

GMAT Source

Generated by Doxygen 1.8.13

Contents

1	Test List	1
2	Todo List	3
3	Namespace Index	5
3.1	Namespace List	5
4	Hierarchical Index	7
4.1	Class Hierarchy	7
5	Class Index	9
5.1	Class List	9
6	File Index	11
6.1	File List	11
7	Namespace Documentation	15
7.1	BodyFixedStateConverterUtil Namespace Reference	15
7.1.1	Function Documentation	15
7.1.1.1	CartesianToSpherical()	15
7.1.1.2	CartesianToSphericalEllipsoid()	16
7.1.1.3	Convert() [1/2]	16
7.1.1.4	Convert() [2/2]	17
7.1.1.5	GetValidRepresentations()	17
7.1.1.6	IsValidStateRepresentation()	17
7.1.1.7	SphericalEllipsoidToCartesian()	18

7.1.1.8	SphericalEllipsoidToSpherical()	19
7.1.1.9	SphericalToCartesian()	19
7.1.1.10	SphericalToSphericalEllipsoid()	20
7.2	Gmat Namespace Reference	20
7.2.1	Typedef Documentation	21
7.2.1.1	PLUGIN_RESOURCE	21
7.2.2	Enumeration Type Documentation	21
7.2.2.1	MessageType	21
7.2.2.2	ObjectType	22
7.2.2.3	ParameterType	23
7.2.2.4	RunState	24
7.2.2.5	StateElementId	25
7.2.2.6	WrapperDataType	25
7.2.2.7	WriteMode	25
7.3	GmatAttitudeConstants Namespace Reference	26
7.3.1	Variable Documentation	26
7.3.1.1	DCM_ORTHONORMALITY_TOLERANCE	26
7.3.1.2	EULER_ANGLE_TOLERANCE	26
7.3.1.3	QUAT_MIN_MAG	26
7.4	GmatEop Namespace Reference	26
7.4.1	Enumeration Type Documentation	26
7.4.1.1	EopFileType	26
7.5	GmatFile Namespace Reference	27
7.5.1	Detailed Description	27
7.6	GmatFileUtil Namespace Reference	27
7.6.1	Detailed Description	28
7.6.2	Function Documentation	28
7.6.2.1	CompareLines()	28
7.6.2.2	CompareNumericColumns() [1/2]	29
7.6.2.3	CompareNumericColumns() [2/2]	29

7.6.2.4	CompareNumericLines() [1/2]	29
7.6.2.5	CompareNumericLines() [2/2]	29
7.6.2.6	CompareTextLines() [1/2]	30
7.6.2.7	CompareTextLines() [2/2]	30
7.6.2.8	ConvertToOsFileName()	30
7.6.2.9	DoesDirectoryExist() [1/2]	30
7.6.2.10	DoesDirectoryExist() [2/2]	30
7.6.2.11	DoesFileExist() [1/2]	31
7.6.2.12	DoesFileExist() [2/2]	31
7.6.2.13	GetApplicationPath()	31
7.6.2.14	GetCurrentWorkingDirectory()	31
7.6.2.15	GetFileListFromDirectory()	31
7.6.2.16	GetFunctionOutputTypes()	31
7.6.2.17	GetGmatPath()	32
7.6.2.18	GetInvalidFileNameMessage()	32
7.6.2.19	GetLine()	32
7.6.2.20	GetPathSeparator()	32
7.6.2.21	GetRealColumns()	32
7.6.2.22	GetTemporaryDirectory()	33
7.6.2.23	GetTextLines()	33
7.6.2.24	HasNoPath()	33
7.6.2.25	IsAppInstalled()	33
7.6.2.26	IsAsciiFile()	33
7.6.2.27	IsOsWindows()	34
7.6.2.28	IsPathAbsolute()	34
7.6.2.29	IsPathRelative() [1/2]	34
7.6.2.30	IsPathRelative() [2/2]	34
7.6.2.31	IsSameFileName() [1/2]	35
7.6.2.32	IsSameFileName() [2/2]	35
7.6.2.33	IsValidFileName()	35

7.6.2.34	ParseFileExtension() [1/2]	35
7.6.2.35	ParseFileExtension() [2/2]	35
7.6.2.36	ParseFileName() [1/2]	35
7.6.2.37	ParseFileName() [2/2]	36
7.6.2.38	ParseFirstPathName()	36
7.6.2.39	ParsePathName() [1/2]	36
7.6.2.40	ParsePathName() [2/2]	36
7.6.2.41	PrepareCompare()	36
7.6.2.42	SetCurrentWorkingDirectory()	37
7.6.2.43	SkipHeaderLines()	37
7.6.3	Variable Documentation	37
7.6.3.1	BUFFER_SIZE	37
7.7	GmatIntegerConstants Namespace Reference	37
7.7.1	Variable Documentation	37
7.7.1.1	INTEGER_UNDEFINED	37
7.8	GmatMathConstants Namespace Reference	37
7.8.1	Enumeration Type Documentation	37
7.8.1.1	SIGN	37
7.9	GmatMathUtil Namespace Reference	38
7.9.1	Function Documentation	39
7.9.1.1	Abs() [1/2]	39
7.9.1.2	Abs() [2/2]	39
7.9.1.3	ACos()	39
7.9.1.4	ACosh()	40
7.9.1.5	ArcsecToDeg()	40
7.9.1.6	ArcsecToRad()	40
7.9.1.7	ASin()	40
7.9.1.8	ASinh()	40
7.9.1.9	ATan()	40
7.9.1.10	ATan2()	41

7.9.1.11	ATanh()	41
7.9.1.12	Cbrt()	41
7.9.1.13	Ceiling()	41
7.9.1.14	Cos()	41
7.9.1.15	Cosh()	42
7.9.1.16	Deg()	42
7.9.1.17	DegToRad()	42
7.9.1.18	Exp()	42
7.9.1.19	Exp10()	42
7.9.1.20	Fix()	43
7.9.1.21	Floor()	43
7.9.1.22	IsEqual() [1/2]	43
7.9.1.23	IsEqual() [2/2]	43
7.9.1.24	IsEven()	43
7.9.1.25	IsInf()	43
7.9.1.26	IsNaN()	44
7.9.1.27	IsNegative()	44
7.9.1.28	IsNonNegative()	44
7.9.1.29	IsOdd()	44
7.9.1.30	IsPositive()	44
7.9.1.31	IsZero()	44
7.9.1.32	Ln()	45
7.9.1.33	Log() [1/3]	45
7.9.1.34	Log() [2/3]	45
7.9.1.35	Log() [3/3]	45
7.9.1.36	Log10()	46
7.9.1.37	Max()	46
7.9.1.38	Min()	46
7.9.1.39	Mod()	46
7.9.1.40	NearestInt()	46

