.2.5.8 `double* jeod::NutationJ2000Init::obliq_coeffs`

Zero order coefficients for calculating the nutation in obliquity.

`trick_units(-)`

Definition at line 112 of file `nututation_j2000_init.hh`.

Referenced by `jeod::NutationJ2000Init_nutation_j2000_default_data::initialize()`, `jeod::NutationJ2000::initialize()`, and `~NutationJ2000Init()`.

#### 8.2.5.9 `double* jeod::NutationJ2000Init::obliq_t_coeffs`

First order coefficients for calculating the nutation in obliquity.

`trick_units(-)`

Definition at line 117 of file `nututation_j2000_init.hh`.

Referenced by `jeod::NutationJ2000Init_nutation_j2000_default_data::initialize()`, `jeod::NutationJ2000::initialize()`, and `~NutationJ2000Init()`.

#### 8.2.5.10 `double* jeod::NutationJ2000Init::omega_coeffs`

The coefficients to calculate the right ascension of the ascending node of the mean mean lunar orbit.

`trick_units(-)`

Definition at line 96 of file `nututation_j2000_init.hh`.

Referenced by `jeod::NutationJ2000Init_nutation_j2000_default_data::initialize()`, `jeod::NutationJ2000::initialize()`, and `~NutationJ2000Init()`.

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

- [nututation\\_j2000\\_init.hh](#)
- [nututation\\_j2000\\_init.cc](#)

## 8.3 `jeod::NutationJ2000Init_nutation_j2000_default_data` Class Reference

```
#include <nututation_j2000.hh>
```

### Public Member Functions

- void [initialize](#) (`NutationJ2000Init *`)

#### 8.3.1 Detailed Description

Definition at line 18 of file `data/include/nutation_j2000.hh`.

### 8.3.2 Member Function Documentation

#### 8.3.2.1 void jeod::NutationJ2000Init\_nutation\_j2000\_default\_data::initialize ( NutationJ2000Init \* NutationJ2000Init\_ptr )

Definition at line 42 of file data\_nutation\_j2000.cc.

References jeod::NutationJ2000Init::D\_coeffs, jeod::NutationJ2000Init::F\_coeffs, jeod::NutationJ2000Init::L\_coeffs, jeod::NutationJ2000Init::long\_coeffs, jeod::NutationJ2000Init::long\_t\_coeffs, jeod::NutationJ2000Init::M\_coeffs, jeod::NutationJ2000Init::num\_coeffs, jeod::NutationJ2000Init::obliq\_coeffs, jeod::NutationJ2000Init::obliq\_t\_coeffs, and jeod::NutationJ2000Init::omega\_coeffs.

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

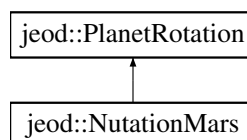
- [data/include/nutation\\_j2000.hh](#)
- [data\\_nutation\\_j2000.cc](#)

## 8.4 jeod::NutationMars Class Reference

Implements the nutation portion of the "Pathfinder" Mars RNP model.

```
#include <nutation_mars.hh>
```

Inheritance diagram for jeod::NutationMars:



### Public Member Functions

- [NutationMars](#) ()  
*Default constructor.*
- virtual [~NutationMars](#) ()  
*Destructor.*
- virtual void [update\\_rotation](#) ()  
*NutationMars specific implementation of update\_rotation, used here to describe nutation effects in longitude and obliquity.*

### Data Fields

- double [nutation\\_in\\_longitude](#)  
*The latest calculated nutation correction term in longitude.*
- double [nutation\\_in\\_obliquity](#)  
*The latest calculated nutation correction term in obliquity.*
- double [l\\_at\\_j2000](#)  
*The (constant) obliquity angle of Mars at the J2000 epoch.*
- double [l\\_dot](#)  
*The (constant) simple secular change in Mars obliquity relative to the Mars mean orbit.*
- double [obliquity\\_angle](#)  
*The latest calculated angle of obliquity, that is, the current orbit inclination angle as measured relative to the Mars mean orbit, measured since the J2000 epoch.*
- double [mean\\_motion](#)

*The mean motion of Mars, that is, the average orbit arc traversed.*

- double [mean\\_anomaly\\_j2000](#)

*The Mars mean anomaly at the J2000 epoch.*

- double [q\\_angle\\_j2000](#)

*Parameter  $q$  is a slowly varying angle defined in Konopliv 2006 as 2 times  $w$ , where  $w$  is the argument of perihelion of the Mars orbit relative to the node of the Mars equator and Mars mean orbit.*

- double \* [l\\_m\\_orig](#)

*The obliquity nutation amplitude parameters.*

- double \* [psi\\_m\\_orig](#)

*The longitude nutation amplitude parameters.*

- double \* [int\\_to\\_double](#)

*0 to 9 cast as doubles for use in summation*

## Private Member Functions

- [NutationMars](#) & [operator=](#) (const [NutationMars](#) &rhs)
- [NutationMars](#) (const [NutationMars](#) &rhs)

## Friends

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

## Additional Inherited Members

### 8.4.1 Detailed Description

Implements the nutation portion of the "Pathfinder" Mars RNP model.

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

### 8.4.2 Constructor & Destructor Documentation

#### 8.4.2.1 `jeod::NutationMars::NutationMars ( void )`

Default constructor.

Definition at line 66 of file `nutation_mars.cc`.

#### 8.4.2.2 `jeod::NutationMars::~~NutationMars ( void ) [virtual]`

Destructor.

Definition at line 87 of file `nutation_mars.cc`.

References `l_m_orig`, `int_to_double`, and `psi_m_orig`.

#### 8.4.2.3 `jeod::NutationMars::NutationMars ( const NutationMars & rhs ) [private]`

### 8.4.3 Member Function Documentation

#### 8.4.3.1 `NutationMars& jeod::NutationMars::operator= ( const NutationMars & rhs ) [private]`

#### 8.4.3.2 void jeod::NutationMars::update\_rotation ( void ) [virtual]

[NutationMars](#) specific implementation of update\_rotation, used here to describe nutation effects in longitude and obliquity.

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

Definition at line 112 of file nutation\_mars.cc.

References [jeod::PlanetRotation::current\\_time](#), [l\\_at\\_j2000](#), [l\\_dot](#), [l\\_m\\_orig](#), [int\\_to\\_double](#), [mean\\_anomaly\\_j2000](#), [mean\\_motion](#), [nutation\\_in\\_longitude](#), [nutation\\_in\\_obliquity](#), [obliquity\\_angle](#), [psi\\_m\\_orig](#), [q\\_angle\\_j2000](#), and [jeod::PlanetRotation::rotation](#).

### 8.4.4 Friends And Related Function Documentation

#### 8.4.4.1 void init\_attrjeod\_\_NutationMars ( ) [friend]

#### 8.4.4.2 friend class InputProcessor [friend]

Definition at line 64 of file nutation\_mars.hh.

### 8.4.5 Field Documentation

#### 8.4.5.1 double jeod::NutationMars::l\_at\_j2000

The (constant) obliquity angle of Mars at the J2000 epoch.

trick\_units(radian)

Definition at line 81 of file nutation\_mars.hh.

Referenced by [jeod::RNPMars\\_rnp\\_mars\\_default\\_data::initialize\(\)](#), and [update\\_rotation\(\)](#).

#### 8.4.5.2 double jeod::NutationMars::l\_dot

The (constant) simple secular change in Mars obliquity relative to the Mars mean orbit.

trick\_units(radian/s)

Definition at line 87 of file nutation\_mars.hh.

Referenced by [jeod::RNPMars\\_rnp\\_mars\\_default\\_data::initialize\(\)](#), and [update\\_rotation\(\)](#).

#### 8.4.5.3 double\* jeod::NutationMars::l\_m\_orig

The obliquity nutation amplitude parameters.

trick\_units(radian)

Definition at line 117 of file nutation\_mars.hh.

Referenced by [jeod::RNPMars\\_rnp\\_mars\\_default\\_data::initialize\(\)](#), [update\\_rotation\(\)](#), and [~NutationMars\(\)](#).

#### 8.4.5.4 double\* jeod::NutationMars::int\_to\_double

0 to 9 cast as doubles for use in summation

trick\_units(-)

Definition at line 127 of file nutation\_mars.hh.

Referenced by [jeod::RNPMars\\_rnp\\_mars\\_default\\_data::initialize\(\)](#), [update\\_rotation\(\)](#), and [~NutationMars\(\)](#).

#### 8.4.5.5 `double jeod::NutationMars::mean_anomaly_j2000`

The Mars mean anomaly at the J2000 epoch.

trick\_units(radian)

Definition at line 104 of file nutation\_mars.hh.

Referenced by `jeod::RNPMars_rnp_mars_default_data::initialize()`, and `update_rotation()`.

#### 8.4.5.6 `double jeod::NutationMars::mean_motion`

The mean motion of Mars, that is, the average orbit arc traversed.

trick\_units(radian/s)

Definition at line 99 of file nutation\_mars.hh.

Referenced by `jeod::RNPMars_rnp_mars_default_data::initialize()`, and `update_rotation()`.

#### 8.4.5.7 `double jeod::NutationMars::nututation_in_longitude`

The latest calculated nutation correction term in longitude.

trick\_units(radian)

Definition at line 71 of file nutation\_mars.hh.

Referenced by `jeod::RotationMars::update_rotation()`, `jeod::PrecessionMars::update_rotation()`, and `update_rotation()`.

#### 8.4.5.8 `double jeod::NutationMars::nututation_in_obliquity`

The latest calculated nutation correction term in obliquity.

trick\_units(radian)

Definition at line 76 of file nutation\_mars.hh.

Referenced by `update_rotation()`.

#### 8.4.5.9 `double jeod::NutationMars::obliquity_angle`

The latest calculated angle of obliquity, that is, the current orbit inclination angle as measured relative to the Mars mean orbit, measured since the J2000 epoch.

trick\_units(radian)

Definition at line 94 of file nutation\_mars.hh.

Referenced by `jeod::RotationMars::update_rotation()`, and `update_rotation()`.

#### 8.4.5.10 `double* jeod::NutationMars::psi_m_orig`

The longitude nutation amplitude parameters.

trick\_units(radian)

Definition at line 122 of file nutation\_mars.hh.

Referenced by `jeod::RNPMars_rnp_mars_default_data::initialize()`, `update_rotation()`, and `~NutationMars()`.



## 8.4.5.11 double jeod::NutationMars::q\_angle\_j2000

Parameter  $q$  is a slowly varying angle defined in Konopliv 2006 as 2 times  $w$ , where  $w$  is the argument of perihelion of the Mars orbit relative to the node of the Mars equator and Mars mean orbit.

This data member is the value of  $q$  at the J2000 epoch.trick\_units(radian)

Definition at line 112 of file nutation\_mars.hh.

Referenced by jeod::RNPMars\_rnp\_mars\_default\_data::initialize(), and update\_rotation().

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

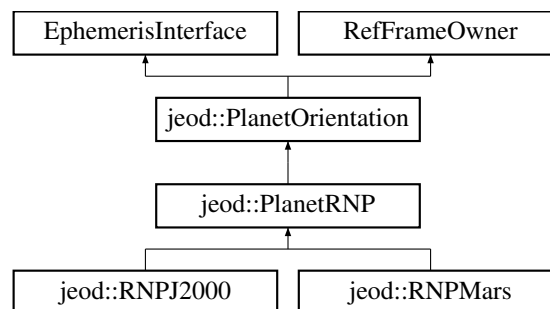
- [nutation\\_mars.hh](#)
- [nutation\\_mars.cc](#)

## 8.5 jeod::PlanetOrientation Class Reference

The generic framework for orientation models to interact with a DynManager object.

```
#include <planet_orientation.hh>
```

Inheritance diagram for jeod::PlanetOrientation:



### Public Member Functions

- [PlanetOrientation](#) ()  
*Default constructor; constructs a [PlanetOrientation](#) object.*
- virtual [~PlanetOrientation](#) ()  
*Class destructor.*
- virtual void [initialize](#) (DynManager &dyn\_manager)  
*Goes to the dyn manager given and searches for the planet indicated by the user inputted name, in the given dyn manager.*
- virtual void [activate](#) ()  
*Activates the [PlanetOrientation](#) model.*
- virtual void [deactivate](#) ()  
*Deactivates the [PlanetOrientation](#) model.*
- virtual void [ephem\\_initialize](#) (EphemeridesManager &manager)  
*Implements the EphemerInterface pure virtual function [ephem\\_initialize](#).*
- virtual void [ephem\\_activate](#) (EphemeridesManager &manager)  
*Mark the model as being activate or inactive.*
- virtual void [ephem\\_build\\_tree](#) (EphemeridesManager &manager)  
*Function, for a particular ephemeris model, nominally builds the particular models portions of the reference frame tree.*
- void [set\\_name](#) (std::string name\_in)

Setter for the name.

- virtual const char \* [get\\_name](#) () const =0

A re-declaration of the pure virtual function in order to convince trick that yes, this is a pure virtual class.

## Data Fields

- bool [active](#)

Is the orientation model actively updating? Defaults to true.

- Planet \* [planet](#)

The planet the attitude model will be working on.

- std::string [name](#)

Name of the planet the attitude model will be working on.

- RefFrameRot \* [planet\\_rot\\_state](#)

The current rotational state of the planet.

- double [planet\\_omega](#)

Nominal axial velocity of the earth.

- EphemerisOrientation [orient\\_interface](#)

The ephemeris interface to the in question orientation.

## Private Member Functions

- [PlanetOrientation](#) & [operator=](#) (const [PlanetOrientation](#) &rhs)

Not implemented.

- [PlanetOrientation](#) (const [PlanetOrientation](#) &rhs)

Not implemented.

## Friends

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

### 8.5.1 Detailed Description

The generic framework for orientation models to interact with a DynManager object.

Definition at line 63 of file planet\_orientation.hh.

### 8.5.2 Constructor & Destructor Documentation

#### 8.5.2.1 [jeod::PlanetOrientation::PlanetOrientation](#) ( void )

Default constructor; constructs a [PlanetOrientation](#) object.

Definition at line 60 of file planet\_orientation.cc.

#### 8.5.2.2 [jeod::PlanetOrientation::~~PlanetOrientation](#) ( void ) [virtual]

Class destructor.

Definition at line 75 of file planet\_orientation.cc.

### 8.5.2.3 jeod::PlanetOrientation::PlanetOrientation ( const PlanetOrientation & rhs ) [private]

Not implemented.

## 8.5.3 Member Function Documentation

### 8.5.3.1 void jeod::PlanetOrientation::activate ( ) [virtual]

Activates the [PlanetOrientation](#) model.

Definition at line 141 of file planet\_orientation.cc.

References [active](#).

### 8.5.3.2 void jeod::PlanetOrientation::deactivate ( ) [virtual]

Deactivates the [PlanetOrientation](#) model.

Definition at line 150 of file planet\_orientation.cc.

References [active](#).

### 8.5.3.3 void jeod::PlanetOrientation::ephem\_activate ( EphemeridesManager & manager ) [virtual]

Mark the model as being activate or inactive.

Parameters

<i>in, out</i>	<i>manager</i>	ephemerides manager
----------------	----------------	---------------------

Definition at line 176 of file planet\_orientation.cc.

### 8.5.3.4 void jeod::PlanetOrientation::ephem\_build\_tree ( EphemeridesManager & manager ) [virtual]

Function, for a particular ephemeris model, nominally builds the particular models portions of the reference frame tree.

This implementation, by default, does nothing

Parameters

<i>in, out</i>	<i>manager</i>	ephemerides manager
----------------	----------------	---------------------

Definition at line 190 of file planet\_orientation.cc.

### 8.5.3.5 void jeod::PlanetOrientation::ephem\_initialize ( EphemeridesManager & manager ) [virtual]

Implements the EphemInterface pure virtual function `ephem_initialize`.

Note that the stand alone "initialize" function MUST be called before this function. Normal operating procedure is the DynamicsManager calls this function, not the user.

Parameters

<i>in, out</i>	<i>manager</i>	ephemerides manager
----------------	----------------	---------------------

Definition at line 165 of file planet\_orientation.cc.

**8.5.3.6** `virtual const char* jeod::PlanetOrientation::get_name ( ) const` `[pure virtual]`

A re-declaration of the pure virtual function in order to convince trick that yes, this is a pure virtual class.

#### Returns

Planet name.

Implemented in [jeod::PlanetRNP](#), [jeod::RNPMars](#), and [jeod::RNPJ2000](#).

**8.5.3.7** `void jeod::PlanetOrientation::initialize ( DynManager & dyn_manager )` `[virtual]`

Goes to the dyn manager given and searches for the planet indicated by the user inputted name, in the given dyn manager.

Will send a fail message if the named planet is not found. Additionally, this will register this ephemeris interface with the given DynManager, and will set up the contained EphemerisOrientation to control the named planet.

#### Parameters

<code>in, out</code>	<code>dyn_manager</code>	DynManager where the attitude will be applied
----------------------	--------------------------	---

Reimplemented in [jeod::RNPMars](#), and [jeod::RNPJ2000](#).

Definition at line 91 of file planet\_orientation.cc.

References `jeod::RNPMessages::initialization_error`, `name`, `orient_interface`, `planet`, `planet_omega`, and `planet_rot_state`.

Referenced by `jeod::RNPJ2000::initialize()`, and `jeod::RNPMars::initialize()`.

**8.5.3.8** `PlanetOrientation& jeod::PlanetOrientation::operator= ( const PlanetOrientation & rhs )` `[private]`

Not implemented.

**8.5.3.9** `void jeod::PlanetOrientation::set_name ( std::string name_in )` `[inline]`

Setter for the name.

Definition at line 136 of file planet\_orientation.hh.

## 8.5.4 Friends And Related Function Documentation

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

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

Definition at line 66 of file planet\_orientation.hh.

## 8.5.5 Field Documentation

**8.5.5.1** `bool jeod::PlanetOrientation::active`

Is the orientation model actively updating? Defaults to true.

`trick_units(-)`

Definition at line 86 of file planet\_orientation.hh.

Referenced by activate(), deactivate(), jeod::RNPNJ2000::ephem\_update(), jeod::RNPMars::ephem\_update(), jeod::RNPNJ2000::update\_axial\_rotation(), jeod::RNPMars::update\_axial\_rotation(), jeod::RNPNJ2000::update\_rnp(), and jeod::RNPMars::update\_rnp().

#### 8.5.5.2 std::string jeod::PlanetOrientation::name

Name of the planet the attitude model will be working on.

Planet must be found in the DynManager sent in at initializationtrick\_units(-)

Definition at line 97 of file planet\_orientation.hh.

Referenced by jeod::RNPNJ2000\_rnp\_j2000\_default\_data::initialize(), jeod::RNPMars\_rnp\_mars\_default\_data::initialize(), and initialize().

#### 8.5.5.3 EphemerisOrientation jeod::PlanetOrientation::orient\_interface

The ephemeris interface to the in question orientation.

trick\_units(-)

Definition at line 130 of file planet\_orientation.hh.

Referenced by jeod::RNPNJ2000::ephem\_update(), jeod::RNPMars::ephem\_update(), and initialize().

#### 8.5.5.4 Planet\* jeod::PlanetOrientation::planet

The planet the attitude model will be working on.

trick\_units(-)

Definition at line 91 of file planet\_orientation.hh.

Referenced by initialize(), jeod::RNPNJ2000::update\_axial\_rotation(), jeod::RNPMars::update\_axial\_rotation(), jeod::RNPNJ2000::update\_rnp(), and jeod::RNPMars::update\_rnp().

#### 8.5.5.5 double jeod::PlanetOrientation::planet\_omega

Nominal axial velocity of the earth.

trick\_units(radian/s)

Definition at line 107 of file planet\_orientation.hh.

Referenced by jeod::RNPNJ2000\_rnp\_j2000\_default\_data::initialize(), jeod::RNPMars\_rnp\_mars\_default\_data::initialize(), initialize(), jeod::RNPNJ2000::initialize(), jeod::RNPMars::initialize(), and jeod::PlanetRNP::propagate\_rnp().

#### 8.5.5.6 RefFrameRot\* jeod::PlanetOrientation::planet\_rot\_state

The current rotational state of the planet.

trick\_units(-)

Definition at line 102 of file planet\_orientation.hh.

Referenced by initialize(), and jeod::PlanetRNP::propagate\_rnp().

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

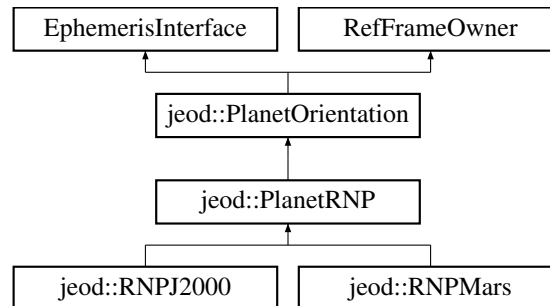
- [planet\\_orientation.hh](#)
- [planet\\_orientation.cc](#)

## 8.6 jeod::PlanetRNP Class Reference

The generic framework for orientation models based on the RNP paradigm.

```
#include <planet_rnp.hh>
```

Inheritance diagram for jeod::PlanetRNP:



### Public Types

- enum [RNPFidelity](#) { [FullRNP](#) = 0, [RotationOnly](#) = 1, [ConstantNP](#) = 2 }
- Specifies the initialization fidelity of the RNP model.*

### Public Member Functions

- [PlanetRNP](#) ()  
*Default constructor; constructs a [PlanetRNP](#) object.*
- virtual [~PlanetRNP](#) ()  
*Class destructor.*
- void [update\\_rnp](#) ()  
*Invokes the calculation for all rotation models contained in the RNP, used on the last time set in each model through [PlanetRotation::set\\_time](#).*
- void [update\\_axial\\_rotation](#) ()  
*Same as [update\\_rnp](#), but only the [axial\\_rotation](#) will be updated.*
- void [propagate\\_rnp](#) ()  
*Multiplies out the (up to) four planet rotation models (nutaton, precession, polar motion and rotation), leaving out models whose pointers are NULL, and feeds it to the planet attitude found in the dyn manager given in initialize.*
- virtual const char \* [get\\_name](#) () const =0  
*A re-declaration of the pure virtual function in order to convince trick that yes, this is a pure virtual class.*

### Data Fields

- [PlanetRotation](#) \* [nutaton](#)  
*Pointer to the nutaton model.*
- [PlanetRotation](#) \* [precession](#)  
*Pointer to the precession model.*
- [PlanetRotation](#) \* [polar\\_motion](#)  
*Pointer to the polar\_motion model.*
- [PlanetRotation](#) \* [rotation](#)  
*Pointer to the rotation model.*
- [RNPFidelity](#) [rnp\\_type](#)  
*The fidelity of the RNP model.*

- bool [enable\\_polar](#)  
*Gives the option of turning on or off polar motion.*
- double [NP\\_matrix](#) [3][3]  
*Rotation Matrix representing:  $\text{transpose}(\text{nutaton} \rightarrow \text{rotation}) * \text{transpose}(\text{precession} \rightarrow \text{rotation})$*

## Protected Attributes

- double [scratch\\_matrix](#) [3][3]  
*A transformation matrix used for intermediate math steps.*

## Private Member Functions

- [PlanetRNP & operator=](#) (const [PlanetRNP](#) &rhs)  
*Not implemented.*
- [PlanetRNP](#) (const [PlanetRNP](#) &rhs)  
*Not implemented.*

## Friends

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

### 8.6.1 Detailed Description

The generic framework for orientation models based on the RNP paradigm.  
Definition at line 71 of file planet\_rnp.hh.

### 8.6.2 Member Enumeration Documentation

#### 8.6.2.1 enum jeod::PlanetRNP::RNPFidelity

Specifies the initialization fidelity of the RNP model.

#### Enumerator

- FullRNP** Full fidelity RNP matrix. Formerly Full\_Term\_RNP.
- RotationOnly** Identity NP matrix, then rotation calculates linearly.
- ConstantNP** Once calculated NP matrix (at the start), then rotation.

Definition at line 82 of file planet\_rnp.hh.

### 8.6.3 Constructor & Destructor Documentation

#### 8.6.3.1 jeod::PlanetRNP::PlanetRNP ( void )

Default constructor; constructs a [PlanetRNP](#) object.  
Definition at line 73 of file planet\_rnp.cc.  
References [NP\\_matrix](#).

### 8.6.3.2 `jeod::PlanetRNP::~~PlanetRNP ( void ) [virtual]`

Class destructor.

Definition at line 89 of file `planet_rnp.cc`.

### 8.6.3.3 `jeod::PlanetRNP::PlanetRNP ( const PlanetRNP & rhs ) [private]`

Not implemented.

## 8.6.4 Member Function Documentation

### 8.6.4.1 `virtual const char* jeod::PlanetRNP::get_name ( ) const [pure virtual]`

A re-declaration of the pure virtual function in order to convince trick that yes, this is a pure virtual class.

#### Returns

Planet name.

Implements [jeod::PlanetOrientation](#).

Implemented in [jeod::RNPMars](#), and [jeod::RNPJ2000](#).

### 8.6.4.2 `PlanetRNP& jeod::PlanetRNP::operator= ( const PlanetRNP & rhs ) [private]`

Not implemented.

### 8.6.4.3 `void jeod::PlanetRNP::propagate_rnp ( void )`

Multiples out the (up to) four planet rotation models (nutation, precession, polar motion and rotation), leaving out models whose pointers are NULL, and feeds it to the planet attitude found in the dyn manager given in initialize.

Definition at line 202 of file `planet_rnp.cc`.

References `ConstantNP`, `enable_polar`, `jeod::RNPMessages::fidelity_error`, `FullRNP`, `NP_matrix`, `jeod::PlanetOrientation::planet_omega`, `jeod::PlanetOrientation::planet_rot_state`, `polar_motion`, `rnp_type`, `jeod::PlanetRotation::rotation`, `rotation`, `RotationOnly`, `scratch_matrix`, and `jeod::RNPMessages::setup_error`.

Referenced by `update_axial_rotation()`, and `update_rnp()`.

### 8.6.4.4 `void jeod::PlanetRNP::update_axial_rotation ( void )`

Same as `update_rnp`, but only the `axial_rotation` will be updated.

Definition at line 174 of file `planet_rnp.cc`.

References `propagate_rnp()`, `rotation`, `jeod::RNPMessages::setup_error`, and `jeod::PlanetRotation::update_rotation()`.

Referenced by `jeod::RNPJ2000::ephem_update()`, `jeod::RNPMars::ephem_update()`, `jeod::RNPJ2000::update_axial_rotation()`, and `jeod::RNPMars::update_axial_rotation()`.

### 8.6.4.5 `void jeod::PlanetRNP::update_rnp ( void )`

Invokes the calculation for all rotation models contained in the RNP, used on the last time set in each model through `PlanetRotation::set_time`.



Then multiplies out and updates the attitude of the planet found during initialization.

Definition at line 103 of file planet\_rnp.cc.

References enable\_polar, jeod::RNPMessages::fidelity\_error, FullRNP, NP\_matrix, nutation, polar\_motion, precession, propagate\_rnp(), rnp\_type, jeod::PlanetRotation::rotation, rotation, jeod::RNPMessages::setup\_error, and jeod::PlanetRotation::update\_rotation().

Referenced by jeod::RNPJ2000::update\_rnp(), and jeod::RNPMars::update\_rnp().

## 8.6.5 Friends And Related Function Documentation

**8.6.5.1** void init\_attrjeod\_PlanetRNP ( ) [friend]

**8.6.5.2** friend class InputProcessor [friend]

Definition at line 73 of file planet\_rnp.hh.

## 8.6.6 Field Documentation

**8.6.6.1** bool jeod::PlanetRNP::enable\_polar

Gives the option of turning on or off polar motion.

trick\_units(-)

Definition at line 120 of file planet\_rnp.hh.

Referenced by jeod::RNPJ2000\_rnp\_j2000\_default\_data::initialize(), jeod::RNPMars\_rnp\_mars\_default\_data::initialize(), propagate\_rnp(), jeod::RNPMars::RNPMars(), jeod::RNPJ2000::update\_rnp(), and update\_rnp().

**8.6.6.2** double jeod::PlanetRNP::NP\_matrix[3][3]

Rotation Matrix representing: transpose(nutation->rotation) \* transpose(precession->rotation)

trick\_units(-)

Definition at line 126 of file planet\_rnp.hh.

Referenced by PlanetRNP(), propagate\_rnp(), and update\_rnp().

**8.6.6.3** PlanetRotation\* jeod::PlanetRNP::nutation

Pointer to the nutation model.

trick\_units(-)

Definition at line 99 of file planet\_rnp.hh.

Referenced by jeod::RNPJ2000::RNPJ2000(), jeod::RNPMars::RNPMars(), jeod::RNPJ2000::update\_rnp(), update\_rnp(), and jeod::RNPMars::update\_rnp().

**8.6.6.4** PlanetRotation\* jeod::PlanetRNP::polar\_motion

Pointer to the polar\_motion model.

trick\_units(-)

Definition at line 107 of file planet\_rnp.hh.

Referenced by propagate\_rnp(), jeod::RNPJ2000::RNPJ2000(), jeod::RNPMars::RNPMars(), jeod::RNPJ2000::update\_rnp(), and update\_rnp().

#### 8.6.6.5 PlanetRotation\* jeod::PlanetRNP::precession

Pointer to the precession model.

trick\_units(-)

Definition at line 103 of file planet\_rnp.hh.

Referenced by jeod::RNPJ2000::RNPJ2000(), jeod::RNPMars::RNPMars(), jeod::RNPJ2000::update\_rnp(), update\_rnp(), and jeod::RNPMars::update\_rnp().

#### 8.6.6.6 RNPFidelity jeod::PlanetRNP::rnp\_type

The fidelity of the RNP model.

trick\_units(-)

Definition at line 116 of file planet\_rnp.hh.

Referenced by jeod::RNPJ2000\_rnp\_j2000\_default\_data::initialize(), jeod::RNPMars\_rnp\_mars\_default\_data::initialize(), jeod::RNPJ2000::initialize(), propagate\_rnp(), jeod::RNPJ2000::update\_axial\_rotation(), jeod::RNPMars::update\_axial\_rotation(), jeod::RNPJ2000::update\_rnp(), update\_rnp(), and jeod::RNPMars::update\_rnp().

#### 8.6.6.7 PlanetRotation\* jeod::PlanetRNP::rotation

Pointer to the rotation model.

trick\_units(-)

Definition at line 111 of file planet\_rnp.hh.

Referenced by propagate\_rnp(), jeod::RNPJ2000::RNPJ2000(), jeod::RNPMars::RNPMars(), jeod::RNPJ2000::update\_axial\_rotation(), update\_axial\_rotation(), jeod::RNPMars::update\_axial\_rotation(), jeod::RNPJ2000::update\_rnp(), update\_rnp(), and jeod::RNPMars::update\_rnp().

#### 8.6.6.8 double jeod::PlanetRNP::scratch\_matrix[3][3] [protected]

A transformation matrix used for intermediate math steps.

trick\_units(-)

Definition at line 133 of file planet\_rnp.hh.

Referenced by propagate\_rnp().

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

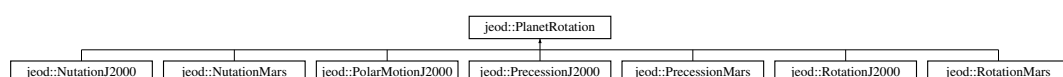
- [planet\\_rnp.hh](#)
- [planet\\_rnp.cc](#)

## 8.7 jeod::PlanetRotation Class Reference

The generic base class for all planet transformations.

```
#include <planet_rotation.hh>
```

Inheritance diagram for jeod::PlanetRotation:



## Public Member Functions

- [PlanetRotation](#) ()  
*Constructor, initialize base level data.*
- virtual [~PlanetRotation](#) ()  
*Destructor.*
- virtual void [update\\_rotation](#) (void)  
*Updates the rotation of this particular object, based on the last time set in "set\_time".*
- virtual void [initialize](#) ([PlanetRotationInit](#) \*init)  
*Initializes the invoking object from an initialization object.*
- virtual void [update\\_time](#) (double time)  
*Update the time that the next update\_rotation call will use.*
- void [get\\_rotation](#) (double rot[3][3])  
*Copy the last calculated rotation to 'rot'.*
- void [get\\_rotation\\_transpose](#) (double rot[3][3])  
*Same as get\_rotation, but returns the transpose.*

## Data Fields

- double [rotation](#) [3][3]  
*The transformation matrix outputted by the model after update is called.*
- double [current\\_time](#)  
*The current time the transformation matrix will be calculated from.*

## Protected Attributes

- double [RADTODEG](#)  
*Convert from radians to degrees.*
- double [DEGTORAD](#)  
*Inverse of RADTODEG.*
- double [DEGTOSEC](#)  
*Convert from degrees to arcseconds.*
- double [SECTODEG](#)  
*Inverse of DEGTOSEC.*
- double [JULIANDAYTOSEC](#)  
*Convert from julian day to seconds.*
- double [SECTOJULIANDAY](#)  
*Inverse of JULIANDAYTOSEC.*
- double [JULIANCENTTODAY](#)  
*Convert from julian centuries to julian days.*
- double [DAYTOJULIANCENT](#)  
*Inverse of JULIANCENTTODAY.*

## Private Member Functions

- [PlanetRotation](#) & [operator=](#) (const [PlanetRotation](#) &rhs)
- [PlanetRotation](#) (const [PlanetRotation](#) &rhs)

## Friends

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

### 8.7.1 Detailed Description

The generic base class for all planet transformations.

Definition at line 57 of file planet\_rotation.hh.

### 8.7.2 Constructor & Destructor Documentation

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

Constructor, initialize base level data.

Definition at line 54 of file planet\_rotation.cc.

References DAYTOJULIANCENT, DEGTORAD, DEGTOSEC, JULIANCENTTODAY, JULIANDAYTOSEC, RADTODEG, rotation, SECTODEG, and SECTOJULIANDAY.

#### 8.7.2.2 `jeod::PlanetRotation::~~PlanetRotation ( void )` `[virtual]`

Destructor.

Definition at line 77 of file planet\_rotation.cc.

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

### 8.7.3 Member Function Documentation

#### 8.7.3.1 `void jeod::PlanetRotation::get_rotation ( double rot[3][3] )`

Copy the last calculated rotation to 'rot'.

Parameters

<code>out</code>	<code>rot</code>	Where the rotation matrix will be stored
------------------	------------------	--

Definition at line 99 of file planet\_rotation.cc.

References rotation.

#### 8.7.3.2 `void jeod::PlanetRotation::get_rotation_transpose ( double rot[3][3] )`

Same as `get_rotation`, but returns the transpose.

Parameters

<code>out</code>	<code>rot</code>	Where the transpose matrix will be stored
------------------	------------------	---

Definition at line 111 of file planet\_rotation.cc.

References rotation.

#### 8.7.3.3 `virtual void jeod::PlanetRotation::initialize ( PlanetRotationInit * init )` `[inline]`, `[virtual]`

Initializes the invoking object from an initialization object.

## Parameters

<i>in</i>	<i>init</i>	The object that will initialize the invoking object.
-----------	-------------	--

Reimplemented in [jeod::NutationJ2000](#), and [jeod::PolarMotionJ2000](#).

Definition at line 155 of file planet\_rotation.hh.

**8.7.3.4 PlanetRotation& jeod::PlanetRotation::operator= ( const PlanetRotation & rhs )** [private]

**8.7.3.5 virtual void jeod::PlanetRotation::update\_rotation ( void )** [inline],[virtual]

Updates the rotation of this particular object, based on the last time set in "set\_time".

Reimplemented in [jeod::NutationJ2000](#), [jeod::NutationMars](#), [jeod::PrecessionMars](#), [jeod::RotationJ2000](#), [jeod::RotationMars](#), [jeod::PolarMotionJ2000](#), and [jeod::PrecessionJ2000](#).

Definition at line 131 of file planet\_rotation.hh.

Referenced by [jeod::PlanetRNP::update\\_axial\\_rotation\(\)](#), and [jeod::PlanetRNP::update\\_rnp\(\)](#).

**8.7.3.6 void jeod::PlanetRotation::update\_time ( double time )** [virtual]

Update the time that the next update\_rotation call will use.

## Parameters

<i>in</i>	<i>time</i>	Time. units and UTC/UT1/TAI etc are determined by the individual <a href="#">Planet-Rotation</a> model
-----------	-------------	--

Definition at line 87 of file planet\_rotation.cc.

References [current\\_time](#).

Referenced by [jeod::RNPJ2000::update\\_axial\\_rotation\(\)](#), [jeod::RNPMars::update\\_axial\\_rotation\(\)](#), [jeod::RNPJ2000::update\\_rnp\(\)](#), and [jeod::RNPMars::update\\_rnp\(\)](#).

## 8.7.4 Friends And Related Function Documentation

**8.7.4.1 void init\_attrjeod\_\_PlanetRotation ( )** [friend]

**8.7.4.2 friend class InputProcessor** [friend]

Definition at line 59 of file planet\_rotation.hh.

## 8.7.5 Field Documentation

**8.7.5.1 double jeod::PlanetRotation::current\_time**

The current time the transformation matrix will be calculated from.

Unitless so that models with different time scales can be used with one underlying modeltrick\_units(–)

Definition at line 74 of file planet\_rotation.hh.

Referenced by [jeod::PrecessionJ2000::update\\_rotation\(\)](#), [jeod::PolarMotionJ2000::update\\_rotation\(\)](#), [jeod::RotationMars::update\\_rotation\(\)](#), [jeod::RotationJ2000::update\\_rotation\(\)](#), [jeod::PrecessionMars::update\\_rotation\(\)](#), [jeod::NutationMars::update\\_rotation\(\)](#), [jeod::NutationJ2000::update\\_rotation\(\)](#), and [update\\_time\(\)](#).

**8.7.5.2 double jeod::PlanetRotation::DAYTOJULIANCENT [protected]**

Inverse of JULIANCENTTODAY.

trick\_units(-)

Definition at line 112 of file planet\_rotation.hh.

Referenced by PlanetRotation().

**8.7.5.3 double jeod::PlanetRotation::DEGTORAD [protected]**

Inverse of RADTODEG.

trick\_units(radian/degree)

Definition at line 88 of file planet\_rotation.hh.

Referenced by PlanetRotation(), jeod::PrecessionJ2000::update\_rotation(), jeod::RotationJ2000::update\_rotation(), and jeod::NutationJ2000::update\_rotation().

**8.7.5.4 double jeod::PlanetRotation::DEGTOSEC [protected]**

Convert from degrees to arcseconds.

trick\_units(arcsecond/degree)

Definition at line 92 of file planet\_rotation.hh.

Referenced by PlanetRotation(), and jeod::PrecessionJ2000::update\_rotation().

**8.7.5.5 double jeod::PlanetRotation::JULIANCENTTODAY [protected]**

Convert from julian centuries to julian days.

trick\_units(-)

Definition at line 108 of file planet\_rotation.hh.

Referenced by PlanetRotation().

**8.7.5.6 double jeod::PlanetRotation::JULIANDAYTOSEC [protected]**

Convert from julian day to seconds.

trick\_units(s/day)

Definition at line 100 of file planet\_rotation.hh.

Referenced by PlanetRotation().

**8.7.5.7 double jeod::PlanetRotation::RADTODEG [protected]**

Convert from radians to degrees.

trick\_units(degree/radian)

Definition at line 84 of file planet\_rotation.hh.

Referenced by PlanetRotation().