7.9.1.41	Pow() [1/2]	46
7.9.1.42	Pow() [2/2]	47
7.9.1.43	Quotient() [1/2]	47
7.9.1.44	Quotient() [2/2]	47
7.9.1.45	Rad()	47
7.9.1.46	RadToDeg()	47
7.9.1.47	Rand()	48
7.9.1.48	Randn()	48
7.9.1.49	Rem()	48
7.9.1.50	Round()	48
7.9.1.51	SetSeed()	48
7.9.1.52	SignOf()	48
7.9.1.53	Sin()	49
7.9.1.54	Sinh()	49
7.9.1.55	SinXOverX()	49
7.9.1.56	Sqrt()	49
7.9.1.57	Tan()	49
7.9.1.58	Tanh()	49
7.10	GmatOrbitConstants Namespace Reference	50
7.11	GmatPhysicalConstants Namespace Reference	50
7.11.1	Variable Documentation	50
7.11.1.1	ABSOLUTE_ZERO_C	50
7.11.1.2	ABSOLUTE_ZERO_K	50
7.11.1.3	ASTRONOMICAL_UNIT	50
7.11.1.4	c	50
7.11.1.5	SPEED_OF_LIGHT_VACUUM	50
7.11.1.6	UNIVERSAL_GRAVITATIONAL_CONSTANT	51
7.12	GmatRealConstants Namespace Reference	51
7.12.1	Detailed Description	51
7.12.2	Variable Documentation	51

7.12.2.1	INTEGER_MAX	51
7.12.2.2	INTEGER_UNDEFINED	51
7.12.2.3	REAL_DIG	52
7.12.2.4	REAL_EPSILON	52
7.12.2.5	REAL_MAX	52
7.12.2.6	REAL_MAX_10_EXP	52
7.12.2.7	REAL_MAX_EXP	52
7.12.2.8	REAL_MIN	52
7.12.2.9	REAL_MIN_10_EXP	52
7.12.2.10	REAL_MIN_EXP	52
7.12.2.11	REAL_TOL	53
7.12.2.12	REAL_UNDEFINED	53
7.12.2.13	REAL_UNDEFINED_LARGE	53
7.12.2.14	SHORT_REAL_RADIX	53
7.13	GmatRealUtil Namespace Reference	53
7.13.1	Detailed Description	54
7.13.2	Function Documentation	54
7.13.2.1	CartesianToRaCodec()	54
7.13.2.2	CartesianToRaDec()	54
7.13.2.3	Max()	54
7.13.2.4	Min()	54
7.13.2.5	operator<<() [1/2]	55
7.13.2.6	operator<<() [2/2]	55
7.13.2.7	operator>>() [1/2]	55
7.13.2.8	operator>>() [2/2]	55
7.13.2.9	RaCodecToCartesian()	56
7.13.2.10	RaCodecToRaDec()	56
7.13.2.11	RaDecToCartesian()	56
7.13.2.12	RaDecToRaCodec()	56
7.13.2.13	RealToString()	56

7.13.2.14 ToString() [1/2]	56
7.13.2.15 ToString() [2/2]	57
7.14 GmatSolarSystemDefaults Namespace Reference	57
7.14.1 Detailed Description	59
7.14.2 Enumeration Type Documentation	59
7.14.2.1 DefaultMoons	59
7.14.2.2 DefaultPlanets	59
7.14.3 Variable Documentation	60
7.14.3.1 ADRASTEIA_NAME	60
7.14.3.2 AMALTHEA_NAME	60
7.14.3.3 ARIEL_NAME	60
7.14.3.4 ATLAS_NAME	60
7.14.3.5 BELINDA_NAME	60
7.14.3.6 BIANCA_NAME	61
7.14.3.7 CALLISTO_NAME	61
7.14.3.8 CALYPSO_NAME	61
7.14.3.9 CHARON_NAME	61
7.14.3.10 CORDELIA_NAME	61
7.14.3.11 CRESSIDA_NAME	61
7.14.3.12 DEIMOS_NAME	61
7.14.3.13 DESDEMONA_NAME	61
7.14.3.14 DESPINA_NAME	62
7.14.3.15 DIONE_NAME	62
7.14.3.16 EARTH_NAME	62
7.14.3.17 ENCELADUS_NAME	62
7.14.3.18 EPIMETHEUS_NAME	62
7.14.3.19 EUROPA_NAME	62
7.14.3.20 GALATEA_NAME	62
7.14.3.21 GANYMEDE_NAME	62
7.14.3.22 HELENE_NAME	63

7.14.3.23 IAPETUS_NAME	63
7.14.3.24 IO_NAME	63
7.14.3.25 JANUS_NAME	63
7.14.3.26 JULIET_NAME	63
7.14.3.27 JUPITER_NAME	63
7.14.3.28 LARISSA_NAME	63
7.14.3.29 MARS_NAME	63
7.14.3.30 MERCURY_NAME	64
7.14.3.31 METIS_NAME	64
7.14.3.32 MIMAS_NAME	64
7.14.3.33 MIRANDA_NAME	64
7.14.3.34 MOON_CENTRAL_BODIES	64
7.14.3.35 MOON_EQUATORIAL_RADIUS	64
7.14.3.36 MOON_FLATTENING	65
7.14.3.37 MOON_MU	65
7.14.3.38 MOON_NAIF_IDS	65
7.14.3.39 MOON_NAME	65
7.14.3.40 MOON_NAMES	66
7.14.3.41 MOON_ORIENTATION_PARAMETERS	66
7.14.3.42 MOON_SPICE_FRAME_ID	66
7.14.3.43 MOON_TWO_BODY_ELEMENTS	66
7.14.3.44 MOON_TWO_BODY_EPOCH	67
7.14.3.45 NAIAD_NAME	67
7.14.3.46 NEPTUNE_NAME	67
7.14.3.47 OBERON_NAME	67
7.14.3.48 OPHELIA_NAME	67
7.14.3.49 PAN_NAME	67
7.14.3.50 PANDORA_NAME	68
7.14.3.51 PHOBOS_NAME	68
7.14.3.52 PHOEBE_NAME	68

7.14.3.53 PLANET_EQUATORIAL_RADIUS	68
7.14.3.54 PLANET_FLATTENING	68
7.14.3.55 PLANET_MU	69
7.14.3.56 PLANET_NAIF_IDS	69
7.14.3.57 PLANET_NAMES	69
7.14.3.58 PLANET_ORIENTATION_PARAMETERS	70
7.14.3.59 PLANET_SPICE_FRAME_ID	70
7.14.3.60 PLANET_TWO_BODY_ELEMENTS	70
7.14.3.61 PLANET_TWO_BODY_EPOCH	71
7.14.3.62 PLUTO_NAME	71
7.14.3.63 PORTIA_NAME	71
7.14.3.64 PROMETHEUS_NAME	71
7.14.3.65 PROTEUS_NAME	71
7.14.3.66 PUCK_NAME	71
7.14.3.67 RHEA_NAME	72
7.14.3.68 ROSALIND_NAME	72
7.14.3.69 SATURN_NAME	72
7.14.3.70 SOLAR_SYSTEM_BARYCENTER_NAME	72
7.14.3.71 SSB_MU	72
7.14.3.72 SSB_NAIF_ID	72
7.14.3.73 STAR_EQUATORIAL_RADIUS	72
7.14.3.74 STAR_FLATTENING	73
7.14.3.75 STAR_MU	73
7.14.3.76 STAR_NAIF_IDS	73
7.14.3.77 STAR_ORIENTATION_PARAMETERS	73
7.14.3.78 STAR_PHOTOSPHERE_RADIUS	73
7.14.3.79 STAR_RADIANT_POWER	73
7.14.3.80 STAR_REFERENCE_DISTANCE	73
7.14.3.81 STAR_SPICE_FRAME_ID	74
7.14.3.82 STAR_TWO_BODY_ELEMENTS	74