## 8.7.5.8 double jeod::PlanetRotation::rotation[3][3]

The transformation matrix outputted by the model after update is called.

trick\_units(—)

Definition at line 67 of file planet\_rotation.hh.

Referenced by get\_rotation(), get\_rotation\_transpose(), PlanetRotation(), jeod::PlanetRNP::propagate\_rnp(), jeod::PlanetRNP::update\_rnp(), jeod::PrecessionJ2000::update\_rotation(), jeod::PolarMotionJ2000::update\_rotation(), jeod::RotationMars::update\_rotation(), jeod::RotationJ2000::update\_rotation(), jeod::PrecessionMars::update\_rotation(), jeod::NutationMars::update\_rotation(), and jeod::NutationJ2000::update\_rotation().

## 8.7.5.9 double jeod::PlanetRotation::SECTODEG [protected]

Inverse of DEGTOSEC.

trick\_units(degree/arcsecond)

Definition at line 96 of file planet\_rotation.hh.

Referenced by PlanetRotation().

## 8.7.5.10 double jeod::PlanetRotation::SECTOJULIANDAY [protected]

Inverse of JULIANDAYTOSEC.

trick\_units(day/s)

Definition at line 104 of file planet\_rotation.hh.

Referenced by PlanetRotation().

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

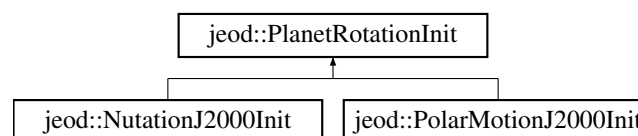
- [planet\\_rotation.hh](#)
- [planet\\_rotation.cc](#)

## 8.8 jeod::PlanetRotationInit Class Reference

The generic base class for all initializing classes for [PlanetRotation](#) derived classes.

```
#include <planet_rotation_init.hh>
```

Inheritance diagram for jeod::PlanetRotationInit:



### Public Member Functions

- [PlanetRotationInit](#) ()  
*constructor*
- virtual [~PlanetRotationInit](#) ()  
*destructor*

## Private Member Functions

- [PlanetRotationInit](#) & [operator=](#) (const [PlanetRotationInit](#) &rhs)
- [PlanetRotationInit](#) (const [PlanetRotationInit](#) &rhs)

## Friends

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

### 8.8.1 Detailed Description

The generic base class for all initializing classes for [PlanetRotation](#) derived classes.

Definition at line 52 of file [planet\\_rotation\\_init.hh](#).

### 8.8.2 Constructor & Destructor Documentation

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

constructor

Definition at line 47 of file [planet\\_rotation\\_init.cc](#).

#### 8.8.2.2 `jeod::PlanetRotationInit::~~PlanetRotationInit ( void )` `[virtual]`

destructor

Definition at line 56 of file [planet\\_rotation\\_init.cc](#).

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

### 8.8.3 Member Function Documentation

#### 8.8.3.1 `PlanetRotationInit& jeod::PlanetRotationInit::operator= ( const PlanetRotationInit & rhs )` `[private]`

### 8.8.4 Friends And Related Function Documentation

#### 8.8.4.1 `void init_attrjeod__PlanetRotationInit ( )` `[friend]`

#### 8.8.4.2 `friend class InputProcessor` `[friend]`

Definition at line 54 of file [planet\\_rotation\\_init.hh](#).

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

- [planet\\_rotation\\_init.hh](#)
- [planet\\_rotation\\_init.cc](#)

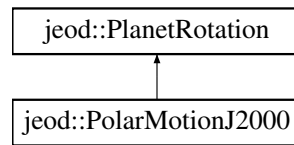
## 8.9 jeod::PolarMotionJ2000 Class Reference

Implements the polar motion portion of the J2000 RNP.

```
#include <polar_motion_j2000.hh>
```



Inheritance diagram for jeod::PolarMotionJ2000:



## Public Member Functions

- [PolarMotionJ2000](#) ()  
*Default constructor.*
- virtual [~PolarMotionJ2000](#) ()  
*Destructor.*
- virtual void [update\\_rotation](#) ()  
*[PolarMotionJ2000](#) specific implementation of [update\\_rotation](#) from [PlanetRotation](#).*
- virtual void [initialize](#) ([PlanetRotationInit](#) \*init)  
*Initialize the coefficients of [PolarMotionJ2000](#).*

## Data Fields

- double [xp](#)  
*Current X Polar coordinate.*
- double [yp](#)  
*Current Y Polar coordinate.*
- double \* [xp\\_tbl](#)  
*X Polar coordinate table.*
- double \* [yp\\_tbl](#)  
*Y Polar coordinate table.*
- double \* [polar\\_mjd](#)  
*Independent variable for the XY coordinate table.*
- bool [override\\_table](#)  
*If true, do no table lookup and use the currently set xp and yp.*
- unsigned int [last\\_table\\_index](#)  
*Size - 1 of [xp\\_tbl](#), [yp\\_tbl](#) and [polar\\_mjd](#) (last index)*
- bool [warn\\_table](#)  
*Have we warned about being off the table end?*

## Private Member Functions

- [PolarMotionJ2000](#) & [operator=](#) (const [PolarMotionJ2000](#) &rhs)
- [PolarMotionJ2000](#) (const [PolarMotionJ2000](#) &rhs)

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_PolarMotionJ2000](#) ()

## Additional Inherited Members

### 8.9.1 Detailed Description

Implements the polar motion portion of the J2000 RNP.

Definition at line 58 of file polar\_motion\_j2000.hh.

### 8.9.2 Constructor & Destructor Documentation

#### 8.9.2.1 jeod::PolarMotionJ2000::PolarMotionJ2000 ( void )

Default constructor.

Initializes all data.

Definition at line 68 of file polar\_motion\_j2000.cc.

#### 8.9.2.2 jeod::PolarMotionJ2000::~~PolarMotionJ2000 ( void ) [virtual]

Destructor.

Definition at line 86 of file polar\_motion\_j2000.cc.

References polar\_mjd, xp\_tbl, and yp\_tbl.

#### 8.9.2.3 jeod::PolarMotionJ2000::PolarMotionJ2000 ( const PolarMotionJ2000 & rhs ) [private]

### 8.9.3 Member Function Documentation

#### 8.9.3.1 void jeod::PolarMotionJ2000::initialize ( PlanetRotationInit \* init ) [virtual]

Initialize the coefficients of [PolarMotionJ2000](#).

init must be of type [PolarMotionJ2000Init](#) or a fail message will occur.

Parameters

in	init	<a href="#">PolarMotionJ2000Init</a> with needed coefficients
----	------	---

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

Definition at line 214 of file polar\_motion\_j2000.cc.

References jeod::RNPMessages::initialization\_error, last\_table\_index, jeod::PolarMotionJ2000Init::last\_table\_index, override\_table, jeod::PolarMotionJ2000Init::override\_table, polar\_mjd, jeod::PolarMotionJ2000Init::polar\_mjd, xp, jeod::PolarMotionJ2000Init::xp, xp\_tbl, jeod::PolarMotionJ2000Init::xp\_tbl, yp, jeod::PolarMotionJ2000Init::yp, yp\_tbl, and jeod::PolarMotionJ2000Init::yp\_tbl.

#### 8.9.3.2 PolarMotionJ2000& jeod::PolarMotionJ2000::operator= ( const PolarMotionJ2000 & rhs ) [private]

#### 8.9.3.3 void jeod::PolarMotionJ2000::update\_rotation ( void ) [virtual]

[PolarMotionJ2000](#) specific implementaiton of update\_rotation from [PlanetRotation](#).

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

Definition at line 109 of file polar\_motion\_j2000.cc.

References jeod::PlanetRotation::current\_time, last\_table\_index, override\_table, polar\_mjd, jeod::RNPMessages::polar\_motion\_table\_warning, jeod::PlanetRotation::rotation, warn\_table, xp, xp\_tbl, yp, and yp\_tbl.

### 8.9.4 Friends And Related Function Documentation

8.9.4.1 void init\_attrjeod\_\_PolarMotionJ2000 ( ) [friend]

8.9.4.2 friend class InputProcessor [friend]

Definition at line 60 of file polar\_motion\_j2000.hh.

### 8.9.5 Field Documentation

8.9.5.1 unsigned int jeod::PolarMotionJ2000::last\_table\_index

Size - 1 of xp\_tbl, yp\_tbl and polar\_mjd (last index)

trick\_units(count)

Definition at line 92 of file polar\_motion\_j2000.hh.

Referenced by initialize(), and update\_rotation().

8.9.5.2 bool jeod::PolarMotionJ2000::override\_table

If true, do no table lookup and use the currently set xp and yp.

trick\_units(-)

Definition at line 88 of file polar\_motion\_j2000.hh.

Referenced by initialize(), and update\_rotation().

8.9.5.3 double\* jeod::PolarMotionJ2000::polar\_mjd

Independent variable for the XY coordinate table.

trick\_units(-)

Definition at line 83 of file polar\_motion\_j2000.hh.

Referenced by initialize(), update\_rotation(), and ~PolarMotionJ2000().

8.9.5.4 bool jeod::PolarMotionJ2000::warn\_table

Have we warned about being off the table end?

trick\_units(-)

Definition at line 96 of file polar\_motion\_j2000.hh.

Referenced by update\_rotation().

8.9.5.5 double jeod::PolarMotionJ2000::xp

Current X Polar coordinate.

trick\_units(radian)

Definition at line 67 of file polar\_motion\_j2000.hh.

Referenced by initialize(), and update\_rotation().

#### 8.9.5.6 `double* jeod::PolarMotionJ2000::xp_tbl`

X Polar coordinate table.

`trick_units(radian)`

Definition at line 75 of file `polar_motion_j2000.hh`.

Referenced by `initialize()`, `update_rotation()`, and `~PolarMotionJ2000()`.

#### 8.9.5.7 `double jeod::PolarMotionJ2000::yp`

Current Y Polar coordinate.

`trick_units(radian)`

Definition at line 71 of file `polar_motion_j2000.hh`.

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

#### 8.9.5.8 `double* jeod::PolarMotionJ2000::yp_tbl`

Y Polar coordinate table.

`trick_units(radian)`

Definition at line 79 of file `polar_motion_j2000.hh`.

Referenced by `initialize()`, `update_rotation()`, and `~PolarMotionJ2000()`.

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

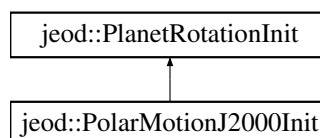
- [polar\\_motion\\_j2000.hh](#)
- [polar\\_motion\\_j2000.cc](#)

## 8.10 `jeod::PolarMotionJ2000Init` Class Reference

Initializes the [PolarMotionJ2000](#) object.

```
#include <polar_motion_j2000_init.hh>
```

Inheritance diagram for `jeod::PolarMotionJ2000Init`:



### Public Member Functions

- [PolarMotionJ2000Init](#) ()  
*constructor.*
- virtual [~PolarMotionJ2000Init](#) ()  
*Destructor.*

## Data Fields

- double [xp](#)  
*Current X Polar coordinate.*
- double [yp](#)  
*Current Y Polar coordinate.*
- double \* [xp\\_tbl](#)  
*X Polar coordinate table.*
- double \* [yp\\_tbl](#)  
*Y Polar coordinate table.*
- double \* [polar\\_mjd](#)  
*Independent variable for the XY coordinate table.*
- bool [override\\_table](#)  
*If true, do no table lookup and use the currently set xp and yp.*
- unsigned int [last\\_table\\_index](#)  
*Size - 1 of xp\_tbl, yp\_tbl and polar\_mjd (last index)*

## Private Member Functions

- [PolarMotionJ2000Init & operator=](#) (const [PolarMotionJ2000Init](#) &rhs)
- [PolarMotionJ2000Init](#) (const [PolarMotionJ2000Init](#) &rhs)

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_PolarMotionJ2000Init](#) ()

### 8.10.1 Detailed Description

Initializes the [PolarMotionJ2000](#) object.

Definition at line 58 of file polar\_motion\_j2000\_init.hh.

### 8.10.2 Constructor & Destructor Documentation

#### 8.10.2.1 jeod::PolarMotionJ2000Init::PolarMotionJ2000Init ( void )

constructor.

Initializes all data.

Definition at line 58 of file polar\_motion\_j2000\_init.cc.

#### 8.10.2.2 jeod::PolarMotionJ2000Init::~~PolarMotionJ2000Init ( void ) [virtual]

Destructor.

Definition at line 75 of file polar\_motion\_j2000\_init.cc.

References [polar\\_mjd](#), [xp\\_tbl](#), and [yp\\_tbl](#).

8.10.2.3 `jeod::PolarMotionJ2000Init::PolarMotionJ2000Init ( const PolarMotionJ2000Init & rhs ) [private]`

### 8.10.3 Member Function Documentation

8.10.3.1 `PolarMotionJ2000Init& jeod::PolarMotionJ2000Init::operator= ( const PolarMotionJ2000Init & rhs ) [private]`

### 8.10.4 Friends And Related Function Documentation

8.10.4.1 `void init_attrjeod__PolarMotionJ2000Init ( ) [friend]`

8.10.4.2 `friend class InputProcessor [friend]`

Definition at line 60 of file `polar_motion_j2000_init.hh`.

### 8.10.5 Field Documentation

8.10.5.1 `unsigned int jeod::PolarMotionJ2000Init::last_table_index`

Size - 1 of `xp_tbl`, `yp_tbl` and `polar_mjd` (last index)

`trick_units(count)`

Definition at line 94 of file `polar_motion_j2000_init.hh`.

Referenced by `jeod::PolarMotionJ2000Init_xpyp_daily_default_data::initialize()`, `jeod::PolarMotionJ2000Init_xpyp_monthly_default_data::initialize()`, and `jeod::PolarMotionJ2000::initialize()`.

8.10.5.2 `bool jeod::PolarMotionJ2000Init::override_table`

If true, do no table lookup and use the currently set `xp` and `yp`.

`trick_units(-)`

Definition at line 90 of file `polar_motion_j2000_init.hh`.

Referenced by `jeod::PolarMotionJ2000Init_xpyp_daily_default_data::initialize()`, `jeod::PolarMotionJ2000Init_xpyp_monthly_default_data::initialize()`, and `jeod::PolarMotionJ2000::initialize()`.

8.10.5.3 `double* jeod::PolarMotionJ2000Init::polar_mjd`

Independent variable for the XY coordinate table.

`trick_units(-)`

Definition at line 85 of file `polar_motion_j2000_init.hh`.

Referenced by `jeod::PolarMotionJ2000Init_xpyp_daily_default_data::initialize()`, `jeod::PolarMotionJ2000Init_xpyp_monthly_default_data::initialize()`, `jeod::PolarMotionJ2000::initialize()`, and `~PolarMotionJ2000Init()`.

8.10.5.4 `double jeod::PolarMotionJ2000Init::xp`

Current X Polar coordinate.

`trick_units(radian)`

Definition at line 69 of file `polar_motion_j2000_init.hh`.

Referenced by `jeod::PolarMotionJ2000::initialize()`.

**8.10.5.5 double\* jeod::PolarMotionJ2000Init::xp\_tbl**

X Polar coordinate table.

trick\_units(radian)

Definition at line 77 of file polar\_motion\_j2000\_init.hh.

Referenced by jeod::PolarMotionJ2000Init\_xpyp\_daily\_default\_data::initialize(), jeod::PolarMotionJ2000Init\_xpyp\_monthly\_default\_data::initialize(), jeod::PolarMotionJ2000::initialize(), and ~PolarMotionJ2000Init().

**8.10.5.6 double jeod::PolarMotionJ2000Init::yp**

Current Y Polar coordinate.

trick\_units(radian)

Definition at line 73 of file polar\_motion\_j2000\_init.hh.

Referenced by jeod::PolarMotionJ2000::initialize().

**8.10.5.7 double\* jeod::PolarMotionJ2000Init::yp\_tbl**

Y Polar coordinate table.

trick\_units(radian)

Definition at line 81 of file polar\_motion\_j2000\_init.hh.

Referenced by jeod::PolarMotionJ2000Init\_xpyp\_daily\_default\_data::initialize(), jeod::PolarMotionJ2000Init\_xpyp\_monthly\_default\_data::initialize(), jeod::PolarMotionJ2000::initialize(), and ~PolarMotionJ2000Init().

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

- [polar\\_motion\\_j2000\\_init.hh](#)
- [polar\\_motion\\_j2000\\_init.cc](#)

**8.11 jeod::PolarMotionJ2000Init\_xpyp\_daily\_default\_data Class Reference**

```
#include <xpyp_daily.hh>
```

**Public Member Functions**

- void [initialize](#) ([PolarMotionJ2000Init \\*](#))

**8.11.1 Detailed Description**

Definition at line 18 of file xpyp\_daily.hh.

**8.11.2 Member Function Documentation****8.11.2.1 void jeod::PolarMotionJ2000Init\_xpyp\_daily\_default\_data::initialize ( [PolarMotionJ2000Init \\*](#) [PolarMotionJ2000Init\\_ptr](#) )**

Definition at line 38 of file xpyp\_daily.cc.

References [jeod::PolarMotionJ2000Init::last\\_table\\_index](#), [jeod::PolarMotionJ2000Init::override\\_table](#), [jeod::PolarMotionJ2000Init::polar\\_mjd](#), [jeod::PolarMotionJ2000Init::xp\\_tbl](#), and [jeod::PolarMotionJ2000Init::yp\\_tbl](#).

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

- [xpyp\\_daily.hh](#)
- [xpyp\\_daily.cc](#)

## 8.12 jeod::PolarMotionJ2000Init\_xpyp\_monthly\_default\_data Class Reference

```
#include <xpyp_monthly.hh>
```

### Public Member Functions

- void [initialize](#) ([PolarMotionJ2000Init](#) \*)

### 8.12.1 Detailed Description

Definition at line 19 of file xpyp\_monthly.hh.

### 8.12.2 Member Function Documentation

8.12.2.1 void [jeod::PolarMotionJ2000Init\\_xpyp\\_monthly\\_default\\_data::initialize](#) ( [PolarMotionJ2000Init](#) \* [PolarMotionJ2000Init\\_ptr](#) )

Definition at line 38 of file xpyp\_monthly.cc.

References [jeod::PolarMotionJ2000Init::last\\_table\\_index](#), [jeod::PolarMotionJ2000Init::override\\_table](#), [jeod::PolarMotionJ2000Init::polar\\_mjd](#), [jeod::PolarMotionJ2000Init::xp\\_tbl](#), and [jeod::PolarMotionJ2000Init::yp\\_tbl](#).

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

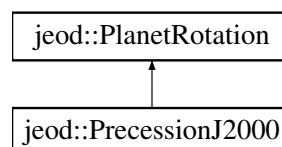
- [xpyp\\_monthly.hh](#)
- [xpyp\\_monthly.cc](#)

## 8.13 jeod::PrecessionJ2000 Class Reference

Implements the precession portion of the J2000 RNP.

```
#include <precession_j2000.hh>
```

Inheritance diagram for [jeod::PrecessionJ2000](#):



### Public Member Functions

- [PrecessionJ2000](#) ()  
*constructor*
- virtual [~PrecessionJ2000](#) ()  
*destructor*



- virtual void [update\\_rotation](#) ()

*Precession J2000 specific implementation of update\_rotation, inherited from Planetrotation.*

## Private Member Functions

- [PrecessionJ2000](#) & [operator=](#) (const [PrecessionJ2000](#) &rhs)
- [PrecessionJ2000](#) (const [PrecessionJ2000](#) &rhs)

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_PrecessionJ2000](#) ()

## Additional Inherited Members

### 8.13.1 Detailed Description

Implements the precession portion of the J2000 RNP.

Definition at line 60 of file `precession_j2000.hh`.

### 8.13.2 Constructor & Destructor Documentation

#### 8.13.2.1 `jeod::PrecessionJ2000::PrecessionJ2000 ( void )`

constructor

Definition at line 54 of file `precession_j2000.cc`.

#### 8.13.2.2 `jeod::PrecessionJ2000::~~PrecessionJ2000 ( void )` `[virtual]`

destructor

Definition at line 62 of file `precession_j2000.cc`.

#### 8.13.2.3 `jeod::PrecessionJ2000::PrecessionJ2000 ( const PrecessionJ2000 & rhs )` `[private]`

### 8.13.3 Member Function Documentation

#### 8.13.3.1 `PrecessionJ2000& jeod::PrecessionJ2000::operator= ( const PrecessionJ2000 & rhs )` `[private]`

#### 8.13.3.2 `void jeod::PrecessionJ2000::update_rotation ( void )` `[virtual]`

Precession J2000 specific implementation of update\_rotation, inherited from Planetrotation.

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

Definition at line 72 of file `precession_j2000.cc`.

References `jeod::PlanetRotation::current_time`, `jeod::PlanetRotation::DEGTORAD`, `jeod::PlanetRotation::DEGTOSEC`, and `jeod::PlanetRotation::rotation`.

### 8.13.4 Friends And Related Function Documentation

8.13.4.1 `void init_attrjeod__PrecessionJ2000 ( ) [friend]`

8.13.4.2 `friend class InputProcessor [friend]`

Definition at line 62 of file `precession_j2000.hh`.

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

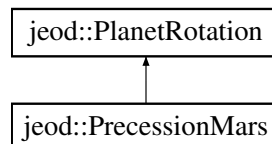
- [precession\\_j2000.hh](#)
- [precession\\_j2000.cc](#)

## 8.14 jeod::PrecessionMars Class Reference

Implements the axial rotation portion of the "Pathfinder" Mars RNP model.

```
#include <precession_mars.hh>
```

Inheritance diagram for `jeod::PrecessionMars`:



### Public Member Functions

- [PrecessionMars](#) ()  
*Default constructor.*
- virtual [~PrecessionMars](#) ()  
*Destructor.*
- virtual void [update\\_rotation](#) ()  
*PrecessionMars specific implementation of update\_rotation, to calculate precession.*
- void [compute\\_fixed\\_matrices](#) ()  
*Calculate constant rotation matrices resulting from N and J.*

### Data Fields

- [NutationMars](#) \* [nutaton](#)  
*Pointer to the Mars nutation object, used to access nutation in longitude information.*
- double [psi\\_at\\_j2000](#)  
*The (constant) precession angle of Mars at the J2000 epoch.*
- double [psi\\_dot](#)  
*The (constant) simple precession rate of Mars.*
- double [psi\\_precess](#)  
*The most recent calculated value of the precession angle for Mars, measured since J2000 epoch.*
- double [N](#)  
*The (constant) angle from the J2000 vernal equinox to the node of the Mars mean orbit and ICRF x-y plane.*
- double [J](#)  
*The (constant) inclination of the Mars mean orbit relative to the ICRF x-y plane.*

## Private Member Functions

- [PrecessionMars](#) & operator= (const [PrecessionMars](#) &rhs)
- [PrecessionMars](#) (const [PrecessionMars](#) &rhs)

## Private Attributes

- double [NJ\\_matrix](#) [3][3]

*The (constant) rotation matrix calculated from N and J.*

## Friends

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

## Additional Inherited Members

### 8.14.1 Detailed Description

Implements the axial rotation portion of the "Pathfinder" Mars RNP model.

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

### 8.14.2 Constructor & Destructor Documentation

#### 8.14.2.1 jeod::PrecessionMars::PrecessionMars ( void )

Default constructor.

Definition at line 63 of file `precession_mars.cc`.

References `NJ_matrix`.

#### 8.14.2.2 jeod::PrecessionMars::~~PrecessionMars ( void ) [virtual]

Destructor.

Definition at line 80 of file `precession_mars.cc`.

#### 8.14.2.3 jeod::PrecessionMars::PrecessionMars ( const [PrecessionMars](#) & rhs ) [private]

### 8.14.3 Member Function Documentation

#### 8.14.3.1 void jeod::PrecessionMars::compute\_fixed\_matrices ( void )

Calculate constant rotation matrices resulting from N and J.

Definition at line 142 of file `precession_mars.cc`.

References `J`, `N`, and `NJ_matrix`.

Referenced by `jeod::RNPMars::initialize()`.

**8.14.3.2** `PrecessionMars& jeod::PrecessionMars::operator= ( const PrecessionMars & rhs )` `[private]`

**8.14.3.3** `void jeod::PrecessionMars::update_rotation ( void )` `[virtual]`

[PrecessionMars](#) specific implementation of `update_rotation`, to calculate precession.

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

Definition at line 92 of file `precession_mars.cc`.

References `jeod::PlanetRotation::current_time`, `NJ_matrix`, `nutaton`, `jeod::NutationMars::nutaton_in_longitude`, `psi_at_j2000`, `psi_dot`, `psi_precess`, `jeod::PlanetRotation::rotation`, and `jeod::RNPMessages::setup_error`.

## 8.14.4 Friends And Related Function Documentation

**8.14.4.1** `void init_attrjeod__PrecessionMars ( )` `[friend]`

**8.14.4.2** `friend class InputProcessor` `[friend]`

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

## 8.14.5 Field Documentation

**8.14.5.1** `double jeod::PrecessionMars::J`

The (constant) inclination of the Mars mean orbit relative to the ICRF x-y plane.

`trick_units(radian)`

Definition at line 101 of file `precession_mars.hh`.

Referenced by `compute_fixed_matrices()`, and `jeod::RNPMars_rnp_mars_default_data::initialize()`.

**8.14.5.2** `double jeod::PrecessionMars::N`

The (constant) angle from the J2000 vernal equinox to the node of the Mars mean orbit and ICRF x-y plane.

`trick_units(radian)`

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

Referenced by `compute_fixed_matrices()`, and `jeod::RNPMars_rnp_mars_default_data::initialize()`.

**8.14.5.3** `double jeod::PrecessionMars::NJ_matrix[3][3]` `[private]`

The (constant) rotation matrix calculated from N and J.

`trick_units(-)`

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

Referenced by `compute_fixed_matrices()`, `PrecessionMars()`, and `update_rotation()`.

**8.14.5.4** `NutationMars* jeod::PrecessionMars::nutaton`

Pointer to the Mars nutation object, used to access nutation in longitude information.

Will be NULL (automatically) if anything but `Full_Term_RNP` is set in the [RNPMars](#) object.`trick_units(-)`

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

Referenced by `jeod::RNPMars::initialize()`, and `update_rotation()`.

## 8.14.5.5 double jeod::PrecessionMars::psi\_at\_j2000

The (constant) precession angle of Mars at the J2000 epoch.

trick\_units(radian)

Definition at line 78 of file precession\_mars.hh.

Referenced by jeod::RNPMars\_rnp\_mars\_default\_data::initialize(), and update\_rotation().

## 8.14.5.6 double jeod::PrecessionMars::psi\_dot

The (constant) simple precession rate of Mars.

trick\_units(radian/s)

Definition at line 83 of file precession\_mars.hh.

Referenced by jeod::RNPMars\_rnp\_mars\_default\_data::initialize(), and update\_rotation().

## 8.14.5.7 double jeod::PrecessionMars::psi\_precess

The most recent calculated value of the precession angle for Mars, measured since J2000 epoch.

trick\_units(radian)

Definition at line 89 of file precession\_mars.hh.

Referenced by update\_rotation().

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

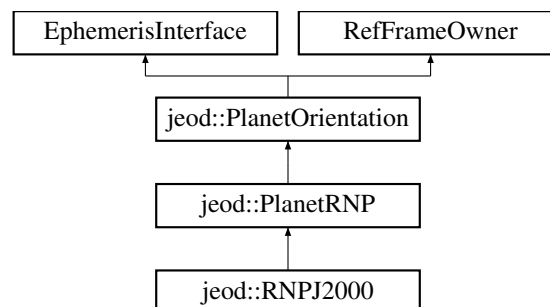
- [precession\\_mars.hh](#)
- [precession\\_mars.cc](#)

## 8.15 jeod::RNPJ2000 Class Reference

Implements the J2000 RNP model using the generic RNP framework.

```
#include <rnp_j2000.hh>
```

Inheritance diagram for jeod::RNPJ2000:



## Public Member Functions

- [RNPJ2000\(\)](#)  
default constructor.
- virtual [~RNPJ2000\(\)](#)

*Destructor.*

- void [initialize](#) (DynManager &manager)  
*Apply the various options, including fidelity of RNP and if polar motion is enabled or not, then call the base class initializer.*
- void [update\\_rnp](#) (TimeTT &time\_tt, TimeGMST &time\_gmst, TimeUT1 &time\_ut1)  
*Updates the complete RNP and supplies it to the Dynamics Manager.*
- void [update\\_axial\\_rotation](#) (TimeGMST &time\_gmst)  
*Updates the axial rotation portion of RNP and supplies the entire RNP to the Dynamics Manager.*
- virtual double [timestamp](#) () const
- virtual const char \* [get\\_name](#) () const  
*A re-declaration of the pure virtual function in order to convince trick that yes, this is a pure virtual class.*
- virtual void [ephem\\_update](#) ()

## Data Fields

- [RotationJ2000 RJ2000](#)  
*Earth J2000 rotation model.*
- [NutationJ2000 NJ2000](#)  
*Earth J2000 nutation model.*
- [PrecessionJ2000 PJ2000](#)  
*Earth J2000 precession model.*
- [PolarMotionJ2000 PMJ2000](#)  
*Earth J2000 polar motion model.*
- char [internal\\_name](#) [9]  
*The hard coded internal name to be returned on calling the overridden EphemerisInterface function "get\_name".*

## Private Member Functions

- void [get\\_dyn\\_time\\_ptr](#) (TimeGMST &gmst)
- [RNPJ2000](#) & [operator=](#) (const [RNPJ2000](#) &rhs)
- [RNPJ2000](#) (const [RNPJ2000](#) &rhs)

## Private Attributes

- TimeGMST \* [gmst\\_ptr](#)  
*Pointer to the TimeGMST used to update this object when [ephem\\_update](#) is invoked.*
- TimeDyn \* [time\\_dyn\\_ptr](#)  
*Pointer to the TimeDyn object, used to time stamp the reference frame when it is being updated.*
- double [last\\_updated\\_time\\_full](#)  
*The last update time, when updated through [update\\_rnp](#), for the RNP, referencing TimeDyn.seconds .*
- bool [never\\_updated\\_full](#)  
*Indicates that [last\\_updated\\_time\\_full](#) has never been populated, and that the update must be done regardless of given time.*
- double [last\\_updated\\_time\\_rotational](#)  
*The last update time, when updated through [update\\_axial\\_rotation](#), referencing TimeDyn.seconds .*
- bool [never\\_updated\\_rotational](#)  
*Indicates that [last\\_updated\\_time\\_rotational](#) has never been populated, and that the update must be done regardless of given time.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_RNPJ2000](#) ()

## Additional Inherited Members

### 8.15.1 Detailed Description

Implements the J2000 RNP model using the generic RNP framework.

Definition at line 73 of file `include/rnp_j2000.hh`.

### 8.15.2 Constructor & Destructor Documentation

#### 8.15.2.1 `jeod::RNPJ2000::RNPJ2000 ( void )`

default constructor.

Initialize all data

Definition at line 79 of file `rnp_j2000.cc`.

References `internal_name`, `NJ2000`, `jeod::PlanetRNP::nutation`, `PJ2000`, `PMJ2000`, `jeod::PlanetRNP::polar_motion`, `jeod::PlanetRNP::precession`, `RJ2000`, and `jeod::PlanetRNP::rotation`.

#### 8.15.2.2 `jeod::RNPJ2000::~~RNPJ2000 ( void )` `[virtual]`

Destructor.

Definition at line 104 of file `rnp_j2000.cc`.

#### 8.15.2.3 `jeod::RNPJ2000::RNPJ2000 ( const RNPJ2000 & rhs )` `[private]`

### 8.15.3 Member Function Documentation

#### 8.15.3.1 `void jeod::RNPJ2000::ephem_update ( )` `[virtual]`

Definition at line 334 of file `rnp_j2000.cc`.

References `jeod::PlanetOrientation::active`, `gmst_ptr`, `jeod::PlanetOrientation::orient_interface`, `jeod::RNPMessages::setup_error`, and `jeod::PlanetRNP::update_axial_rotation()`.

#### 8.15.3.2 `void jeod::RNPJ2000::get_dyn_time_ptr ( TimeGMST & gmst )` `[private]`

Definition at line 355 of file `rnp_j2000.cc`.

References `jeod::RNPMessages::setup_error`, and `time_dyn_ptr`.

Referenced by `update_axial_rotation()`, and `update_rnp()`.

#### 8.15.3.3 `const char * jeod::RNPJ2000::get_name ( ) const` `[virtual]`

A re-declaration of the pure virtual function in order to convince trick that yes, this is a pure virtual class.

**Returns**

Planet name.

Implements [jeod::PlanetRNP](#).

Definition at line 321 of file rnp\_j2000.cc.

References [internal\\_name](#).

#### 8.15.3.4 void jeod::RNPJ2000::initialize ( DynManager & dyn\_manager ) [virtual]

Apply the various options, including fidelity of RNP and if polar motion is enabled or not, then call the base class initializer.

Note that this function will still run even if the active flag is set to false. Additionally, note that this function does not update the rotation matrix between inertial and planet fixed. It only sets up the object to work with the DynManager. [update\\_rnp](#) must be called to update the RNP matrix. This function will also cache of TimeGMST from the given TimeManager

**Parameters**

<i>in, out</i>	<i>dyn_manager</i>	DynManager where the planet attitude to be updated is contained
----------------	--------------------	---

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

Definition at line 123 of file rnp\_j2000.cc.

References [jeod::PlanetRNP::ConstantNP](#), [jeod::PlanetRNP::FullRNP](#), [jeod::PlanetOrientation::initialize\(\)](#), [NJ2000](#), [jeod::RotationJ2000::nutation](#), [jeod::PlanetOrientation::planet\\_omega](#), [jeod::RotationJ2000::planet\\_rotational\\_velocity](#), [RJ2000](#), [jeod::PlanetRNP::rnp\\_type](#), [jeod::PlanetRNP::RotationOnly](#), and [jeod::RotationJ2000::use\\_full\\_rnp](#).

#### 8.15.3.5 RNPJ2000& jeod::RNPJ2000::operator= ( const RNPJ2000 & rhs ) [private]

#### 8.15.3.6 double jeod::RNPJ2000::timestamp ( ) const [virtual]

Definition at line 317 of file rnp\_j2000.cc.

References [last\\_updated\\_time\\_rotational](#).

#### 8.15.3.7 void jeod::RNPJ2000::update\_axial\_rotation ( TimeGMST & time\_gmst )

Updates the axial rotation portion of RNP and supplies the entire RNP to the Dynamics Manager.

**Parameters**

<i>in</i>	<i>time_gmst</i>	current time in the GMST time standard Units: The
-----------	------------------	--

Definition at line 265 of file rnp\_j2000.cc.

References [jeod::PlanetOrientation::active](#), [jeod::PlanetRNP::FullRNP](#), [get\\_dyn\\_time\\_ptr\(\)](#), [last\\_updated\\_time\\_rotational](#), [never\\_updated\\_rotational](#), [jeod::PlanetOrientation::planet](#), [jeod::PlanetRNP::rnp\\_type](#), [jeod::PlanetRNP::rotation](#), [time\\_dyn\\_ptr](#), [jeod::PlanetRNP::update\\_axial\\_rotation\(\)](#), and [jeod::PlanetRotation::update\\_time\(\)](#).

#### 8.15.3.8 void jeod::RNPJ2000::update\_rnp ( TimeTT & time\_tt, TimeGMST & time\_gmst, TimeUT1 & time\_ut1 )

Updates the complete RNP and supplies it to the Dynamics Manager.



## Parameters

in	<i>time_tt</i>	The current time in the TT time standard
in	<i>time_gmst</i>	current time in the GMST time standard Units: The
in	<i>time_ut1</i>	current time in the UT1 time standard Units: The

Definition at line 166 of file rnp\_j2000.cc.

References jeod::PlanetOrientation::active, jeod::PlanetRNP::enable\_polar, jeod::PlanetRNP::FullRNP, get\_dyn\_time\_ptr(), gmst\_ptr, last\_updated\_time\_full, last\_updated\_time\_rotational, never\_updated\_full, never\_updated\_rotational, jeod::PlanetRNP::nututation, jeod::PlanetOrientation::planet, jeod::PlanetRNP::polar\_motion, jeod::PlanetRNP::precession, jeod::PlanetRNP::rnp\_type, jeod::PlanetRNP::rotation, time\_dyn\_ptr, jeod::PlanetRNP::update\_rnp(), and jeod::PlanetRotation::update\_time().

## 8.15.4 Friends And Related Function Documentation

8.15.4.1 void init\_attrjeod\_RNPJ2000 ( ) [friend]

8.15.4.2 friend class InputProcessor [friend]

Definition at line 75 of file include/rnp\_j2000.hh.

## 8.15.5 Field Documentation

8.15.5.1 TimeGMST\* jeod::RNPJ2000::gmst\_ptr [private]

Pointer to the TimeGMST used to update this object when ephemeris\_update is invoked.

trick\_units(-)

Definition at line 154 of file include/rnp\_j2000.hh.

Referenced by ephemeris\_update(), and update\_rnp().

8.15.5.2 char jeod::RNPJ2000::internal\_name[9]

The hard coded internal name to be returned on calling the overridden EphemerisInterface function "get\_name".

trick\_units(-)

Definition at line 139 of file include/rnp\_j2000.hh.

Referenced by get\_name(), and RNPJ2000().

8.15.5.3 double jeod::RNPJ2000::last\_updated\_time\_full [private]

The last update time, when updated through update\_rnp, for the RNP, referencing TimeDyn.seconds .

If the time from time\_dyn\_ptr is the same as this update time, then the RNP will not be updated. This is to prevent unnecessary updating.trick\_units(s)

Definition at line 169 of file include/rnp\_j2000.hh.

Referenced by update\_rnp().

8.15.5.4 double jeod::RNPJ2000::last\_updated\_time\_rotational [private]

The last update time, when updated through update\_axial\_rotation, referencing TimeDyn.seconds .

If the time from `time_dyn_ptr` is the same as this update time, then the R component of RNP will not be updated. This is to prevent unnecessary updating.`trick_units(s)`

Definition at line 184 of file `include/rnp_j2000.hh`.

Referenced by `timestamp()`, `update_axial_rotation()`, and `update_rnp()`.

#### 8.15.5.5 `bool jeod::RNPJ2000::never_updated_full` [private]

Indicates that `last_updated_time_full` has never been populated, and that the update must be done regardless of given time.

`trick_units(-)`

Definition at line 175 of file `include/rnp_j2000.hh`.

Referenced by `update_rnp()`.

#### 8.15.5.6 `bool jeod::RNPJ2000::never_updated_rotational` [private]

Indicates that `last_updated_time_rotational` has never been populated, and that the update must be done regardless of given time.

`trick_units(-)`

Definition at line 190 of file `include/rnp_j2000.hh`.

Referenced by `update_axial_rotation()`, and `update_rnp()`.

#### 8.15.5.7 `NutationJ2000 jeod::RNPJ2000::NJ2000`

Earth J2000 nutation model.

`trick_units(-)`

Definition at line 88 of file `include/rnp_j2000.hh`.

Referenced by `initialize()`, and `RNPJ2000()`.

#### 8.15.5.8 `PrecessionJ2000 jeod::RNPJ2000::PJ2000`

Earth J2000 precession model.

`trick_units(-)`

Definition at line 92 of file `include/rnp_j2000.hh`.