7.14.3.83 STAR_TWO_BODY_EPOCH	74
7.14.3.84 SUN_NAME	74
7.14.3.85 TELESTO_NAME	74
7.14.3.86 TETHYS_NAME	74
7.14.3.87 THALASSA_NAME	74
7.14.3.88 THEBE_NAME	75
7.14.3.89 TITAN_NAME	75
7.14.3.90 TITANIA_NAME	75
7.14.3.91 TRITON_NAME	75
7.14.3.92 UMBRIEL_NAME	75
7.14.3.93 URANUS_NAME	75
7.14.3.94 VENUS_NAME	75
7.15 GmatStringUtil Namespace Reference	75
7.15.1 Detailed Description	79
7.15.2 Enumeration Type Documentation	79
7.15.2.1 AlignmentType	79
7.15.2.2 StripType	79
7.15.3 Function Documentation	79
7.15.3.1 AddEnclosingString()	79
7.15.3.2 AreAllBracketsBalanced()	80
7.15.3.3 AreAllNamesValid()	80
7.15.3.4 BuildNumber()	80
7.15.3.5 Capitalize()	81
7.15.3.6 DecomposeBy()	81
7.15.3.7 EndsWith()	81
7.15.3.8 EndsWithPathSeparator()	81
7.15.3.9 FindFirstAndLast()	81
7.15.3.10 FindLastParenMatch()	82
7.15.3.11 FindMatchingBracket()	82
7.15.3.12 FindMatchingParen()	82

7.15.3.13 FindParenMatch()	82
7.15.3.14 GetAlignmentString()	82
7.15.3.15 GetArrayCommaIndex()	83
7.15.3.16 GetArrayIndex() [1/2]	83
7.15.3.17 GetArrayIndex() [2/2]	83
7.15.3.18 GetArrayIndexVar()	83
7.15.3.19 GetArrayName()	83
7.15.3.20 GetClosingBracket()	84
7.15.3.21 GetInvalidNameMessageFormat()	84
7.15.3.22 GetVarNames()	84
7.15.3.23 HasMissingQuote()	84
7.15.3.24 HasNoBrackets()	84
7.15.3.25 IsBlank()	85
7.15.3.26 IsBracketBalanced()	85
7.15.3.27 IsBracketPartOfArray()	85
7.15.3.28 IsCommaPartOfArray()	85
7.15.3.29 IsEnclosedWith()	85
7.15.3.30 IsEnclosedWithBraces()	85
7.15.3.31 IsEnclosedWithBrackets()	86
7.15.3.32 IsEnclosedWithExtraParen()	86
7.15.3.33 IsLastNumberPartOfName()	86
7.15.3.34 IsMathEquation()	86
7.15.3.35 IsMathOperator()	86
7.15.3.36 IsNumber()	86
7.15.3.37 IsOneElementArray()	87
7.15.3.38 IsOuterParen()	87
7.15.3.39 IsParenBalanced()	87
7.15.3.40 IsParenEmpty()	87
7.15.3.41 IsParenPartOfArray()	87
7.15.3.42 IsSimpleArrayElement()	87

7.15.3.43 IsSingleItem()	87
7.15.3.44 IsStringInsideSymbols()	88
7.15.3.45 IsThereEqualSign()	88
7.15.3.46 IsThereMathSymbol()	88
7.15.3.47 IsValidExtendedIdentity()	88
7.15.3.48 IsValidFileName()	88
7.15.3.49 IsValidFullFileName()	89
7.15.3.50 IsValidFunctionCall()	89
7.15.3.51 IsValidIdentity()	89
7.15.3.52 IsValidName() [1/2]	89
7.15.3.53 IsValidName() [2/2]	89
7.15.3.54 IsValidNumber()	89
7.15.3.55 IsValidParameterName()	90
7.15.3.56 IsValidReal()	90
7.15.3.57 MakeCommentLines()	90
7.15.3.58 NumberOfOccurrences()	90
7.15.3.59 NumberOfScientificNotation()	90
7.15.3.60 PadWithBlanks()	90
7.15.3.61 ParseFunctionCall()	90
7.15.3.62 ParseFunctionName()	91
7.15.3.63 ParseName()	91
7.15.3.64 ParseParameter()	91
7.15.3.65 RealToString() [1/2]	92
7.15.3.66 RealToString() [2/2]	92
7.15.3.67 RemoveAll() [1/2]	92
7.15.3.68 RemoveAll() [2/2]	92
7.15.3.69 RemoveAllBlanks()	92
7.15.3.70 RemoveEnclosingString()	93
7.15.3.71 RemoveExtraParen()	93
7.15.3.72 RemoveInlineComment()	93

7.15.3.73 RemoveLastNumber()	93
7.15.3.74 RemoveLastString()	93
7.15.3.75 RemoveMathSymbols()	93
7.15.3.76 RemoveMultipleSpaces()	94
7.15.3.77 RemoveOuterParen()	94
7.15.3.78 RemoveOuterString()	94
7.15.3.79 RemoveScientificNotation()	94
7.15.3.80 RemoveSpaceInBrackets()	94
7.15.3.81 RemoveTrailingZeros()	94
7.15.3.82 Replace()	95
7.15.3.83 ReplaceChainedUnaryOperators()	95
7.15.3.84 ReplaceFirst()	95
7.15.3.85 ReplaceName()	95
7.15.3.86 ReplaceNumber()	95
7.15.3.87 SeparateBrackets()	96
7.15.3.88 SeparateBy()	96
7.15.3.89 SeparateByComma()	96
7.15.3.90 SeparateDots()	96
7.15.3.91 StartsWith()	96
7.15.3.92 StringToWideString()	96
7.15.3.93 Strip()	97
7.15.3.94 ToBoolean() [1/2]	97
7.15.3.95 ToBoolean() [2/2]	97
7.15.3.96 ToBooleanArray()	97
7.15.3.97 ToInteger() [1/3]	97
7.15.3.98 ToInteger() [2/3]	97
7.15.3.99 ToInteger() [3/3]	98
7.15.3.100ToIntegerArray() [1/2]	98
7.15.3.101ToIntegerArray() [2/2]	98
7.15.3.102ToLower()	98

7.15.3.103ToOnOff()	98
7.15.3.104ToOrdinal()	98
7.15.3.105ToReal() [1/4]	99
7.15.3.106ToReal() [2/4]	99
7.15.3.107ToReal() [3/4]	99
7.15.3.108ToReal() [4/4]	99
7.15.3.109ToRealArray()	99
7.15.3.110ToString() [1/5]	100
7.15.3.111ToString() [2/5]	100
7.15.3.112ToString() [3/5]	100
7.15.3.113ToString() [4/5]	100
7.15.3.114ToString() [5/5]	100
7.15.3.115ToStringArray()	100
7.15.3.116ToStringNoZeros()	101
7.15.3.117ToUnsignedInt() [1/2]	101
7.15.3.118ToUnsignedInt() [2/2]	101
7.15.3.119ToUnsignedIntArray()	101
7.15.3.120ToUpper()	101
7.15.3.121Trim()	101
7.15.3.122WideStringToString() [1/2]	102
7.15.3.123WideStringToString() [2/2]	102
7.15.3.124WriteStringArray()	102
7.16 GmatTimeConstants Namespace Reference	102
7.16.1 Enumeration Type Documentation	103
7.16.1.1 DayName	103
7.16.1.2 MonthName	103
7.16.2 Variable Documentation	103
7.16.2.1 A1_TAI_OFFSET	103
7.16.2.2 A1MJD_OF_J2000	104
7.16.2.3 DAYS_BEFORE_MONTH	104

7.16.2.4	DAYS_IN_MONTH	104
7.16.2.5	DAYS_PER_JULIAN_CENTURY	104
7.16.2.6	DAYS_PER_SEC	104
7.16.2.7	DAYS_PER_YEAR	104
7.16.2.8	JD_JAN_5_1941	105
7.16.2.9	JD_MJD_OFFSET	105
7.16.2.10	JD_NOV_17_1858	105
7.16.2.11	JD_OF_J2000	105
7.16.2.12	JULIAN_DATE_OF_010541	105
7.16.2.13	LEAP_YEAR_DAYS_BEFORE_MONTH	105
7.16.2.14	LEAP_YEAR_DAYS_IN_MONTH	106
7.16.2.15	MJD_EPOCH_PRECISION	106
7.16.2.16	MJD_OF_J2000	106
7.16.2.17	SECS_PER_DAY	106
7.16.2.18	SECS_PER_HOUR	106
7.16.2.19	SECS_PER_MINUTE	106
7.16.2.20	TIME_OF_J2000	106
7.16.2.21	TT_TAI_OFFSET	107
7.17	GmatTimeUtil Namespace Reference	107
7.17.1	Function Documentation	107
7.17.1.1	FormatCurrentTime()	107
7.17.1.2	GetGregorianFormat()	107
7.17.1.3	GetMonth()	107
7.17.1.4	GetMonthName()	107
7.17.1.5	IsValidMonthName()	107

8 Class Documentation	109
8.1 A1Date Class Reference	109
8.1.1 Constructor & Destructor Documentation	110
8.1.1.1 A1Date() [1/7]	110
8.1.1.2 A1Date() [2/7]	110
8.1.1.3 A1Date() [3/7]	111
8.1.1.4 A1Date() [4/7]	111
8.1.1.5 A1Date() [5/7]	111
8.1.1.6 A1Date() [6/7]	111
8.1.1.7 A1Date() [7/7]	112
8.1.1.8 ~A1Date()	112
8.1.2 Member Function Documentation	112
8.1.2.1 operator<()	112
8.1.2.2 operator=()	112
8.1.2.3 operator>()	113
8.2 A1Mjd Class Reference	113
8.2.1 Detailed Description	114
8.2.2 Constructor & Destructor Documentation	114
8.2.2.1 A1Mjd() [1/3]	114
8.2.2.2 A1Mjd() [2/3]	114
8.2.2.3 A1Mjd() [3/3]	115
8.2.2.4 ~A1Mjd()	115
8.2.3 Member Function Documentation	115
8.2.3.1 Clone()	115
8.2.3.2 Get()	115
8.2.3.3 GetDataDescriptions()	115
8.2.3.4 GetNumData()	116
8.2.3.5 GetReal()	116
8.2.3.6 operator"!=()	116
8.2.3.7 operator+() [1/2]	116

8.2.3.8	operator+() [2/2]	117
8.2.3.9	operator+=() [1/2]	117
8.2.3.10	operator+=() [2/2]	117
8.2.3.11	operator-() [1/2]	117
8.2.3.12	operator-() [2/2]	117
8.2.3.13	operator-=() [1/2]	117
8.2.3.14	operator-=() [2/2]	118
8.2.3.15	operator<()	118
8.2.3.16	operator<=()	118
8.2.3.17	operator=()	118
8.2.3.18	operator==()	118
8.2.3.19	operator>()	119
8.2.3.20	operator>=()	119
8.2.3.21	Set()	119
8.2.3.22	SetReal()	120
8.2.3.23	Subtract() [1/2]	120
8.2.3.24	Subtract() [2/2]	120
8.2.3.25	ToA1Date()	120
8.2.3.26	ToUtcDate()	120
8.2.3.27	ToUtcMjd()	121
8.2.3.28	ToValueStrings()	121
8.2.3.29	UtcMjdToA1Mjd()	121
8.2.4	Member Data Documentation	121
8.2.4.1	J2000	121
8.3	RealUtilitiesExceptions::ArgumentError Class Reference	122
8.3.1	Constructor & Destructor Documentation	122
8.3.1.1	ArgumentError()	123
8.4	ArrayTemplateExceptions::ArrayAlreadySized Class Reference	123
8.4.1	Constructor & Destructor Documentation	124
8.4.1.1	ArrayAlreadySized()	124

8.5	ArrayTemplate< T > Class Template Reference	124
8.5.1	Constructor & Destructor Documentation	125
8.5.1.1	ArrayTemplate() [1/4]	125
8.5.1.2	ArrayTemplate() [2/4]	125
8.5.1.3	ArrayTemplate() [3/4]	126
8.5.1.4	ArrayTemplate() [4/4]	126
8.5.1.5	~ArrayTemplate()	126
8.5.2	Member Function Documentation	126
8.5.2.1	GetDataVector()	126
8.5.2.2	GetElement()	126
8.5.2.3	GetSize()	126
8.5.2.4	init()	127
8.5.2.5	IsSized()	127
8.5.2.6	operator"!="()	127
8.5.2.7	operator>() [1/2]	127
8.5.2.8	operator>() [2/2]	127
8.5.2.9	operator=()	127
8.5.2.10	operator==()	128
8.5.2.11	operator[]() [1/2]	128
8.5.2.12	operator[]() [2/2]	128
8.5.2.13	Resize()	128
8.5.2.14	SetElement()	128
8.5.2.15	SetSize()	128
8.5.3	Member Data Documentation	129
8.5.3.1	elementD	129
8.5.3.2	isSizedD	129
8.5.3.3	sizeD	129
8.6	ArrayTemplateExceptions Class Reference	129
8.6.1	Detailed Description	129
8.7	AttitudeConversionUtility Class Reference	130

8.7.1	Detailed Description	130
8.7.2	Member Function Documentation	130
8.7.2.1	DCMToEulerAxisAndAngle()	130
8.7.2.2	EulerAxisAndAngleToDCM()	131
8.7.2.3	IsValidEulerSequence()	131
8.7.2.4	ToAngularVelocity()	131
8.7.2.5	ToCosineMatrix() [1/3]	133
8.7.2.6	ToCosineMatrix() [2/3]	133
8.7.2.7	ToCosineMatrix() [3/3]	134
8.7.2.8	ToEulerAngleRates()	134
8.7.2.9	ToEulerAngles() [1/2]	135
8.7.2.10	ToEulerAngles() [2/2]	135
8.7.2.11	ToMRPs()	136
8.7.2.12	ToQuaternion() [1/3]	136
8.7.2.13	ToQuaternion() [2/3]	137
8.7.2.14	ToQuaternion() [3/3]	137
8.8	BaseException Class Reference	137
8.8.1	Detailed Description	139
8.8.2	Constructor & Destructor Documentation	139
8.8.2.1	BaseException() [1/2]	139
8.8.2.2	BaseException() [2/2]	139
8.8.2.3	~BaseException()	139
8.8.3	Member Function Documentation	140
8.8.3.1	GetDetails()	140
8.8.3.2	GetFullMessage()	140
8.8.3.3	GetMessageType()	140
8.8.3.4	IsFatal()	140
8.8.3.5	operator=() [1/2]	140
8.8.3.6	operator=() [2/2]	140
8.8.3.7	SetDetails() [1/2]	140