Referenced by `RNPJ2000()`.

#### 8.15.5.9 `PolarMotionJ2000 jeod::RNPJ2000::PMJ2000`

Earth J2000 polar motion model.

`trick_units(-)`

Definition at line 96 of file `include/rnp_j2000.hh`.

Referenced by `RNPJ2000()`.

#### 8.15.5.10 `RotationJ2000 jeod::RNPJ2000::RJ2000`

Earth J2000 rotation model.

trick\_units(-)

Definition at line 84 of file include/rnp\_j2000.hh.

Referenced by initialize(), and RNPJ2000().

#### 8.15.5.11 TimeDyn\* jeod::RNPJ2000::time\_dyn\_ptr [private]

Pointer to the TimeDyn object, used to time stamp the reference frame when it is being updated.

trick\_units(-)

Definition at line 160 of file include/rnp\_j2000.hh.

Referenced by get\_dyn\_time\_ptr(), update\_axial\_rotation(), and update\_rnp().

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

- [include/rnp\\_j2000.hh](#)
- [rnp\\_j2000.cc](#)

## 8.16 jeod::RNPJ2000\_rnp\_j2000\_default\_data Class Reference

```
#include <rnp_j2000.hh>
```

### Public Member Functions

- void [initialize](#) (RNPJ2000 \*)

### 8.16.1 Detailed Description

Definition at line 18 of file data/include/rnp\_j2000.hh.

### 8.16.2 Member Function Documentation

#### 8.16.2.1 void jeod::RNPJ2000\_rnp\_j2000\_default\_data::initialize ( RNPJ2000 \* RNPJ2000\_ptr )

Definition at line 44 of file data\_rnp\_j2000.cc.

References [jeod::PlanetRNP::enable\\_polar](#), [jeod::PlanetRNP::FullRNP](#), [jeod::PlanetOrientation::name](#), [jeod::PlanetOrientation::planet\\_omega](#), and [jeod::PlanetRNP::rnp\\_type](#).

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

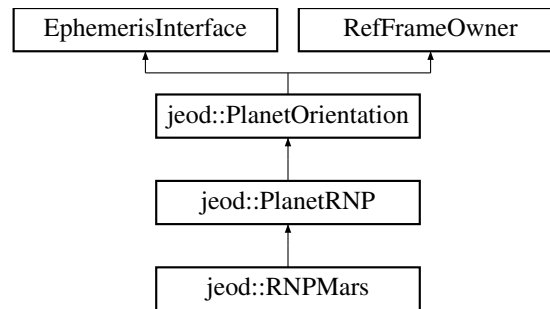
- [data/include/rnp\\_j2000.hh](#)
- [data\\_rnp\\_j2000.cc](#)

## 8.17 jeod::RNPMars Class Reference

Implements the "Pathfinder" Mars RNP model using the generic RNP framework.

```
#include <rnp_mars.hh>
```

Inheritance diagram for jeod::RNPMars:



## Public Member Functions

- [RNPMars](#) ()  
*Default constructor.*
- virtual [~RNPMars](#) ()  
*Destructor.*
- void [initialize](#) (DynManager &manager)  
*Perform setup of the object, then call the base class initializer.*
- void [update\\_rnp](#) (TimeTT &time\_tt)  
*Update the complete RNP and supply it to the Dynamics Manager.*
- void [update\\_axial\\_rotation](#) (TimeTT &time\_tt)  
*Update rotation portion of RNP and supply RNP to dynamics manager.*
- virtual double [timestamp](#) () const  
*Return the last time at which the RNP was updated.*
- virtual const char \* [get\\_name](#) () const  
*Return the internal name of the object.*
- virtual void [ephem\\_update](#) ()  
*Define 'alias' to call axial update function, for polymorphism.*

## Data Fields

- [RotationMars](#) RMars  
*"Pathfinder" Mars rotation model.*
- [NutationMars](#) NMars  
*"Pathfinder" Mars nutation model.*
- [PrecessionMars](#) PMars  
*"Pathfinder" Mars precession model.*
- char [internal\\_name](#) [8]  
*The hard coded internal name to be returned on calling the overridden EphemerisInterface function "get\_name".*

## Private Member Functions

- void [get\\_dyn\\_time\\_ptr](#) (TimeTT &time\_tt)  
*Get simulation time via a two-step pointer trail.*
- [RNPMars](#) & [operator=](#) (const [RNPMars](#) &rhs)
- [RNPMars](#) (const [RNPMars](#) &rhs)

## Private Attributes

- TimeTT \* [tt\\_ptr](#)  
*Pointer to the TimeTT used to update this object when ephemeris\_update is invoked.*
- TimeDyn \* [time\\_dyn\\_ptr](#)  
*Pointer to the TimeDyn object, used to time stamp the reference frame when it is being updated.*
- double [last\\_updated\\_time\\_full](#)  
*The last update time for the RNP, when updated through update\_rnp.*
- bool [never\\_updated\\_full](#)  
*Indicates that last\_updated\_time\_full has never been populated, and that the update must be done regardless of given time.*
- double [last\\_updated\\_time\\_rotational](#)  
*The last rotational update time, when updated through update\_axial\_rotation, referencing TimeDyn.seconds.*
- bool [never\\_updated\\_rotational](#)  
*Indicates that last\_updated\_time\_rotational has never been populated, and that the update must be done regardless of given time.*

## Friends

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

## Additional Inherited Members

### 8.17.1 Detailed Description

Implements the "Pathfinder" Mars RNP model using the generic RNP framework.

Definition at line 67 of file include/rnp\_mars.hh.

### 8.17.2 Constructor & Destructor Documentation

#### 8.17.2.1 jeod::RNPMars::RNPMars ( void )

Default constructor.

Definition at line 71 of file rnp\_mars.cc.

References [jeod::PlanetRNP::enable\\_polar](#), [internal\\_name](#), [NMars](#), [jeod::PlanetRNP::nutaton](#), [PMars](#), [jeod::PlanetRNP::polar\\_motion](#), [jeod::PlanetRNP::precession](#), [RMars](#), and [jeod::PlanetRNP::rotation](#).

#### 8.17.2.2 jeod::RNPMars::~~RNPMars ( ) [virtual]

Destructor.

Definition at line 104 of file rnp\_mars.cc.

#### 8.17.2.3 jeod::RNPMars::RNPMars ( const RNPMars & rhs ) [private]

### 8.17.3 Member Function Documentation

#### 8.17.3.1 void jeod::RNPMars::ephemeris\_update ( ) [virtual]

Define 'alias' to call axial update function, for polymorphism.

Definition at line 283 of file rnp\_mars.cc.

References `jeod::PlanetOrientation::active`, `jeod::PlanetOrientation::orient_interface`, `jeod::RNPMessages::setup_error`, `tt_ptr`, and `jeod::PlanetRNP::update_axial_rotation()`.

**8.17.3.2** `void jeod::RNPMars::get_dyn_time_ptr ( TimeTT & time_tt ) [private]`

Get simulation time via a two-step pointer trail.

Parameters

<i>in</i>	<i>time_tt</i>	Current TT time
-----------	----------------	-----------------

Definition at line 308 of file rnp\_mars.cc.

References `jeod::RNPMessages::setup_error`, and `time_dyn_ptr`.

Referenced by `update_axial_rotation()`, and `update_rnp()`.

**8.17.3.3** `const char * jeod::RNPMars::get_name ( ) const [virtual]`

Return the internal name of the object.

Implements [jeod::PlanetRNP](#).

Definition at line 274 of file rnp\_mars.cc.

References `internal_name`.

**8.17.3.4** `void jeod::RNPMars::initialize ( DynManager & dyn_manager ) [virtual]`

Perform setup of the object, then call the base class initializer.

Parameters

<i>in, out</i>	<i>dyn_manager</i>	Ref to dynamics manager
----------------	--------------------	-------------------------

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

Definition at line 113 of file rnp\_mars.cc.

References `jeod::PrecessionMars::compute_fixed_matrices()`, `jeod::PlanetRNP::ConstantNP`, `jeod::PlanetRNP::FullRNP`, `jeod::PlanetOrientation::initialize()`, `NMars`, `jeod::PrecessionMars::nutation`, `jeod::RotationMars::nutation`, `jeod::PlanetOrientation::planet_omega`, `jeod::RotationMars::planet_rotational_velocity`, `PMars`, `RMars`, `jeod::PlanetRNP::rnp_type`, `jeod::PlanetRNP::RotationOnly`, and `jeod::RotationMars::use_full_rnp`.

**8.17.3.5** `RNPMars& jeod::RNPMars::operator= ( const RNPMars & rhs ) [private]`

**8.17.3.6** `double jeod::RNPMars::timestamp ( ) const [virtual]`

Return the last time at which the RNP was updated.

Definition at line 265 of file rnp\_mars.cc.

References `last_updated_time_rotational`.

**8.17.3.7** `void jeod::RNPMars::update_axial_rotation ( TimeTT & time_tt )`

Update rotation portion of RNP and supply RNP to dynamics manager.

## Parameters

<i>in</i>	<i>time_tt</i>	Current TT time
-----------	----------------	-----------------

Definition at line 211 of file rnp\_mars.cc.

References `jeod::PlanetOrientation::active`, `jeod::PlanetRNP::FullRNP`, `get_dyn_time_ptr()`, `last_updated_time_rotational`, `never_updated_rotational`, `jeod::PlanetOrientation::planet`, `jeod::PlanetRNP::rnp_type`, `jeod::PlanetRNP::rotation`, `time_dyn_ptr`, `jeod::PlanetRNP::update_axial_rotation()`, and `jeod::PlanetRotation::update_time()`.

#### 8.17.3.8 void jeod::RNPMars::update\_rnp ( TimeTT & *time\_tt* )

Update the complete RNP and supply it to the Dynamics Manager.

## Parameters

<i>in</i>	<i>time_tt</i>	Current Terrestrial Time
-----------	----------------	--------------------------

Definition at line 148 of file rnp\_mars.cc.

References `jeod::PlanetOrientation::active`, `jeod::PlanetRNP::FullRNP`, `get_dyn_time_ptr()`, `last_updated_time_full`, `last_updated_time_rotational`, `never_updated_full`, `never_updated_rotational`, `jeod::PlanetRNP::nutration`, `jeod::PlanetOrientation::planet`, `jeod::PlanetRNP::precession`, `jeod::PlanetRNP::rnp_type`, `jeod::PlanetRNP::rotation`, `time_dyn_ptr`, `tt_ptr`, `jeod::PlanetRNP::update_rnp()`, and `jeod::PlanetRotation::update_time()`.

### 8.17.4 Friends And Related Function Documentation

#### 8.17.4.1 void init\_attrjeod\_\_RNPMars ( ) [*friend*]

#### 8.17.4.2 friend class InputProcessor [*friend*]

Definition at line 69 of file include/rnp\_mars.hh.

### 8.17.5 Field Documentation

#### 8.17.5.1 char jeod::RNPMars::internal\_name[8]

The hard coded internal name to be returned on calling the overridden EphemerisInterface function "get\_name".  
trick\_units(-)

Definition at line 92 of file include/rnp\_mars.hh.

Referenced by `get_name()`, and `RNPMars()`.

#### 8.17.5.2 double jeod::RNPMars::last\_updated\_time\_full [*private*]

The last update time for the RNP, when updated through `update_rnp`.

References `TimeDyn.seconds`. If the time from `time_dyn_ptr` is the same as this update time, then the RNP will not be updated; prevents unnecessary updating.  
trick\_units(s)

Definition at line 113 of file include/rnp\_mars.hh.

Referenced by `update_rnp()`.

#### 8.17.5.3 double jeod::RNPMars::last\_updated\_time\_rotational [*private*]

The last rotational update time, when updated through `update_axial_rotation`, referencing `TimeDyn.seconds`.

If the time from `time_dyn_ptr` matches this update time, then the R component of the RNP will not be updated; prevents unnecessary updating.

Definition at line 127 of file `include/rnp_mars.hh`.

Referenced by `timestamp()`, `update_axial_rotation()`, and `update_rnp()`.

#### 8.17.5.4 `bool jeod::RNPMars::never_updated_full` [private]

Indicates that `last_updated_time_full` has never been populated, and that the update must be done regardless of given time.

`trick_units(-)`

Definition at line 119 of file `include/rnp_mars.hh`.

Referenced by `update_rnp()`.

#### 8.17.5.5 `bool jeod::RNPMars::never_updated_rotational` [private]

Indicates that `last_updated_time_rotational` has never been populated, and that the update must be done regardless of given time.

`trick_units(-)`

Definition at line 133 of file `include/rnp_mars.hh`.

Referenced by `update_axial_rotation()`, and `update_rnp()`.

#### 8.17.5.6 `NutationMars jeod::RNPMars::NMars`

"Pathfinder" Mars nutation model.

`trick_units(-)`

Definition at line 81 of file `include/rnp_mars.hh`.

Referenced by `jeod::RNPMars_rnp_mars_default_data::initialize()`, `initialize()`, and `RNPMars()`.

#### 8.17.5.7 `PrecessionMars jeod::RNPMars::PMars`

"Pathfinder" Mars precession model.

`trick_units(-)`

Definition at line 86 of file `include/rnp_mars.hh`.

Referenced by `jeod::RNPMars_rnp_mars_default_data::initialize()`, `initialize()`, and `RNPMars()`.

#### 8.17.5.8 `RotationMars jeod::RNPMars::RMars`

"Pathfinder" Mars rotation model.

`trick_units(-)`

Definition at line 76 of file `include/rnp_mars.hh`.

Referenced by `jeod::RNPMars_rnp_mars_default_data::initialize()`, `initialize()`, and `RNPMars()`.

#### 8.17.5.9 `TimeDyn* jeod::RNPMars::time_dyn_ptr` [private]

Pointer to the `TimeDyn` object, used to time stamp the reference frame when it is being updated.



trick\_units(-)

Definition at line 105 of file include/rnp\_mars.hh.

Referenced by get\_dyn\_time\_ptr(), update\_axial\_rotation(), and update\_rnp().

#### 8.17.5.10 TimeTT\* jeod::RNPMars::tt\_ptr [private]

Pointer to the TimeTT used to update this object when ephemeris\_update is invoked.

trick\_units(-)

Definition at line 99 of file include/rnp\_mars.hh.

Referenced by ephemeris\_update(), and update\_rnp().

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

- [include/rnp\\_mars.hh](#)
- [rnp\\_mars.cc](#)

## 8.18 jeod::RNPMars\_rnp\_mars\_default\_data Class Reference

```
#include <rnp_mars.hh>
```

### Public Member Functions

- void [initialize](#) (RNPMars \*)

### 8.18.1 Detailed Description

Definition at line 18 of file data/include/rnp\_mars.hh.

### 8.18.2 Member Function Documentation

#### 8.18.2.1 void jeod::RNPMars\_rnp\_mars\_default\_data::initialize ( RNPMars \* RNPMars\_ptr )

Definition at line 47 of file data\_rnp\_mars.cc.

References [jeod::PlanetRNP::enable\\_polar](#), [jeod::PlanetRNP::FullRNP](#), [jeod::NutationMars::l\\_at\\_j2000](#), [jeod::NutationMars::l\\_dot](#), [jeod::NutationMars::l\\_m\\_orig](#), [jeod::NutationMars::int\\_to\\_double](#), [jeod::PrecessionMars::J](#), [jeod::NutationMars::mean\\_anomaly\\_j2000](#), [jeod::NutationMars::mean\\_motion](#), [jeod::PrecessionMars::N](#), [jeod::PlanetOrientation::name](#), [jeod::RNPMars::NMars](#), [jeod::RotationMars::phi\\_at\\_j2000](#), [jeod::PlanetOrientation::planet\\_omega](#), [jeod::RNPMars::PMars](#), [jeod::PrecessionMars::psi\\_at\\_j2000](#), [jeod::PrecessionMars::psi\\_dot](#), [jeod::NutationMars::psi\\_m\\_orig](#), [jeod::NutationMars::q\\_angle\\_j2000](#), [jeod::RNPMars::RMars](#), and [jeod::PlanetRNP::rnp\\_type](#).

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

- [data/include/rnp\\_mars.hh](#)
- [data\\_rnp\\_mars.cc](#)

## 8.19 jeod::RNPMessages Class Reference

Describes messages used in the RNP model.

```
#include <RNP_messages.hh>
```

## Static Public Attributes

- static char const \* [initialization\\_error](#)  
*Indicates an error during initialization.*
- static char const \* [fidelity\\_error](#)  
*Indicates a mismatch between the requested fidelity and what is available to the model.*
- static char const \* [setup\\_error](#)  
*Indicates an error during setup of the RNP model.*
- static char const \* [polar\\_motion\\_table\\_warning](#)  
*Indicates a problem with the interpolation table commonly found in polar motion implementations.*

## Private Member Functions

- [RNPMessages](#) (void)
- [RNPMessages](#) (const [RNPMessages](#) &rhs)
- [RNPMessages](#) & operator= (const [RNPMessages](#) &rhs)

## Friends

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

### 8.19.1 Detailed Description

Describes messages used in the RNP model.

Definition at line 49 of file RNP\_messages.hh.

### 8.19.2 Constructor & Destructor Documentation

8.19.2.1 `jeod::RNPMessages::RNPMessages ( void )` [private]

8.19.2.2 `jeod::RNPMessages::RNPMessages ( const RNPMessages & rhs )` [private]

### 8.19.3 Member Function Documentation

8.19.3.1 `RNPMessages& jeod::RNPMessages::operator= ( const RNPMessages & rhs )` [private]

### 8.19.4 Friends And Related Function Documentation

8.19.4.1 `void init_attrjeod__RNPMessages ( )` [friend]

8.19.4.2 `friend class InputProcessor` [friend]

Definition at line 51 of file RNP\_messages.hh.

### 8.19.5 Field Documentation

8.19.5.1 `char const * jeod::RNPMessages::fidelity_error` [static]

**Initial value:**

```
=
    "environment/RNP/" "fidelity_error"
```

Indicates a mismatch between the requested fidelity and what is available to the model.

trick\_units(—)

Definition at line 68 of file RNP\_messages.hh.

Referenced by jeod::PlanetRNP::propagate\_rnp(), and jeod::PlanetRNP::update\_rnp().

#### 8.19.5.2 char const \* jeod::RNPMessages::initialization\_error [static]

**Initial value:**

```
=
    "environment/RNP/" "initialization_error"
```

Indicates an error during initialization.

trick\_units(—)

Definition at line 62 of file RNP\_messages.hh.

Referenced by jeod::PlanetOrientation::initialize(), jeod::PolarMotionJ2000::initialize(), and jeod::NutationJ2000::initialize().

#### 8.19.5.3 char const \* jeod::RNPMessages::polar\_motion\_table\_warning [static]

**Initial value:**

```
=
    "environment/RNP/" "polar_motion_table_warning"
```

Indicates a problem with the interpolation table commonly found in polar motion implementations.

trick\_units(—)

Definition at line 81 of file RNP\_messages.hh.

Referenced by jeod::PolarMotionJ2000::update\_rotation().

#### 8.19.5.4 char const \* jeod::RNPMessages::setup\_error [static]

**Initial value:**

```
=
    "environment/RNP/" "setup_error"
```

Indicates an error during setup of the RNP model.

trick\_units(—)

Definition at line 73 of file RNP\_messages.hh.

Referenced by jeod::RNPMars::ephem\_update(), jeod::RNPMars::ephem\_update(), jeod::RNPMars::get\_dyn\_time\_ptr(), jeod::RNPMars::get\_dyn\_time\_ptr(), jeod::PlanetRNP::propagate\_rnp(), jeod::PlanetRNP::update\_axial\_rotation(), jeod::PlanetRNP::update\_rnp(), jeod::RotationMars::update\_rotation(), jeod::RotationJ2000::update\_rotation(), and jeod::PrecessionMars::update\_rotation().