8.8.3.8	SetDetails() [2/2]	141
8.8.3.9	SetFatal()	141
8.8.3.10	SetMessage()	141
8.8.3.11	SetMessageType()	141
8.8.4	Member Data Documentation	141
8.8.4.1	MAX_MESSAGE_LENGTH	141
8.9	GmatTimeUtil::CalDate Class Reference	141
8.9.1	Constructor & Destructor Documentation	142
8.9.1.1	CalDate() [1/2]	142
8.9.1.2	CalDate() [2/2]	142
8.9.2	Member Data Documentation	142
8.9.2.1	day	142
8.9.2.2	hour	143
8.9.2.3	minute	143
8.9.2.4	month	143
8.9.2.5	second	143
8.9.2.6	year	143
8.10	CholeskyFactorization Class Reference	144
8.10.1	Detailed Description	144
8.10.2	Constructor & Destructor Documentation	145
8.10.2.1	CholeskyFactorization() [1/2]	145
8.10.2.2	CholeskyFactorization() [2/2]	145
8.10.2.3	~CholeskyFactorization()	145
8.10.3	Member Function Documentation	145
8.10.3.1	Factor()	145
8.10.3.2	Invert() [1/2]	146
8.10.3.3	Invert() [2/2]	146
8.10.3.4	operator=()	146
8.11	ConsoleAppException Class Reference	147
8.11.1	Detailed Description	147

8.11.2	Constructor & Destructor Documentation	148
8.11.2.1	ConsoleAppException()	148
8.11.2.2	~ConsoleAppException()	148
8.12	ConsoleMessageReceiver Class Reference	148
8.12.1	Detailed Description	149
8.12.2	Member Function Documentation	150
8.12.2.1	ClearMessage()	150
8.12.2.2	ClearMessageQueue()	150
8.12.2.3	CloseLogFile()	150
8.12.2.4	GetLogEnable()	150
8.12.2.5	GetLogFileName()	150
8.12.2.6	GetMessage()	151
8.12.2.7	Instance()	151
8.12.2.8	LogMessage() [1/2]	151
8.12.2.9	LogMessage() [2/2]	151
8.12.2.10	OpenLogFile()	152
8.12.2.11	PopupMessage() [1/2]	152
8.12.2.12	PopupMessage() [2/2]	152
8.12.2.13	PutMessage()	153
8.12.2.14	SetLogEnable()	153
8.12.2.15	SetLogFile()	153
8.12.2.16	SetLogPath()	154
8.12.2.17	ShowMessage() [1/2]	154
8.12.2.18	ShowMessage() [2/2]	154
8.13	Date Class Reference	155
8.13.1	Detailed Description	156
8.13.2	Constructor & Destructor Documentation	156
8.13.2.1	Date() [1/7]	156
8.13.2.2	Date() [2/7]	157
8.13.2.3	Date() [3/7]	157

8.13.2.4	Date() [4/7]	157
8.13.2.5	Date() [5/7]	157
8.13.2.6	Date() [6/7]	157
8.13.2.7	Date() [7/7]	158
8.13.2.8	~Date()	158
8.13.3	Member Function Documentation	158
8.13.3.1	GetDataDescriptions()	158
8.13.3.2	GetDay()	158
8.13.3.3	GetDayName()	158
8.13.3.4	GetDaysPerMonth()	158
8.13.3.5	GetHour()	158
8.13.3.6	GetMinute()	159
8.13.3.7	GetMonth()	159
8.13.3.8	GetMonthName()	159
8.13.3.9	GetNumData()	159
8.13.3.10	GetSecond()	159
8.13.3.11	GetSecondsOfDay()	159
8.13.3.12	GetYear()	159
8.13.3.13	IsValid()	159
8.13.3.14	operator<()	160
8.13.3.15	operator>()	160
8.13.3.16	ToDayOfYear()	160
8.13.3.17	ToPackedCalendarReal()	160
8.13.3.18	ToPackedCalendarString()	160
8.13.3.19	ToPackedHHMMSS()	160
8.13.3.20	ToPackedYYYYMMDD()	160
8.13.3.21	ToValueStrings()	161
8.13.3.22	ToYearDOYHourMinSec()	161
8.13.3.23	ToYearMonthDayHourMinSec() [1/3]	161
8.13.3.24	ToYearMonthDayHourMinSec() [2/3]	161

8.13.3.25 ToYearMonthDayHourMinSec() [3/3]	161
8.13.4 Member Data Documentation	162
8.13.4.1 DATA_DESCRIPTIONS	162
8.13.4.2 dayD	162
8.13.4.3 monthD	162
8.13.4.4 mPackedString	162
8.13.4.5 NUM_DATA	162
8.13.4.6 secondsOfDayD	162
8.13.4.7 stringValues	163
8.13.4.8 yearD	163
8.14 DateUtil Class Reference	163
8.14.1 Detailed Description	164
8.14.2 Member Function Documentation	164
8.14.2.1 FormatGregorian()	164
8.14.2.2 IsValidGregorian()	164
8.14.2.3 JulianDay()	165
8.14.3 Friends And Related Function Documentation	165
8.14.3.1 IsLeapYear	165
8.14.3.2 IsValidTime	165
8.14.3.3 JulianDate	165
8.14.3.4 ModifiedJulianDate	166
8.14.3.5 ModifiedJulianDateGT	167
8.14.3.6 ToDOYFromYearMonthDay	167
8.14.3.7 ToHMSFromSecondsOfDay	167
8.14.3.8 ToMonthDayFromYearDOY	168
8.14.3.9 ToSecondsOfDayFromHMS	168
8.14.3.10 UnpackDate	168
8.14.3.11 UnpackDateWithDOY	169
8.14.3.12 UnpackTime	169
8.14.4 Member Data Documentation	169

8.14.4.1	EARLIEST_VALID_GREGORIAN	169
8.14.4.2	EARLIEST_VALID_MJD	169
8.14.4.3	EARLIEST_VALID_MJD_VALUE	170
8.14.4.4	LATEST_VALID_GREGORIAN	170
8.14.4.5	LATEST_VALID_MJD	170
8.14.4.6	LATEST_VALID_MJD_VALUE	170
8.14.4.7	MAX_DAY	170
8.14.4.8	MAX_HOUR	170
8.14.4.9	MAX_MINUTE	170
8.14.4.10	MAX_MONTH	170
8.14.4.11	MAX_SEC	171
8.14.4.12	MAX_YEAR	171
8.14.4.13	MIN_DAY	171
8.14.4.14	MIN_HOUR	171
8.14.4.15	MIN_MINUTE	171
8.14.4.16	MIN_MONTH	171
8.14.4.17	MIN_SEC	171
8.14.4.18	MIN_YEAR	172
8.15	TableTemplateExceptions::DimensionError Class Reference	172
8.15.1	Constructor & Destructor Documentation	173
8.15.1.1	DimensionError()	173
8.16	ArrayTemplateExceptions::DimensionError Class Reference	173
8.16.1	Constructor & Destructor Documentation	174
8.16.1.1	DimensionError()	174
8.17	Rmatrix::DivideByZero Class Reference	174
8.17.1	Constructor & Destructor Documentation	175
8.17.1.1	DivideByZero()	175
8.18	GmatTimeUtil::ElapsedDate Class Reference	175
8.18.1	Constructor & Destructor Documentation	175
8.18.1.1	ElapsedDate() <small>[1/2]</small>	176

8.18.1.2 ElapsedDate() [2/2]	176
8.18.2 Member Data Documentation	176
8.18.2.1 days	176
8.18.2.2 hours	176
8.18.2.3 minutes	176
8.18.2.4 seconds	176
8.19 ElapsedTime Class Reference	177
8.19.1 Detailed Description	177
8.19.2 Constructor & Destructor Documentation	177
8.19.2.1 ElapsedTime() [1/2]	177
8.19.2.2 ElapsedTime() [2/2]	177
8.19.2.3 ~ElapsedTime()	178
8.19.3 Member Function Documentation	178
8.19.3.1 Get()	178
8.19.3.2 GetDataDescriptions()	178
8.19.3.3 GetNumData()	178
8.19.3.4 operator"!=()	178
8.19.3.5 operator+()	178
8.19.3.6 operator+=()	178
8.19.3.7 operator-()	179
8.19.3.8 operator-=()	179
8.19.3.9 operator<()	179
8.19.3.10 operator<=()	179
8.19.3.11 operator=()	179
8.19.3.12 operator==()	179
8.19.3.13 operator>()	179
8.19.3.14 operator>=()	180
8.19.3.15 Set()	180
8.19.3.16 ToElapsedTime()	180
8.19.3.17 ToValueStrings()	180

8.20 Element Struct Reference	180
8.20.1 Member Data Documentation	180
8.20.1.1 index	181
8.20.1.2 value	181
8.21 EopFile Class Reference	181
8.21.1 Constructor & Destructor Documentation	182
8.21.1.1 EopFile() [1/2]	182
8.21.1.2 EopFile() [2/2]	182
8.21.1.3 ~EopFile()	183
8.21.2 Member Function Documentation	183
8.21.2.1 GetFileName()	183
8.21.2.2 GetPolarMotionAndLod()	183
8.21.2.3 GetPolarMotionData()	184
8.21.2.4 GetTimeRange()	184
8.21.2.5 GetUt1UtcOffset()	184
8.21.2.6 Initialize()	184
8.21.2.7 IsBlank()	185
8.21.2.8 operator=()	185
8.21.2.9 ResetEopFile()	185
8.21.3 Member Data Documentation	185
8.21.3.1 eopFileName	185
8.21.3.2 eopFType	186
8.21.3.3 isInitialized	186
8.21.3.4 lastIndex	186
8.21.3.5 lastOffset	186
8.21.3.6 lastTaiMjd	186
8.21.3.7 lastUtcJd	186
8.21.3.8 MAX_TABLE_SIZE	186
8.21.3.9 polarMotion	187
8.21.3.10 previousIndex	187

8.21.3.11 tableSz	187
8.21.3.12 taiTime	187
8.21.3.13 theTimeConverter	187
8.21.3.14 ut1UtcOffsets	187
8.22 ExponentialAtmosphere Class Reference	188
8.22.1 Detailed Description	188
8.22.2 Constructor & Destructor Documentation	189
8.22.2.1 ExponentialAtmosphere() [1/2]	189
8.22.2.2 ~ExponentialAtmosphere()	189
8.22.2.3 ExponentialAtmosphere() [2/2]	189
8.22.3 Member Function Documentation	189
8.22.3.1 Clone()	189
8.22.3.2 Density()	190
8.22.3.3 FindBand()	190
8.22.3.4 GetScaleHeight()	190
8.22.3.5 operator=()	191
8.22.3.6 SetConstants()	191
8.22.3.7 Smooth()	191
8.22.4 Member Data Documentation	192
8.22.4.1 altitudeBands	192
8.22.4.2 refDensity	192
8.22.4.3 refHeight	192
8.22.4.4 scaleHeight	192
8.22.4.5 smoothDensity	193
8.23 FileManager Class Reference	193
8.23.1 Detailed Description	195
8.23.2 Member Enumeration Documentation	195
8.23.2.1 FileType	195
8.23.3 Constructor & Destructor Documentation	197
8.23.3.1 ~FileManager()	197

8.23.4	Member Function Documentation	197
8.23.4.1	AddGmatFunctionPath() [1/2]	197
8.23.4.2	AddGmatFunctionPath() [2/2]	197
8.23.4.3	AddGmatIncludePath() [1/2]	197
8.23.4.4	AddGmatIncludePath() [2/2]	197
8.23.4.5	AddMatlabFunctionPath() [1/2]	198
8.23.4.6	AddMatlabFunctionPath() [2/2]	198
8.23.4.7	AddPythonModulePath()	198
8.23.4.8	AdjustSettings()	198
8.23.4.9	ClearGmatFunctionPath()	198
8.23.4.10	ClearGmatIncludePath()	199
8.23.4.11	ClearMatlabFunctionPath()	199
8.23.4.12	ConvertToAbsPath()	199
8.23.4.13	CopyFile()	199
8.23.4.14	DoesDirectoryExist()	199
8.23.4.15	DoesFileExist()	199
8.23.4.16	FindMainIconFile()	200
8.23.4.17	FindPath() [1/2]	200
8.23.4.18	FindPath() [2/2]	200
8.23.4.19	GetAbsPathname() [1/2]	201
8.23.4.20	GetAbsPathname() [2/2]	201
8.23.4.21	GetAllGmatFunctionPaths()	202
8.23.4.22	GetAllGmatIncludePaths()	202
8.23.4.23	GetAllMatlabFunctionPaths()	202
8.23.4.24	GetAllPythonModulePaths()	202
8.23.4.25	GetBinDirectory()	202
8.23.4.26	GetBody3dModelFile()	202
8.23.4.27	GetCurrentWorkingDirectory()	203
8.23.4.28	GetFilename() [1/2]	203
8.23.4.29	GetFilename() [2/2]	203

8.23.4.30 GetFullPathname() [1/2]	204
8.23.4.31 GetFullPathname() [2/2]	204
8.23.4.32 GetFullStartupFilePath()	205
8.23.4.33 GetGmatFunctionPath() [1/2]	205
8.23.4.34 GetGmatFunctionPath() [2/2]	205
8.23.4.35 GetGmatIncludePath() [1/2]	205
8.23.4.36 GetGmatIncludePath() [2/2]	205
8.23.4.37 GetGmatWorkingDirectory()	205
8.23.4.38 GetLastFilePathMessage()	206
8.23.4.39 GetMatlabFunctionPath() [1/2]	206
8.23.4.40 GetMatlabFunctionPath() [2/2]	206
8.23.4.41 GetPathname() [1/2]	206
8.23.4.42 GetPathname() [2/2]	206
8.23.4.43 GetPathSeparator()	207
8.23.4.44 GetPluginList()	207
8.23.4.45 GetRootPath()	207
8.23.4.46 GetStartupFileDir()	208
8.23.4.47 GetStartupFileName()	208
8.23.4.48 GetTextureMapFile()	208
8.23.4.49 Instance()	208
8.23.4.50 ReadStartupFile()	208
8.23.4.51 RenameFile()	209
8.23.4.52 SetAbsPathname() [1/4]	209
8.23.4.53 SetAbsPathname() [2/4]	209
8.23.4.54 SetAbsPathname() [3/4]	210
8.23.4.55 SetAbsPathname() [4/4]	210
8.23.4.56 SetBinDirectory()	210
8.23.4.57 SetCurrentWorkingDirectory()	211
8.23.4.58 SetGmatWorkingDirectory()	211
8.23.4.59 ValidatePaths()	211