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

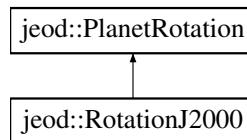
- [RNP\\_messages.hh](#)
- [RNP\\_messages.cc](#)

## 8.20 jeod::RotationJ2000 Class Reference

Implements the axial rotation portion of J2000 RNP.

```
#include <rotation_j2000.hh>
```

Inheritance diagram for jeod::RotationJ2000:



### Public Member Functions

- [RotationJ2000](#) ()  
*default constructor, initialize low level data*
- virtual [~RotationJ2000](#) ()  
*destructor*
- virtual void [update\\_rotation](#) ()  
*J2000 specific implementation of update\_rotation, from [PlanetRotation](#).*

### Data Fields

- double [planet\\_rotational\\_velocity](#)  
*The nominal axial rotational velocity of the earth.*
- [NutationJ2000](#) \* [nutation](#)  
*Pointer to the J2000 nutation object, used for get obliquity information out.*
- bool [use\\_full\\_rnp](#)  
*Tells the rotation object if it should use a full blown rotation formulation, or just use the time passed multiplied by the rotational velocity.*
- double [theta\\_gast](#)  
*The last theta\_gast (angle the earth had axially rotated) calculated.*
- double [GMST](#)  
*GMST, currently saved for logging purposes.*

### Private Member Functions

- [RotationJ2000](#) & [operator=](#) (const [RotationJ2000](#) &rhs)
- [RotationJ2000](#) (const [RotationJ2000](#) &rhs)

### Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_RotationJ2000](#) ()

### Additional Inherited Members

#### 8.20.1 Detailed Description

Implements the axial rotation portion of J2000 RNP.

Definition at line 64 of file `rotation_j2000.hh`.

## 8.20.2 Constructor & Destructor Documentation

### 8.20.2.1 jeod::RotationJ2000::RotationJ2000 ( void )

default constructor, initialize low level data

Definition at line 67 of file rotation\_j2000.cc.

### 8.20.2.2 jeod::RotationJ2000::~~RotationJ2000 ( void ) [virtual]

destructor

Definition at line 80 of file rotation\_j2000.cc.

### 8.20.2.3 jeod::RotationJ2000::RotationJ2000 ( const RotationJ2000 & rhs ) [private]

## 8.20.3 Member Function Documentation

### 8.20.3.1 RotationJ2000& jeod::RotationJ2000::operator= ( const RotationJ2000 & rhs ) [private]

### 8.20.3.2 void jeod::RotationJ2000::update\_rotation ( void ) [virtual]

J2000 specific implementation of update\_rotation, from [PlanetRotation](#).

For axial rotation

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

Definition at line 91 of file rotation\_j2000.cc.

References [jeod::PlanetRotation::current\\_time](#), [jeod::PlanetRotation::DEGTORAD](#), [jeod::NutationJ2000::equa\\_of\\_equi](#), [nutaton](#), [planet\\_rotational\\_velocity](#), [jeod::PlanetRotation::rotation](#), [jeod::RNPMessages::setup\\_error](#), [theta\\_gast](#), and [use\\_full\\_rnp](#).

## 8.20.4 Friends And Related Function Documentation

### 8.20.4.1 void init\_attrjeod\_\_RotationJ2000 ( ) [friend]

### 8.20.4.2 friend class InputProcessor [friend]

Definition at line 66 of file rotation\_j2000.hh.

## 8.20.5 Field Documentation

### 8.20.5.1 double jeod::RotationJ2000::GMST

GMST, currently saved for logging purposes.

trick\_units(–)

Definition at line 100 of file rotation\_j2000.hh.

### 8.20.5.2 NutationJ2000\* jeod::RotationJ2000::nutaton

Pointer to the J2000 nutation object, used for get obliquity information out.

Will be NULL (automatically) if anything but Full\_Term\_RNP is set in the [RNPJ2000](#) object [trick\\_units\(–\)](#)

Definition at line 80 of file rotation\_j2000.hh.

Referenced by `jeod::RNPJ2000::initialize()`, and `update_rotation()`.

#### 8.20.5.3 `double jeod::RotationJ2000::planet_rotational_velocity`

The nominal axial rotational velocity of the earth.

`trick_units(radian/s)`

Definition at line 73 of file `rotation_j2000.hh`.

Referenced by `jeod::RNPJ2000::initialize()`, and `update_rotation()`.

#### 8.20.5.4 `double jeod::RotationJ2000::theta_gast`

The last `theta_gast` (angle the earth had axially rotated) calculated.

`trick_units(radian)`

Definition at line 95 of file `rotation_j2000.hh`.

Referenced by `update_rotation()`.

#### 8.20.5.5 `bool jeod::RotationJ2000::use_full_rnp`

Tells the rotation object if it should use a full blown rotation formulation, or just use the time passed multiplied by the rotational velocity.

Used with the different initialization options for the main RNP class `trick_units(-)`

Definition at line 89 of file `rotation_j2000.hh`.

Referenced by `jeod::RNPJ2000::initialize()`, and `update_rotation()`.

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

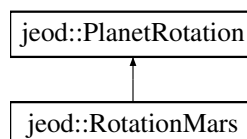
- [rotation\\_j2000.hh](#)
- [rotation\\_j2000.cc](#)

## 8.21 `jeod::RotationMars` Class Reference

Implements the axial rotation portion of the "Pathfinder" Mars RNP model.

```
#include <rotation_mars.hh>
```

Inheritance diagram for `jeod::RotationMars`:



### Public Member Functions

- [RotationMars](#) ()  
*Default constructor, initialize low level data.*
- `virtual ~RotationMars` ()  
*Destructor.*

- virtual void [update\\_rotation](#) ()  
*RotationMars specific implementation of update\_rotation, for axial rotation.*

## Data Fields

- double [planet\\_rotational\\_velocity](#)  
*The Mars average axial rotational velocity.*
- [NutationMars](#) \* [nututation](#)  
*Pointer to the Mars nutation object, used to access nutation in longitude information.*
- bool [use\\_full\\_rnp](#)  
*Tells the rotation object if it should use a full blown rotation formulation, or if it should just use the time passed multiplied by the rotational velocity.*
- double [phi\\_at\\_j2000](#)  
*The (constant) rotated angle of Mars at the J2000 epoch.*
- double [phi\\_spin](#)  
*The most recent calculated value of the rotation angle for Mars, measured since J2000 epoch.*

## Private Member Functions

- [RotationMars](#) & [operator=](#) (const [RotationMars](#) &rhs)
- [RotationMars](#) (const [RotationMars](#) &rhs)

## Friends

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

## Additional Inherited Members

### 8.21.1 Detailed Description

Implements the axial rotation portion of the "Pathfinder" Mars RNP model.  
Definition at line 63 of file `rotation_mars.hh`.

### 8.21.2 Constructor & Destructor Documentation

#### 8.21.2.1 jeod::RotationMars::RotationMars ( void )

Default constructor, initialize low level data.  
Definition at line 63 of file `rotation_mars.cc`.

#### 8.21.2.2 jeod::RotationMars::~~RotationMars ( void ) [virtual]

Destructor.  
Definition at line 78 of file `rotation_mars.cc`.

8.21.2.3 `jeod::RotationMars::RotationMars ( const RotationMars & rhs ) [private]`

### 8.21.3 Member Function Documentation

8.21.3.1 `RotationMars& jeod::RotationMars::operator= ( const RotationMars & rhs ) [private]`

8.21.3.2 `void jeod::RotationMars::update_rotation ( void ) [virtual]`

[RotationMars](#) specific implementation of `update_rotation`, for axial rotation.

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

Definition at line 89 of file `rotation_mars.cc`.

References `jeod::PlanetRotation::current_time`, `nutaton`, `jeod::NutationMars::nutaton_in_longitude`, `jeod::NutationMars::obliquity_angle`, `phi_at_j2000`, `phi_spin`, `planet_rotational_velocity`, `jeod::PlanetRotation::rotation`, `jeod::RNPMessages::setup_error`, and `use_full_rnp`.

### 8.21.4 Friends And Related Function Documentation

8.21.4.1 `void init_attrjeod__RotationMars ( ) [friend]`

8.21.4.2 `friend class InputProcessor [friend]`

Definition at line 65 of file `rotation_mars.hh`.

### 8.21.5 Field Documentation

8.21.5.1 `NutationMars* jeod::RotationMars::nutaton`

Pointer to the Mars nutation object, used to access nutation in longitude information.

Will be NULL (automatically) if anything but `Full_Term_RNP` is set in the [RNPMars](#) object.`trick_units(-)`

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

Referenced by `jeod::RNPMars::initialize()`, and `update_rotation()`.

8.21.5.2 `double jeod::RotationMars::phi_at_j2000`

The (constant) rotated angle of Mars at the J2000 epoch.

`trick_units(radian)`

Definition at line 92 of file `rotation_mars.hh`.

Referenced by `jeod::RNPMars_rnp_mars_default_data::initialize()`, and `update_rotation()`.

8.21.5.3 `double jeod::RotationMars::phi_spin`

The most recent calculated value of the rotation angle for Mars, measured since J2000 epoch.

`trick_units(radian)`

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

Referenced by `update_rotation()`.



#### 8.21.5.4 double jeod::RotationMars::planet\_rotational\_velocity

The Mars average axial rotational velocity.

trick\_units(radian/s)

Definition at line 72 of file rotation\_mars.hh.

Referenced by jeod::RNPMars::initialize(), and update\_rotation().

#### 8.21.5.5 bool jeod::RotationMars::use\_full\_rnp

Tells the rotation object if it should use a full blown rotation formulation, or if it should just use the time passed multiplied by the rotational velocity.

Used with the different initialization options for the main RNP class.trick\_units(-)

Definition at line 87 of file rotation\_mars.hh.

Referenced by jeod::RNPMars::initialize(), and update\_rotation().

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

- [rotation\\_mars.hh](#)
- [rotation\\_mars.cc](#)



## Chapter 9

# File Documentation

### 9.1 `class_declarations.hh` File Reference

Forward declarations of classes defined for JEOD 2.0 Generic RNP.

#### Namespaces

- [jeod](#)

*Namespace jeod.*

#### 9.1.1 Detailed Description

Forward declarations of classes defined for JEOD 2.0 Generic RNP.

Definition in file [GenericRNP/include/class\\_declarations.hh](#).

### 9.2 `class_declarations.hh` File Reference

Forward declarations of classes defined for JEOD 2.0 J2000 RNP.

#### Namespaces

- [jeod](#)

*Namespace jeod.*

#### 9.2.1 Detailed Description

Forward declarations of classes defined for JEOD 2.0 J2000 RNP.

Definition in file [RNPJ2000/include/class\\_declarations.hh](#).

### 9.3 `data_nutation_j2000.cc` File Reference

```
#include "environment/RNP/GenericRNP/include/planet_rotation_init.hh"
```

```
#include "environment/RNP/RNPJ2000/include/nutation_j2000_init.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/nutation_j2000.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

## Macros

- `#define` [JEOD\\_FRIEND\\_CLASS](#) NutationJ2000Init\_nutation\_j2000\_default\_data

### 9.3.1 Macro Definition Documentation

9.3.1.1 `#define` [JEOD\\_FRIEND\\_CLASS](#) NutationJ2000Init\_nutation\_j2000\_default\_data

Definition at line 26 of file data\_nutation\_j2000.cc.

## 9.4 data\_rnp\_j2000.cc File Reference

```
#include "environment/RNP/GenericRNP/include/planet_rnp.hh"
#include "environment/RNP/RNPJ2000/include/rnp_j2000.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/ref_frames/include/ref_frame_interface.hh"
#include "../include/rnp_j2000.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

## Macros

- `#define` [JEOD\\_FRIEND\\_CLASS](#) RNPJ2000\_rnp\_j2000\_default\_data

### 9.4.1 Macro Definition Documentation

9.4.1.1 `#define` [JEOD\\_FRIEND\\_CLASS](#) RNPJ2000\_rnp\_j2000\_default\_data

Definition at line 27 of file data\_rnp\_j2000.cc.

## 9.5 data\_rnp\_mars.cc File Reference

```
#include <cmath>
#include "environment/RNP/GenericRNP/include/planet_rotation.hh"
#include "environment/RNP/GenericRNP/include/planet_rnp.hh"
#include "environment/RNP/RNPMars/include/rnp_mars.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/rnp_mars.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

### Macros

- `#define` [JEOD\\_FRIEND\\_CLASS](#) RNPMars\_rnp\_mars\_default\_data

#### 9.5.1 Macro Definition Documentation

##### 9.5.1.1 `#define` JEOD\_FRIEND\_CLASS RNPMars\_rnp\_mars\_default\_data

Definition at line 26 of file data\_rnp\_mars.cc.

## 9.6 nutation\_j2000.cc File Reference

Implementation for the NutationJ2000 class.

```
#include <cstdlib>
#include <cmath>
#include "environment/RNP/GenericRNP/include/RNP_messages.hh"
#include "utils/math/include/matrix3x3.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/nutation_j2000.hh"
#include "../include/nutation_j2000_init.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

#### 9.6.1 Detailed Description

Implementation for the NutationJ2000 class.

Definition in file [nutation\\_j2000.cc](#).

## 9.7 nutation\_j2000.hh File Reference

### Data Structures

- class [jeod::NutationJ2000Init\\_nutation\\_j2000\\_default\\_data](#)

### Namespaces

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

## 9.8 nutation\_j2000.hh File Reference

Model the nutation portion of the RNP routine for the Standard Epoch J2000 This is form of the IAU-76/FK5 RNP model, as implemented in Jeod 1.52.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "environment/RNP/GenericRNP/include/planet_rotation.hh"
```

### Data Structures

- class [jeod::NutationJ2000](#)  
*Implements the nutation portion of the J2000 RNP.*

### Namespaces

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

### 9.8.1 Detailed Description

Model the nutation portion of the RNP routine for the Standard Epoch J2000 This is form of the IAU-76/FK5 RNP model, as implemented in Jeod 1.52.

Definition in file [include/nutation\\_j2000.hh](#).

## 9.9 nutation\_j2000\_init.cc File Reference

Implementation of NutationJ2000Init.

```
#include <cstdlib>
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/nutation_j2000_init.hh"
```

### Namespaces

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

### 9.9.1 Detailed Description

Implementation of NutationJ2000Init.

Definition in file [nutation\\_j2000\\_init.cc](#).

## 9.10 nutation\_j2000\_init.hh File Reference

Used to load data and initialize the NutationJ2000 class through NutationJ2000::initialize.

```
#include "environment/RNP/GenericRNP/include/planet_rotation_init.hh"
#include "utils/sim_interface/include/jeod_class.hh"
```

### Data Structures

- class [jeod::NutationJ2000Init](#)

The *NutationJ2000Init* contains coefficients and other data used to initialize a *NutationJ2000* object.

### Namespaces

- [jeod](#)

Namespace *jeod*.

### 9.10.1 Detailed Description

Used to load data and initialize the NutationJ2000 class through NutationJ2000::initialize.

Definition in file [nutation\\_j2000\\_init.hh](#).

## 9.11 nutation\_mars.cc File Reference

Implementation for the NutationMars class.

```
#include <cstdlib>
#include <cmath>
#include "environment/RNP/GenericRNP/include/RNP_messages.hh"
#include "utils/math/include/matrix3x3.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/nutation_mars.hh"
```

### Namespaces

- [jeod](#)

Namespace *jeod*.

### 9.11.1 Detailed Description

Implementation for the NutationMars class.

Definition in file [nutation\\_mars.cc](#).

## 9.12 nutation\_mars.hh File Reference

Represent the nutation portion of the JPL-developed "Pathfinder" Mars orientation model.

```
#include "environment/RNP/GenericRNP/include/planet_rotation.hh"
#include "utils/sim_interface/include/jeod_class.hh"
```

### Data Structures

- class [jeod::NutationMars](#)

*Implements the nutation portion of the "Pathfinder" Mars RNP model.*

### Namespaces

- [jeod](#)

*Namespace jeod.*

#### 9.12.1 Detailed Description

Represent the nutation portion of the JPL-developed "Pathfinder" Mars orientation model.

Definition in file [nutation\\_mars.hh](#).

## 9.13 planet\_orientation.cc File Reference

Implement PlanetOrientation.

```
#include <cstdint>
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "environment/planet/include/planet.hh"
#include "utils/math/include/matrix3x3.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/planet_orientation.hh"
#include "../include/RNP_messages.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

#### 9.13.1 Detailed Description

Implement PlanetOrientation.

Definition in file [planet\\_orientation.cc](#).

## 9.14 planet\_orientation.hh File Reference

Establish a pure virtual framework for interfacing with the DynManager to set a planet orientation, meaning the transformation information from a planet's inertial frame to it's planet fixed frame.



```
#include <string>
#include <utility>
#include "utils/sim_interface/include/jeod_class.hh"
#include "utils/ref_frames/include/ref_frame_interface.hh"
#include "environment/ephemerides/ephem_interface/include/ephem_interface.-
hh"
#include "environment/ephemerides/ephem_item/include/ephem_orient.hh"
```

## Data Structures

- class [jeod::PlanetOrientation](#)

*The generic framework for orientation models to interact with a DynManager object.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.14.1 Detailed Description

Establish a pure virtual framework for interfacing with the DynManager to set a planet orientation, meaning the transformation information from a planet's inertial frame to it's planet fixed frame.

Definition in file [planet\\_orientation.hh](#).

## 9.15 planet\_rnp.cc File Reference

Implement PlanetRNP.

```
#include <cstdint>
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "environment/planet/include/planet.hh"
#include "utils/math/include/matrix3x3.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/planet_rnp.hh"
#include "../include/planet_rotation.hh"
#include "../include/RNP_messages.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.15.1 Detailed Description

Implement PlanetRNP.

Definition in file [planet\\_rnp.cc](#).

## 9.16 planet\_rnp.hh File Reference

Establish a framework for rotation-nutation-precession style planet attitude models.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "planet_orientation.hh"
```

### Data Structures

- class [jeod::PlanetRNP](#)  
*The generic framework for orientation models based on the RNP paradigm.*

### Namespaces

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

#### 9.16.1 Detailed Description

Establish a framework for rotation-nutation-precession style planet attitude models.

Definition in file [planet\\_rnp.hh](#).

## 9.17 planet\_rotation.cc File Reference

Implementation for PlanetRotation.

```
#include <cmath>
#include "utils/math/include/matrix3x3.hh"
#include "../include/planet_rotation.hh"
#include "../include/planet_rotation_init.hh"
```

### Namespaces

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

#### 9.17.1 Detailed Description

Implementation for PlanetRotation.

Definition in file [planet\\_rotation.cc](#).

## 9.18 planet\_rotation.hh File Reference

Pure virtual polymorphic base class for all forms of planet rotation in the RNP model including precession, nutation, polar motion and axial rotation.

```
#include <math.h>
#include "utils/sim_interface/include/jeod_class.hh"
```

## Data Structures

- class [jeod::PlanetRotation](#)  
*The generic base class for all planet transformations.*

## Namespaces

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

### 9.18.1 Detailed Description

Pure virtual polymorphic base class for all forms of planet rotation in th RNP model including precession, nutation, polar motion and axial rotation.

Definition in file [planet\\_rotation.hh](#).

## 9.19 planet\_rotation\_init.cc File Reference

Implementation for PlanetRotationInit.

```
#include "../include/planet_rotation_init.hh"
```

## Namespaces

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

### 9.19.1 Detailed Description

Implementation for PlanetRotationInit.

Definition in file [planet\\_rotation\\_init.cc](#).

## 9.20 planet\_rotation\_init.hh File Reference

Pure virtual polymorphic base class meant to be used by PlanetRotation::initialize when a large amount of data that could possibly change must be used for rotation calculation.

```
#include "utils/sim_interface/include/jeod_class.hh"
```

## Data Structures

- class [jeod::PlanetRotationInit](#)  
*The generic base class for all initializing classes for [PlanetRotation](#) derived classes.*

## Namespaces

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

### 9.20.1 Detailed Description

Pure virtual polymorphic base class meant to be used by PlanetRotation::initialize when a large amount of data that could possibly change must be used for rotation calculation.