8.23.4.60 WriteStartupFile()	211
8.24 geoparms Struct Reference	211
8.24.1 Member Data Documentation	212
8.24.1.1 tkp	212
8.24.1.2 xtemp	212
8.25 GmatGlobal Class Reference	212
8.25.1 Member Enumeration Documentation	215
8.25.1.1 GuiMode	215
8.25.1.2 LogfileSource	215
8.25.1.3 MatlabMode	215
8.25.1.4 PlotMode	215
8.25.1.5 RunMode	216
8.25.2 Member Function Documentation	216
8.25.2.1 AddHiddenCommand()	216
8.25.2.2 ClearHiddenCommands()	216
8.25.2.3 EchoCommands()	216
8.25.2.4 GetActualFormat()	217
8.25.2.5 GetDataPrecision()	217
8.25.2.6 GetDataWidth()	217
8.25.2.7 GetDetailedRunState()	217
8.25.2.8 GetEopFile()	217
8.25.2.9 GetGmatVersion()	217
8.25.2.10 GetGuiMode()	217
8.25.2.11 GetHiddenCommands()	218
8.25.2.12 GetIncludeFoundInScriptResource()	218
8.25.2.13 GetIntegerWidth()	218
8.25.2.14 GetItrfCoefficientsFile()	218
8.25.2.15 GetLogfileName()	218
8.25.2.16 GetLogfileSource()	218
8.25.2.17 GetMatlabFuncNameExt()	219

8.25.2.18 GetMatlabMode()	219
8.25.2.19 GetOutputPath()	219
8.25.2.20 GetPlotMode()	219
8.25.2.21 GetRunInterrupted()	219
8.25.2.22 GetRunMode()	219
8.25.2.23 GetRunModeStartUp()	219
8.25.2.24 GetRunState()	220
8.25.2.25 GetSpacing()	220
8.25.2.26 GetTimePrecision()	220
8.25.2.27 GetTimeWidth()	220
8.25.2.28 GetWritePersonalizationFile()	220
8.25.2.29 Instance()	220
8.25.2.30 IsBatchMode()	220
8.25.2.31 IsBinaryIn()	221
8.25.2.32 IsBinaryOut()	221
8.25.2.33 IsEventLocationAvailable()	221
8.25.2.34 IsGmatCompiledIn64Bit()	221
8.25.2.35 IsUISavable()	221
8.25.2.36 IsHiddenCommand() [1/2]	221
8.25.2.37 IsHiddenCommand() [2/2]	221
8.25.2.38 IsHorizontal()	222
8.25.2.39 IsMatlabAvailable()	222
8.25.2.40 IsMatlabDebugOn()	222
8.25.2.41 IsMissionTreeDebugOn()	222
8.25.2.42 IsNitsClient()	222
8.25.2.43 IsScientific()	222
8.25.2.44 IsWritingFilePathInfo()	222
8.25.2.45 IsWritingGmatKeyword()	222
8.25.2.46 IsWritingParameterInfo()	223
8.25.2.47 RemoveHiddenCommand()	223

8.25.2.48 SetActualFormat()	223
8.25.2.49 SetAppendEol()	223
8.25.2.50 SetBatchMode()	223
8.25.2.51 SetBinaryIn()	223
8.25.2.52 SetBinaryOut()	224
8.25.2.53 SetCommandEchoMode()	224
8.25.2.54 SetCurrentFormat()	224
8.25.2.55 SetDataPrecision()	224
8.25.2.56 SetDataWidth()	224
8.25.2.57 SetDefaultFormat()	225
8.25.2.58 SetDetailedRunState()	225
8.25.2.59 SetEopFile()	225
8.25.2.60 SetEventLocationAvailable()	225
8.25.2.61 SetGuiMode()	225
8.25.2.62 SetHorizontal()	225
8.25.2.63 SetIncludeFoundInScriptResource()	226
8.25.2.64 SetIntegerWidth()	226
8.25.2.65 SetItrfCoefficientsFile()	226
8.25.2.66 SetLogfileName()	226
8.25.2.67 SetLogfileSource()	226
8.25.2.68 SetMatlabAvailable()	226
8.25.2.69 SetMatlabDebug()	227
8.25.2.70 SetMatlabFuncNameExt()	227
8.25.2.71 SetMatlabMode()	227
8.25.2.72 SetMissionTreeDebug()	227
8.25.2.73 SetNitsClient()	227
8.25.2.74 SetOutputPath()	227
8.25.2.75 SetPlotMode()	227
8.25.2.76 SetPrefix()	228
8.25.2.77 SetRunInterrupted()	228

8.25.2.78 SetRunMode()	228
8.25.2.79 SetRunState()	228
8.25.2.80 SetScientific()	228
8.25.2.81 SetShowPoint()	228
8.25.2.82 SetSkipSplashMode()	228
8.25.2.83 SetSpacing()	229
8.25.2.84 SetTimePrecision()	229
8.25.2.85 SetTimeWidth()	229
8.25.2.86 SetToCurrentFormat()	229
8.25.2.87 SetToDefaultFormat()	229
8.25.2.88 SetWriteFilePathInfo()	229
8.25.2.89 SetWriteGmatKeyword()	230
8.25.2.90 SetWriteParameterInfo()	230
8.25.2.91 SetWritePersonalizationFile()	230
8.25.2.92 ShowPoint()	230
8.25.2.93 SkipSplashMode()	230
8.25.3 Member Data Documentation	230
8.25.3.1 DATA_PRECISION	230
8.25.3.2 DATA_WIDTH	231
8.25.3.3 INTEGER_WIDTH	231
8.25.3.4 TIME_PRECISION	231
8.25.3.5 TIME_WIDTH	231
8.26 GmatTime Class Reference	231
8.26.1 Detailed Description	232
8.26.2 Constructor & Destructor Documentation	232
8.26.2.1 GmatTime() [1/3]	232
8.26.2.2 ~GmatTime()	233
8.26.2.3 GmatTime() [2/3]	233
8.26.2.4 GmatTime() [3/3]	233
8.26.3 Member Function Documentation	233

8.26.3.1	AddSeconds()	233
8.26.3.2	Clone()	233
8.26.3.3	GetDays()	233
8.26.3.4	GetFracSec()	233
8.26.3.5	GetMjd()	234
8.26.3.6	GetSec()	234
8.26.3.7	GetTimeInSec()	234
8.26.3.8	IsNearlyEqual()	234
8.26.3.9	operator"!=" [1/2]	234
8.26.3.10	operator"!=" [2/2]	234
8.26.3.11	operator*()	234
8.26.3.12	operator+() [1/2]	235
8.26.3.13	operator+() [2/2]	235
8.26.3.14	operator+=() [1/2]	235
8.26.3.15	operator+=() [2/2]	235
8.26.3.16	operator-() [1/2]	235
8.26.3.17	operator-() [2/2]	235
8.26.3.18	operator-=() [1/2]	235
8.26.3.19	operator-=() [2/2]	236
8.26.3.20	operator/()	236
8.26.3.21	operator<() [1/2]	236
8.26.3.22	operator<() [2/2]	236
8.26.3.23	operator<=() [1/2]	236
8.26.3.24	operator<=() [2/2]	236
8.26.3.25	operator=() [1/2]	236
8.26.3.26	operator=() [2/2]	237
8.26.3.27	operator==() [1/2]	237
8.26.3.28	operator==() [2/2]	237
8.26.3.29	operator>() [1/2]	237
8.26.3.30	operator>() [2/2]	237