Definition in file [planet\\_rotation\\_init.hh](#).

## 9.21 polar\_motion\_j2000.cc File Reference

Implementation of PolarMotionJ2000.

```
#include <cstdint>
#include "environment/RNP/GenericRNP/include/RNP_messages.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/polar_motion_j2000.hh"
#include "../include/polar_motion_j2000_init.hh"
```

### Namespaces

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

### 9.21.1 Detailed Description

Implementation of PolarMotionJ2000.

Definition in file [polar\\_motion\\_j2000.cc](#).

## 9.22 polar\_motion\_j2000.hh File Reference

Model the polar motion portion of the RNP routine for the Standard Epoch J2000.

```
#include "environment/RNP/GenericRNP/include/planet_rotation.hh"
#include "utils/sim_interface/include/jeod_class.hh"
```

### Data Structures

- class [jeod::PolarMotionJ2000](#)  
*Implements the polar motion portion of the J2000 RNP.*

### Namespaces

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

### 9.22.1 Detailed Description

Model the polar motion portion of the RNP routine for the Standard Epoch J2000. This is a form of the IAU-76/FK5 RNP model, as implemented in Jeod 1.52

Definition in file [polar\\_motion\\_j2000.hh](#).

## 9.23 polar\_motion\_j2000\_init.cc File Reference

Implementation for PolarMotionJ2000Init.

```
#include <cstdint>
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/polar_motion_j2000_init.hh"
```

### Namespaces

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

#### 9.23.1 Detailed Description

Implementation for PolarMotionJ2000Init.

Definition in file [polar\\_motion\\_j2000\\_init.cc](#).

## 9.24 polar\_motion\_j2000\_init.hh File Reference

Used to load data and initialize the PolarMotionJ2000 class through PolarMotionJ2000::initialize.

```
#include "environment/RNP/GenericRNP/include/planet_rotation_init.hh"
#include "utils/sim_interface/include/jeod_class.hh"
```

### Data Structures

- class [jeod::PolarMotionJ2000Init](#)  
*Initializes the [PolarMotionJ2000](#) object.*

### Namespaces

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

#### 9.24.1 Detailed Description

Used to load data and initialize the PolarMotionJ2000 class through PolarMotionJ2000::initialize.

Definition in file [polar\\_motion\\_j2000\\_init.hh](#).

## 9.25 precession\_j2000.cc File Reference

Implementation for PrecessionJ2000.

```
#include "../include/precession_j2000.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.25.1 Detailed Description

Implementation for PrecessionJ2000.

Definition in file [precession\\_j2000.cc](#).

## 9.26 precession\_j2000.hh File Reference

Model the precession for the RNP routine for the Standard Epoch J2000 This is a form of the IAU-76/FK5 RNP model, as implemented in Jeod 1.52.

```
#include "environment/RNP/GenericRNP/include/planet_rotation.hh"
#include "utils/sim_interface/include/jeod_class.hh"
```

## Data Structures

- class [jeod::PrecessionJ2000](#)

*Implements the precession portion of the J2000 RNP.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.26.1 Detailed Description

Model the precession for the RNP routine for the Standard Epoch J2000 This is a form of the IAU-76/FK5 RNP model, as implemented in Jeod 1.52.

Definition in file [precession\\_j2000.hh](#).

## 9.27 precession\_mars.cc File Reference

Implementation of PrecessionMars.

```
#include <cstddef>
#include "environment/RNP/GenericRNP/include/RNP_messages.hh"
#include "utils/math/include/matrix3x3.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/precession_mars.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.27.1 Detailed Description

Implementation of PrecessionMars.

Definition in file [precession\\_mars.cc](#).

## 9.28 precession\_mars.hh File Reference

Represent the precession portion of the JPL-developed "Pathfinder" Mars orientation model.

```
#include "environment/RNP/GenericRNP/include/planet_rotation.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "nututation_mars.hh"
```

### Data Structures

- class [jeod::PrecessionMars](#)

*Implements the axial rotation portion of the "Pathfinder" Mars RNP model.*

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.28.1 Detailed Description

Represent the precession portion of the JPL-developed "Pathfinder" Mars orientation model.

Definition in file [precession\\_mars.hh](#).

## 9.29 rnp\_j2000.cc File Reference

Implementation for RNPJ2000.

```
#include <cstdlib>
#include "environment/time/include/time_manager.hh"
#include "environment/time/include/time_tt.hh"
#include "environment/time/include/time_ut1.hh"
#include "environment/time/include/time_gmst.hh"
#include "environment/time/include/time_dyn.hh"
#include "environment/planet/include/planet.hh"
#include "utils/math/include/matrix3x3.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "environment/RNP/GenericRNP/include/RNP_messages.hh"
#include "../include/rnp_j2000.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.29.1 Detailed Description

Implementation for RNPJ2000.

Definition in file [rnp\\_j2000.cc](#).

## 9.30 rnp\_j2000.hh File Reference

### Data Structures

- class [jeod::RNPJ2000\\_rnp\\_j2000\\_default\\_data](#)

### Namespaces

- [jeod](#)

*Namespace jeod.*

## 9.31 rnp\_j2000.hh File Reference

A specific implementation of PlanetRNP, for Earth.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "environment/RNP/GenericRNP/include/planet_rnp.hh"
#include "nutation_j2000.hh"
#include "polar_motion_j2000.hh"
#include "rotation_j2000.hh"
#include "precession_j2000.hh"
```

### Data Structures

- class [jeod::RNPJ2000](#)

*Implements the J2000 RNP model using the generic RNP framework.*

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.31.1 Detailed Description

A specific implementation of PlanetRNP, for Earth. This is a form of the IAU-76/FK5 RNP model, as implemented in Jeod 1.52

Definition in file [include/rnp\\_j2000.hh](#).

## 9.32 rnp\_mars.cc File Reference

Implementation of RNPMars.



```
#include <cstdint>
#include "environment/time/include/time_manager.hh"
#include "environment/time/include/time_tt.hh"
#include "environment/time/include/time_dyn.hh"
#include "environment/planet/include/planet.hh"
#include "environment/RNP/GenericRNP/include/RNP_messages.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/rnp_mars.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.32.1 Detailed Description

Implementation of RNPMars.

Definition in file [rnp\\_mars.cc](#).

## 9.33 rnp\_mars.hh File Reference

### Data Structures

- class [jeod::RNPMars\\_rnp\\_mars\\_default\\_data](#)

## Namespaces

- [jeod](#)

*Namespace jeod.*

## 9.34 rnp\_mars.hh File Reference

A specific implementation of PlanetRNP, for Mars.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "environment/RNP/GenericRNP/include/planet_rnp.hh"
#include "nutation_mars.hh"
#include "rotation_mars.hh"
#include "precession_mars.hh"
```

### Data Structures

- class [jeod::RNPMars](#)

*Implements the "Pathfinder" Mars RNP model using the generic RNP framework.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.34.1 Detailed Description

A specific implementation of PlanetRNP, for Mars. This is a form of the JPL-developed "Pathfinder" Mars orientation model.

Definition in file [include/rnp\\_mars.hh](#).

## 9.35 RNP\_messages.cc File Reference

Implement RNP\_messages.

```
#include "../include/RNP_messages.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

## Macros

- `#define` [PATH](#) "environment/RNP/"

### 9.35.1 Detailed Description

Implement RNP\_messages.

Definition in file [RNP\\_messages.cc](#).

## 9.36 RNP\_messages.hh File Reference

Implement RNP\_messages.

```
#include "utils/sim_interface/include/jeod_class.hh"
```

## Data Structures

- class [jeod::RNPMessages](#)

*Describes messages used in the RNP model.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.36.1 Detailed Description

Implement RNP\_messages.

Definition in file [RNP\\_messages.hh](#).

## 9.37 rotation\_j2000.cc File Reference

Implementation of RotationJ2000.

```
#include <cstdint>
#include <cmath>
#include "utils/message/include/message_handler.hh"
#include "environment/RNP/GenericRNP/include/RNP_messages.hh"
#include "../include/rotation_j2000.hh"
```

### Namespaces

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

### 9.37.1 Detailed Description

Implementation of RotationJ2000.

Definition in file [rotation\\_j2000.cc](#).

## 9.38 rotation\_j2000.hh File Reference

Model the axial rotation portion of the RNP routine for the Standard Epoch J2000.

```
#include "environment/RNP/GenericRNP/include/planet_rotation.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "nutation_j2000.hh"
```

### Data Structures

- class [jeod::RotationJ2000](#)  
*Implements the axial rotation portion of J2000 RNP.*

### Namespaces

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

### 9.38.1 Detailed Description

Model the axial rotation portion of the RNP routine for the Standard Epoch J2000. This is a form of the IAU-76/FK5 RNP model, as implemented in Jeod 1.52

Definition in file [rotation\\_j2000.hh](#).

## 9.39 rotation\_mars.cc File Reference

Implementation of RotationMars.

```
#include <cstdint>
#include <cmath>
#include "environment/RNP/GenericRNP/include/RNP_messages.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/rotation_mars.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.39.1 Detailed Description

Implementation of RotationMars.

Definition in file [rotation\\_mars.cc](#).

## 9.40 rotation\_mars.hh File Reference

Represent the axial rotation portion of the JPL-developed "Pathfinder" Mars orientation model.

```
#include "environment/RNP/GenericRNP/include/planet_rotation.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "nutation_mars.hh"
```

### Data Structures

- class [jeod::RotationMars](#)

*Implements the axial rotation portion of the "Pathfinder" Mars RNP model.*

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.40.1 Detailed Description

Represent the axial rotation portion of the JPL-developed "Pathfinder" Mars orientation model.

Definition in file [rotation\\_mars.hh](#).

## 9.41 xpyp\_daily.cc File Reference

```
#include "environment/RNP/GenericRNP/include/planet_rotation_init.hh"
```

```
#include "environment/RNP/RNPJ2000/include/polar_motion_j2000_init.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/xpyp_daily.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

## Macros

- #define [JEOD\\_FRIEND\\_CLASS](#) PolarMotionJ2000Init\_xpyp\_daily\_default\_data

### 9.41.1 Macro Definition Documentation

#### 9.41.1.1 #define JEOD\_FRIEND\_CLASS PolarMotionJ2000Init\_xpyp\_daily\_default\_data

Definition at line 21 of file xpyp\_daily.cc.

## 9.42 xpyp\_daily.hh File Reference

### Data Structures

- class [jeod::PolarMotionJ2000Init\\_xpyp\\_daily\\_default\\_data](#)

## Namespaces

- [jeod](#)

*Namespace jeod.*

## 9.43 xpyp\_monthly.cc File Reference

```
#include "environment/RNP/GenericRNP/include/planet_rotation_init.hh"
#include "environment/RNP/RNPJ2000/include/polar_motion_j2000_init.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/xpyp_monthly.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

## Macros

- #define [JEOD\\_FRIEND\\_CLASS](#) PolarMotionJ2000Init\_xpyp\_monthly\_default\_data

### 9.43.1 Macro Definition Documentation

#### 9.43.1.1 `#define JEOD_FRIEND_CLASS PolarMotionJ2000Init_xpyp_monthly_default_data`

Definition at line 21 of file `xpyp_monthly.cc`.

## 9.44 `xpyp_monthly.hh` File Reference

### Data Structures

- class [jeod::PolarMotionJ2000Init\\_xpyp\\_monthly\\_default\\_data](#)

### Namespaces

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

# Index

- ~NutationJ2000
  - jeod::NutationJ2000, [21](#)
- ~NutationJ2000Init
  - jeod::NutationJ2000Init, [26](#)
- ~NutationMars
  - jeod::NutationMars, [30](#)
- ~PlanetOrientation
  - jeod::PlanetOrientation, [34](#)
- ~PlanetRNP
  - jeod::PlanetRNP, [39](#)
- ~PlanetRotation
  - jeod::PlanetRotation, [44](#)
- ~PlanetRotationInit
  - jeod::PlanetRotationInit, [48](#)
- ~PolarMotionJ2000
  - jeod::PolarMotionJ2000, [50](#)
- ~PolarMotionJ2000Init
  - jeod::PolarMotionJ2000Init, [53](#)
- ~PrecessionJ2000
  - jeod::PrecessionJ2000, [57](#)
- ~PrecessionMars
  - jeod::PrecessionMars, [59](#)
- ~RNPJ2000
  - jeod::RNPJ2000, [63](#)
- ~RNPMars
  - jeod::RNPMars, [69](#)
- ~RotationJ2000
  - jeod::RotationJ2000, [77](#)
- ~RotationMars
  - jeod::RotationMars, [79](#)
- activate
  - jeod::PlanetOrientation, [35](#)
- active
  - jeod::PlanetOrientation, [36](#)
- class\_declarations.hh, [83](#)
- compute\_fixed\_matrices
  - jeod::PrecessionMars, [59](#)
- ConstantNP
  - jeod::PlanetRNP, [39](#)
- current\_time
  - jeod::PlanetRotation, [45](#)
- D
  - jeod::NutationJ2000, [22](#)
- D\_coeffs
  - jeod::NutationJ2000, [22](#)
  - jeod::NutationJ2000Init, [26](#)
- DAYTOJULIANCENT
  - jeod::PlanetRotation, [45](#)
- DEGTORAD
  - jeod::PlanetRotation, [46](#)
- DEGTOTSEC
  - jeod::PlanetRotation, [46](#)
- data\_nutation\_j2000.cc, [83](#)
- data\_rnp\_j2000.cc, [84](#)
- data\_rnp\_mars.cc, [85](#)
- deactivate
  - jeod::PlanetOrientation, [35](#)
- enable\_polar
  - jeod::PlanetRNP, [41](#)
- Environment, [12](#)
- ephem\_activate
  - jeod::PlanetOrientation, [35](#)
- ephem\_build\_tree
  - jeod::PlanetOrientation, [35](#)
- ephem\_initialize
  - jeod::PlanetOrientation, [35](#)
- ephem\_update
  - jeod::RNPJ2000, [63](#)
  - jeod::RNPMars, [69](#)
- epsilon\_bar
  - jeod::NutationJ2000, [22](#)
- equa\_of\_equi
  - jeod::NutationJ2000, [22](#)
- F
  - jeod::NutationJ2000, [22](#)
- F\_coeffs
  - jeod::NutationJ2000, [22](#)
  - jeod::NutationJ2000Init, [27](#)
- fidelity\_error
  - jeod::RNPMessages, [74](#)
- FullRNP
  - jeod::PlanetRNP, [39](#)
- GMST
  - jeod::RotationJ2000, [77](#)
- GenericRNP, [14](#)
- PATH, [14](#)
- get\_dyn\_time\_ptr
  - jeod::RNPJ2000, [63](#)
  - jeod::RNPMars, [70](#)
- get\_name
  - jeod::PlanetOrientation, [35](#)
  - jeod::PlanetRNP, [40](#)
  - jeod::RNPJ2000, [63](#)
  - jeod::RNPMars, [70](#)

- get\_rotation
  - jeod::PlanetRotation, [44](#)
- get\_rotation\_transpose
  - jeod::PlanetRotation, [44](#)
- gmst\_ptr
  - jeod::RNPJ2000, [65](#)
- I\_at\_j2000
  - jeod::NutationMars, [31](#)
- I\_dot
  - jeod::NutationMars, [31](#)
- I\_m\_orig
  - jeod::NutationMars, [31](#)
- init\_attrjeod\_\_NutationJ2000
  - jeod::NutationJ2000, [21](#)
- init\_attrjeod\_\_NutationJ2000Init
  - jeod::NutationJ2000Init, [26](#)
- init\_attrjeod\_\_NutationMars
  - jeod::NutationMars, [31](#)
- init\_attrjeod\_\_PlanetOrientation
  - jeod::PlanetOrientation, [36](#)
- init\_attrjeod\_\_PlanetRNP
  - jeod::PlanetRNP, [41](#)
- init\_attrjeod\_\_PlanetRotation
  - jeod::PlanetRotation, [45](#)
- init\_attrjeod\_\_PlanetRotationInit
  - jeod::PlanetRotationInit, [48](#)
- init\_attrjeod\_\_PolarMotionJ2000
  - jeod::PolarMotionJ2000, [51](#)
- init\_attrjeod\_\_PolarMotionJ2000Init
  - jeod::PolarMotionJ2000Init, [54](#)
- init\_attrjeod\_\_PrecessionJ2000
  - jeod::PrecessionJ2000, [58](#)
- init\_attrjeod\_\_PrecessionMars
  - jeod::PrecessionMars, [60](#)
- init\_attrjeod\_\_RNPJ2000
  - jeod::RNPJ2000, [65](#)
- init\_attrjeod\_\_RNPMars
  - jeod::RNPMars, [71](#)
- init\_attrjeod\_\_RNPMessages
  - jeod::RNPMessages, [74](#)
- init\_attrjeod\_\_RotationJ2000
  - jeod::RotationJ2000, [77](#)
- init\_attrjeod\_\_RotationMars
  - jeod::RotationMars, [80](#)
- initialization\_error
  - jeod::RNPMessages, [75](#)
- initialize
  - jeod::NutationJ2000, [21](#)
  - jeod::NutationJ2000Init\_nutation\_j2000\_default\_data, [29](#)
  - jeod::PlanetOrientation, [36](#)
  - jeod::PlanetRotation, [44](#)
  - jeod::PolarMotionJ2000, [50](#)
  - jeod::PolarMotionJ2000Init\_xpyp\_daily\_default\_data, [55](#)
  - jeod::PolarMotionJ2000Init\_xpyp\_monthly\_default\_data, [56](#)
  - jeod::RNPJ2000, [64](#)
  - jeod::RNPJ2000\_rnp\_j2000\_default\_data, [67](#)
  - jeod::RNPMars, [70](#)
  - jeod::RNPMars\_rnp\_mars\_default\_data, [73](#)
- InputProcessor
  - jeod::NutationJ2000, [21](#)
  - jeod::NutationJ2000Init, [26](#)
  - jeod::NutationMars, [31](#)
  - jeod::PlanetOrientation, [36](#)
  - jeod::PlanetRNP, [41](#)
  - jeod::PlanetRotation, [45](#)
  - jeod::PlanetRotationInit, [48](#)
  - jeod::PolarMotionJ2000, [51](#)
  - jeod::PolarMotionJ2000Init, [54](#)
  - jeod::PrecessionJ2000, [58](#)
  - jeod::PrecessionMars, [60](#)
  - jeod::RNPJ2000, [65](#)
  - jeod::RNPMars, [71](#)
  - jeod::RNPMessages, [74](#)
  - jeod::RotationJ2000, [77](#)
  - jeod::RotationMars, [80](#)
- int\_to\_double
  - jeod::NutationMars, [31](#)
- internal\_name
  - jeod::RNPJ2000, [65](#)
  - jeod::RNPMars, [71](#)
- J
  - jeod::PrecessionMars, [60](#)
- JEOD\_FRIEND\_CLASS
  - data\_nutation\_j2000.cc, [84](#)
  - data\_rnp\_j2000.cc, [84](#)
  - data\_rnp\_mars.cc, [85](#)
  - xpyp\_daily.cc, [101](#)
  - xpyp\_monthly.cc, [102](#)
- JULIANCENTTODAY
  - jeod::PlanetRotation, [46](#)
- JULIANDAYTOSEC
  - jeod::PlanetRotation, [46](#)
- jeod, [17](#)
- jeod::PlanetRNP
  - ConstantNP, [39](#)
  - FullRNP, [39](#)
  - RotationOnly, [39](#)
- jeod::NutationJ2000, [19](#)
- ~NutationJ2000, [21](#)
- D, [22](#)
- D\_coeffs, [22](#)
- epsilon\_bar, [22](#)
- equa\_of\_equi, [22](#)
- F, [22](#)
- F\_coeffs, [22](#)
- init\_attrjeod\_\_NutationJ2000, [21](#)
- initialize, [21](#)
- InputProcessor, [21](#)
- L, [23](#)
- L\_coeffs, [23](#)
- long\_coeffs, [23](#)
- long\_t\_coeffs, [23](#)
- M, [23](#)



- M\_coeffs, 23
- num\_coeffs, 23
- nutation\_in\_longitude, 24
- nutation\_in\_obliquity, 24
- NutationJ2000, 21
- obliq\_coeffs, 24
- obliq\_t\_coeffs, 24
- omega, 24
- omega\_coeffs, 24
- operator=, 21
- update\_rotation, 21
- jeod::NutationJ2000Init, 25
  - ~NutationJ2000Init, 26
  - D\_coeffs, 26
  - F\_coeffs, 27
  - init\_attrjeod\_\_NutationJ2000Init, 26
  - InputProcessor, 26
  - L\_coeffs, 27
  - long\_coeffs, 27
  - long\_t\_coeffs, 27
  - M\_coeffs, 27
  - num\_coeffs, 27
  - NutationJ2000Init, 26
  - obliq\_coeffs, 28
  - obliq\_t\_coeffs, 28
  - omega\_coeffs, 28
  - operator=, 26
- jeod::NutationJ2000Init\_nutation\_j2000\_default\_data, 28
  - initialize, 29
- jeod::NutationMars, 29
  - ~NutationMars, 30
  - I\_at\_j2000, 31
  - I\_dot, 31
  - I\_m\_orig, 31
  - init\_attrjeod\_\_NutationMars, 31
  - InputProcessor, 31
  - int\_to\_double, 31
  - mean\_anomaly\_j2000, 31
  - mean\_motion, 32
  - nutation\_in\_longitude, 32
  - nutation\_in\_obliquity, 32
  - NutationMars, 30
  - obliquity\_angle, 32
  - operator=, 30
  - psi\_m\_orig, 32
  - q\_angle\_j2000, 32
  - update\_rotation, 30
- jeod::PlanetOrientation, 33
  - ~PlanetOrientation, 34
  - activate, 35
  - active, 36
  - deactivate, 35
  - ephem\_activate, 35
  - ephem\_build\_tree, 35
  - ephem\_initialize, 35
  - get\_name, 35
  - init\_attrjeod\_\_PlanetOrientation, 36
  - initialize, 36
  - InputProcessor, 36
  - name, 37
  - operator=, 36
  - orient\_interface, 37
  - planet, 37
  - planet\_omega, 37
  - planet\_rot\_state, 37
  - PlanetOrientation, 34
  - set\_name, 36
- jeod::PlanetRNP, 38
  - ~PlanetRNP, 39
  - enable\_polar, 41
  - get\_name, 40
  - init\_attrjeod\_\_PlanetRNP, 41
  - InputProcessor, 41
  - NP\_matrix, 41
  - nutation, 41
  - operator=, 40
  - PlanetRNP, 39, 40
  - polar\_motion, 41
  - precession, 41
  - propagate\_rnp, 40
  - RNPFidelity, 39
  - rnp\_type, 42
  - rotation, 42
  - scratch\_matrix, 42
  - update\_axial\_rotation, 40
  - update\_rnp, 40
- jeod::PlanetRotation, 42
  - ~PlanetRotation, 44
  - current\_time, 45
  - DAYTOJULIANCENT, 45
  - DEGTORAD, 46
  - DEGTOSEC, 46
  - get\_rotation, 44
  - get\_rotation\_transpose, 44
  - init\_attrjeod\_\_PlanetRotation, 45
  - initialize, 44
  - InputProcessor, 45
  - JULIANCENTTODAY, 46
  - JULIANDAYTOSEC, 46
  - operator=, 45
  - PlanetRotation, 44
  - RADTODEG, 46
  - rotation, 46
  - SECTODEG, 47
  - SECTOJULIANDAY, 47
  - update\_rotation, 45
  - update\_time, 45
- jeod::PlanetRotationInit, 47
  - ~PlanetRotationInit, 48
  - init\_attrjeod\_\_PlanetRotationInit, 48
  - InputProcessor, 48
  - operator=, 48
  - PlanetRotationInit, 48
- jeod::PolarMotionJ2000, 48
  - ~PolarMotionJ2000, 50

- init\_attrjeod\_\_PolarMotionJ2000, 51
- initialize, 50
- InputProcessor, 51
- last\_table\_index, 51
- operator=, 50
- override\_table, 51
- polar\_mjd, 51
- PolarMotionJ2000, 50
- update\_rotation, 50
- warn\_table, 51
- xp, 51
- xp\_tbl, 51
- yp, 52
- yp\_tbl, 52
- jeod::PolarMotionJ2000Init, 52
  - ~PolarMotionJ2000Init, 53
  - init\_attrjeod\_\_PolarMotionJ2000Init, 54
  - InputProcessor, 54
  - last\_table\_index, 54
  - operator=, 54
  - override\_table, 54
  - polar\_mjd, 54
  - PolarMotionJ2000Init, 53
  - xp, 54
  - xp\_tbl, 54
  - yp, 55
  - yp\_tbl, 55
- jeod::PolarMotionJ2000Init\_xpyp\_daily\_default\_data, 55
  - initialize, 55
- jeod::PolarMotionJ2000Init\_xpyp\_monthly\_default\_data, 56
  - initialize, 56
- jeod::PrecessionJ2000, 56
  - ~PrecessionJ2000, 57
  - init\_attrjeod\_\_PrecessionJ2000, 58
  - InputProcessor, 58
  - operator=, 57
  - PrecessionJ2000, 57
  - update\_rotation, 57
- jeod::PrecessionMars, 58
  - ~PrecessionMars, 59
  - compute\_fixed\_matrices, 59
  - init\_attrjeod\_\_PrecessionMars, 60
  - InputProcessor, 60
  - J, 60
  - N, 60
  - NJ\_matrix, 60
  - nutaton, 60
  - operator=, 59
  - PrecessionMars, 59
  - psi\_at\_j2000, 60
  - psi\_dot, 61
  - psi\_precess, 61
  - update\_rotation, 60
- jeod::RNPNJ2000, 61
  - ~RNPNJ2000, 63
  - ephem\_update, 63
  - get\_dyn\_time\_ptr, 63
  - get\_name, 63
  - gmst\_ptr, 65
  - init\_attrjeod\_\_RNPNJ2000, 65
  - initialize, 64
  - InputProcessor, 65
  - internal\_name, 65
  - last\_updated\_time\_full, 65
  - last\_updated\_time\_rotational, 65
  - NJ2000, 66
  - never\_updated\_full, 66
  - never\_updated\_rotational, 66
  - operator=, 64
  - PJ2000, 66
  - PMJ2000, 66
  - RJ2000, 66
  - RNPNJ2000, 63
  - time\_dyn\_ptr, 67
  - timestamp, 64
  - update\_axial\_rotation, 64
  - update\_rnp, 64
- jeod::RNPNJ2000\_rnp\_j2000\_default\_data, 67
  - initialize, 67
- jeod::RNPMars, 67
  - ~RNPMars, 69
  - ephem\_update, 69
  - get\_dyn\_time\_ptr, 70
  - get\_name, 70
  - init\_attrjeod\_\_RNPMars, 71
  - initialize, 70
  - InputProcessor, 71
  - internal\_name, 71
  - last\_updated\_time\_full, 71
  - last\_updated\_time\_rotational, 71
  - NMars, 72
  - never\_updated\_full, 72
  - never\_updated\_rotational, 72
  - operator=, 70
  - PMars, 72
  - RMars, 72
  - RNPMars, 69
  - time\_dyn\_ptr, 72
  - timestamp, 70
  - tt\_ptr, 73
  - update\_axial\_rotation, 70
  - update\_rnp, 71
- jeod::RNPMars\_rnp\_mars\_default\_data, 73
  - initialize, 73
- jeod::RNPMessages, 73
  - fidelity\_error, 74
  - init\_attrjeod\_\_RNPMessages, 74
  - initialization\_error, 75
  - InputProcessor, 74
  - operator=, 74
  - polar\_motion\_table\_warning, 75
  - RNPMessages, 74
  - setup\_error, 75
- jeod::RotationJ2000, 76
  - ~RotationJ2000, 77

- GMST, [77](#)
- init\_attrjeod\_\_RotationJ2000, [77](#)
- InputProcessor, [77](#)
- nutaton, [77](#)
- operator=, [77](#)
- planet\_rotational\_velocity, [78](#)
- RotationJ2000, [77](#)
- theta\_gast, [78](#)
- update\_rotation, [77](#)
- use\_full\_rnp, [78](#)
- jeod::RotationMars, [78](#)
- ~RotationMars, [79](#)
- init\_attrjeod\_\_RotationMars, [80](#)
- InputProcessor, [80](#)
- nutaton, [80](#)
- operator=, [80](#)
- phi\_at\_j2000, [80](#)
- phi\_spin, [80](#)
- planet\_rotational\_velocity, [80](#)
- RotationMars, [79](#)
- update\_rotation, [80](#)
- use\_full\_rnp, [81](#)
- L
  - jeod::NutationJ2000, [23](#)
- L\_coeffs
  - jeod::NutationJ2000, [23](#)
  - jeod::NutationJ2000Init, [27](#)
- last\_table\_index
  - jeod::PolarMotionJ2000, [51](#)
  - jeod::PolarMotionJ2000Init, [54](#)
- last\_updated\_time\_full
  - jeod::RNPJ2000, [65](#)
  - jeod::RNPMars, [71](#)
- last\_updated\_time\_rotational
  - jeod::RNPJ2000, [65](#)
  - jeod::RNPMars, [71](#)
- long\_coeffs
  - jeod::NutationJ2000, [23](#)
  - jeod::NutationJ2000Init, [27](#)
- long\_t\_coeffs
  - jeod::NutationJ2000, [23](#)
  - jeod::NutationJ2000Init, [27](#)
- M
  - jeod::NutationJ2000, [23](#)
- M\_coeffs
  - jeod::NutationJ2000, [23](#)
  - jeod::NutationJ2000Init, [27](#)
- mean\_anomaly\_j2000
  - jeod::NutationMars, [31](#)
- mean\_motion
  - jeod::NutationMars, [32](#)
- Models, [11](#)
- N
  - jeod::PrecessionMars, [60](#)
- NJ2000
  - jeod::RNPJ2000, [66](#)
- NJ\_matrix
  - jeod::PrecessionMars, [60](#)
- NMars
  - jeod::RNPMars, [72](#)
- NP\_matrix
  - jeod::PlanetRNP, [41](#)
- name
  - jeod::PlanetOrientation, [37](#)
- never\_updated\_full
  - jeod::RNPJ2000, [66](#)
  - jeod::RNPMars, [72](#)
- never\_updated\_rotational
  - jeod::RNPJ2000, [66](#)
  - jeod::RNPMars, [72](#)
- num\_coeffs
  - jeod::NutationJ2000, [23](#)
  - jeod::NutationJ2000Init, [27](#)
- nutaton
  - jeod::PlanetRNP, [41](#)
  - jeod::PrecessionMars, [60](#)
  - jeod::RotationJ2000, [77](#)
  - jeod::RotationMars, [80](#)
- nutaton\_in\_longitude
  - jeod::NutationJ2000, [24](#)
  - jeod::NutationMars, [32](#)
- nutaton\_in\_obliquity
  - jeod::NutationJ2000, [24](#)
  - jeod::NutationMars, [32](#)
- nutaton\_j2000.cc, [85](#)
- nutaton\_j2000.hh, [86](#)
- nutaton\_j2000\_init.cc, [86](#)
- nutaton\_j2000\_init.hh, [87](#)
- nutaton\_mars.cc, [87](#)
- nutaton\_mars.hh, [88](#)
- NutationJ2000
  - jeod::NutationJ2000, [21](#)
- NutationJ2000Init
  - jeod::NutationJ2000Init, [26](#)
- NutationMars
  - jeod::NutationMars, [30](#)
- obliq\_coeffs
  - jeod::NutationJ2000, [24](#)
  - jeod::NutationJ2000Init, [28](#)
- obliq\_t\_coeffs
  - jeod::NutationJ2000, [24](#)
  - jeod::NutationJ2000Init, [28](#)
- obliquity\_angle
  - jeod::NutationMars, [32](#)
- omega
  - jeod::NutationJ2000, [24](#)
- omega\_coeffs
  - jeod::NutationJ2000, [24](#)
  - jeod::NutationJ2000Init, [28](#)
- operator=
  - jeod::NutationJ2000, [21](#)
  - jeod::NutationJ2000Init, [26](#)
  - jeod::NutationMars, [30](#)
  - jeod::PlanetOrientation, [36](#)

- jeod::PlanetRNP, 40
- jeod::PlanetRotation, 45
- jeod::PlanetRotationInit, 48
- jeod::PolarMotionJ2000, 50
- jeod::PolarMotionJ2000Init, 54
- jeod::PrecessionJ2000, 57
- jeod::PrecessionMars, 59
- jeod::RNPJ2000, 64
- jeod::RNPMars, 70
- jeod::RNPMessages, 74
- jeod::RotationJ2000, 77
- jeod::RotationMars, 80
- orient\_interface
  - jeod::PlanetOrientation, 37
- override\_table
  - jeod::PolarMotionJ2000, 51
  - jeod::PolarMotionJ2000Init, 54
- PATH
  - GenericRNP, 14
- PJ2000
  - jeod::RNPJ2000, 66
- PMJ2000
  - jeod::RNPJ2000, 66
- PMars
  - jeod::RNPMars, 72
- phi\_at\_j2000
  - jeod::RotationMars, 80
- phi\_spin
  - jeod::RotationMars, 80
- planet
  - jeod::PlanetOrientation, 37
- planet\_omega
  - jeod::PlanetOrientation, 37
- planet\_orientation.cc, 88
- planet\_orientation.hh, 88
- planet\_rnp.cc, 89
- planet\_rnp.hh, 90
- planet\_rot\_state
  - jeod::PlanetOrientation, 37
- planet\_rotation.cc, 90
- planet\_rotation.hh, 90
- planet\_rotation\_init.cc, 91
- planet\_rotation\_init.hh, 91
- planet\_rotational\_velocity
  - jeod::RotationJ2000, 78
  - jeod::RotationMars, 80
- PlanetOrientation
  - jeod::PlanetOrientation, 34
- PlanetRNP
  - jeod::PlanetRNP, 39, 40
- PlanetRotation
  - jeod::PlanetRotation, 44
- PlanetRotationInit
  - jeod::PlanetRotationInit, 48
- polar\_mjd
  - jeod::PolarMotionJ2000, 51
  - jeod::PolarMotionJ2000Init, 54
- polar\_motion
  - jeod::PlanetRNP, 41
  - polar\_motion\_j2000.cc, 92
  - polar\_motion\_j2000.hh, 92
  - polar\_motion\_j2000\_init.cc, 93
  - polar\_motion\_j2000\_init.hh, 93
  - polar\_motion\_table\_warning
    - jeod::RNPMessages, 75
  - PolarMotionJ2000
    - jeod::PolarMotionJ2000, 50
  - PolarMotionJ2000Init
    - jeod::PolarMotionJ2000Init, 53
  - precession
    - jeod::PlanetRNP, 41
  - precession\_j2000.cc, 93
  - precession\_j2000.hh, 94
  - precession\_mars.cc, 94
  - precession\_mars.hh, 95
  - PrecessionJ2000
    - jeod::PrecessionJ2000, 57
  - PrecessionMars
    - jeod::PrecessionMars, 59
  - propagate\_rnp
    - jeod::PlanetRNP, 40
  - psi\_at\_j2000
    - jeod::PrecessionMars, 60
  - psi\_dot
    - jeod::PrecessionMars, 61
  - psi\_m\_orig
    - jeod::NutationMars, 32
  - psi\_precess
    - jeod::PrecessionMars, 61
  - q\_angle\_j2000
    - jeod::NutationMars, 32
- RADTODEG
  - jeod::PlanetRotation, 46
- RJ2000
  - jeod::RNPJ2000, 66
- RMars
  - jeod::RNPMars, 72
- RNP, 13
- RNP\_messages.cc, 98
- RNP\_messages.hh, 98
- RNPFidelity
  - jeod::PlanetRNP, 39
- RNPJ2000, 15
  - jeod::RNPJ2000, 63
- RNPMars, 16
  - jeod::RNPMars, 69
- RNPMessages
  - jeod::RNPMessages, 74
- rnp\_j2000.cc, 95
- rnp\_j2000.hh, 96
- rnp\_mars.cc, 96
- rnp\_mars.hh, 97
- rnp\_type
  - jeod::PlanetRNP, 42
- rotation

- jeod::PlanetRNP, [42](#)
- jeod::PlanetRotation, [46](#)
- RotationOnly
  - jeod::PlanetRNP, [39](#)
- rotation\_j2000.cc, [99](#)
- rotation\_j2000.hh, [99](#)
- rotation\_mars.cc, [100](#)
- rotation\_mars.hh, [100](#)
- RotationJ2000
  - jeod::RotationJ2000, [77](#)
- RotationMars
  - jeod::RotationMars, [79](#)
- SECTODEG
  - jeod::PlanetRotation, [47](#)
- SECTOJULIANDAY
  - jeod::PlanetRotation, [47](#)
- scratch\_matrix
  - jeod::PlanetRNP, [42](#)
- set\_name
  - jeod::PlanetOrientation, [36](#)
- setup\_error
  - jeod::RNPMessages, [75](#)
- theta\_gast
  - jeod::RotationJ2000, [78](#)
- time\_dyn\_ptr
  - jeod::RNPJ2000, [67](#)
  - jeod::RNPMars, [72](#)
- timestamp
  - jeod::RNPJ2000, [64](#)
  - jeod::RNPMars, [70](#)
- tt\_ptr
  - jeod::RNPMars, [73](#)
- update\_axial\_rotation
  - jeod::PlanetRNP, [40](#)
  - jeod::RNPJ2000, [64](#)
  - jeod::RNPMars, [70](#)
- update\_rnp
  - jeod::PlanetRNP, [40](#)
  - jeod::RNPJ2000, [64](#)
  - jeod::RNPMars, [71](#)
- update\_rotation
  - jeod::NutationJ2000, [21](#)
  - jeod::NutationMars, [30](#)
  - jeod::PlanetRotation, [45](#)
  - jeod::PolarMotionJ2000, [50](#)
  - jeod::PrecessionJ2000, [57](#)
  - jeod::PrecessionMars, [60](#)
  - jeod::RotationJ2000, [77](#)
  - jeod::RotationMars, [80](#)
- update\_time
  - jeod::PlanetRotation, [45](#)
- use\_full\_rnp
  - jeod::RotationJ2000, [78](#)
  - jeod::RotationMars, [81](#)
- warn\_table
  - jeod::PolarMotionJ2000, [51](#)
- xp
  - jeod::PolarMotionJ2000, [51](#)
  - jeod::PolarMotionJ2000Init, [54](#)
- xp\_tbl
  - jeod::PolarMotionJ2000, [51](#)
  - jeod::PolarMotionJ2000Init, [54](#)
- xpyp\_daily.cc, [100](#)
- JEOD\_FRIEND\_CLASS, [101](#)
- xpyp\_daily.hh, [101](#)
- xpyp\_monthly.cc, [101](#)
- JEOD\_FRIEND\_CLASS, [102](#)
- xpyp\_monthly.hh, [102](#)
- yp
  - jeod::PolarMotionJ2000, [52](#)
  - jeod::PolarMotionJ2000Init, [55](#)
- yp\_tbl
  - jeod::PolarMotionJ2000, [52](#)
  - jeod::PolarMotionJ2000Init, [55](#)