8.26.3.31 operator>=()	[1 / 2]	237
8.26.3.32 operator>=()	[2 / 2]	237
8.26.3.33 SetDays()		238
8.26.3.34 SetFracSec()		238
8.26.3.35 SetMjdString()		238
8.26.3.36 SetSec()		238
8.26.3.37 SetTimeInSec()		238
8.26.3.38 SubtractSeconds()		238
8.26.3.39 ToString()		238
8.26.4 Member Data Documentation		239
8.26.4.1 Days		239
8.26.4.2 FracSec		239
8.26.4.3 Sec		239
8.27 GravityFileException Class Reference		239
8.27.1 Constructor & Destructor Documentation		240
8.27.1.1 GravityFileException()		240
8.28 GregorianCalendar Class Reference		240
8.28.1 Detailed Description		241
8.28.2 Constructor & Destructor Documentation		241
8.28.2.1 GregorianCalendar()	[1 / 3]	241
8.28.2.2 GregorianCalendar()	[2 / 3]	241
8.28.2.3 GregorianCalendar()	[3 / 3]	242
8.28.2.4 ~GregorianCalendar()		242
8.28.3 Member Function Documentation		242
8.28.3.1 GetDate()		242
8.28.3.2 GetType()		242
8.28.3.3 GetYMDHMS()		242
8.28.3.4 IsValid()	[1 / 2]	243
8.28.3.5 IsValid()	[2 / 2]	243
8.28.3.6 SetDate()	[1 / 2]	243

8.28.3.7	SetDate() [2/2]	243
8.28.3.8	SetType()	244
8.29	GregorianDate::GregorianDateException Class Reference	244
8.29.1	Constructor & Destructor Documentation	245
8.29.1.1	GregorianDateException()	245
8.30	TableTemplateExceptions::IllegalSize Class Reference	245
8.30.1	Constructor & Destructor Documentation	246
8.30.1.1	IllegalSize()	246
8.31	ArrayTemplateExceptions::IllegalSize Class Reference	247
8.31.1	Constructor & Destructor Documentation	247
8.31.1.1	IllegalSize()	248
8.32	RealUtilitiesExceptions::IllegalTime Class Reference	248
8.32.1	Constructor & Destructor Documentation	249
8.32.1.1	IllegalTime()	249
8.33	Interpolator Class Reference	249
8.33.1	Detailed Description	250
8.33.2	Constructor & Destructor Documentation	250
8.33.2.1	Interpolator() [1/2]	250
8.33.2.2	~Interpolator()	251
8.33.2.3	Interpolator() [2/2]	251
8.33.3	Member Function Documentation	251
8.33.3.1	AddPoint()	251
8.33.3.2	AllocateArrays()	252
8.33.3.3	CleanupArrays()	252
8.33.3.4	Clear()	252
8.33.3.5	Clone()	252
8.33.3.6	CopyArrays()	252
8.33.3.7	GetBufferSize()	253
8.33.3.8	GetForceInterpolation()	253
8.33.3.9	GetName()	253

8.33.3.10	GetPointCount()	253
8.33.3.11	GetRange()	253
8.33.3.12	Interpolate()	254
8.33.3.13	IsInterpolationFeasible()	254
8.33.3.14	operator=()	255
8.33.3.15	SetExtrapolation()	255
8.33.3.16	SetForceInterpolation()	255
8.33.3.17	SetRange()	255
8.33.4	Member Data Documentation	255
8.33.4.1	allowExtrapolation	256
8.33.4.2	bufferSize	256
8.33.4.3	dataIncreases	256
8.33.4.4	dependent	256
8.33.4.5	dimension	256
8.33.4.6	forceInterpolation	256
8.33.4.7	independent	257
8.33.4.8	instanceName	257
8.33.4.9	latestPoint	257
8.33.4.10	pointCount	257
8.33.4.11	previousX	257
8.33.4.12	range	257
8.33.4.13	rangeCalculated	258
8.33.4.14	requiredPoints	258
8.34	InterpolatorException Class Reference	258
8.34.1	Detailed Description	259
8.34.2	Constructor & Destructor Documentation	259
8.34.2.1	InterpolatorException()	259
8.34.2.2	~InterpolatorException()	259
8.35	InvalidStateRepresentationException Class Reference	259
8.35.1	Detailed Description	260

8.35.2	Constructor & Destructor Documentation	260
8.35.2.1	InvalidStateRepresentationException()	260
8.36	InvalidTimeException Class Reference	261
8.36.1	Constructor & Destructor Documentation	261
8.36.1.1	InvalidTimeException()	262
8.37	Rmatrix::IsSingular Class Reference	262
8.37.1	Constructor & Destructor Documentation	263
8.37.1.1	IsSingular()	263
8.38	LagrangeInterpolator Class Reference	263
8.38.1	Detailed Description	264
8.38.2	Constructor & Destructor Documentation	265
8.38.2.1	LagrangeInterpolator() [1/2]	265
8.38.2.2	~LagrangeInterpolator()	265
8.38.2.3	LagrangeInterpolator() [2/2]	265
8.38.3	Member Function Documentation	265
8.38.3.1	AddPoint()	266
8.38.3.2	AllocateArrays()	266
8.38.3.3	BuildDataPoints()	266
8.38.3.4	CleanupArrays()	266
8.38.3.5	Clear()	266
8.38.3.6	Clone()	267
8.38.3.7	CopyArrays()	267
8.38.3.8	FindStartingPoint()	267
8.38.3.9	Interpolate()	267
8.38.3.10	IsDataNearCenter()	268
8.38.3.11	IsInterpolationFeasible()	268
8.38.3.12	operator=()	268
8.38.3.13	UpdateBeginAndEndIndex()	269
8.38.4	Member Data Documentation	269
8.38.4.1	actualSize	269

8.38.4.2	beginIndex	269
8.38.4.3	dataIndex	269
8.38.4.4	endIndex	270
8.38.4.5	lastX	270
8.38.4.6	MAX_BUFFER_SIZE	270
8.38.4.7	order	270
8.38.4.8	startPoint	270
8.38.4.9	x	270
8.38.4.10	y	271
8.39	LeapSecondInformation Struct Reference	271
8.39.1	Detailed Description	271
8.39.2	Member Data Documentation	271
8.39.2.1	julianDate	271
8.39.2.2	offset1	271
8.39.2.3	offset2	272
8.39.2.4	offset3	272
8.39.2.5	taiMJD	272
8.40	LeapSecsFileReader Class Reference	272
8.40.1	Constructor & Destructor Documentation	272
8.40.1.1	LeapSecsFileReader() [1/2]	273
8.40.1.2	~LeapSecsFileReader()	273
8.40.1.3	LeapSecsFileReader() [2/2]	273
8.40.2	Member Function Documentation	273
8.40.2.1	GetFirstLeapSecondMJD()	273
8.40.2.2	Initialize()	274
8.40.2.3	IsInLeapSecond()	274
8.40.2.4	NumberOfLeapSecondsFrom()	274
8.40.2.5	operator=()	274
8.41	Date::LeapYearError Class Reference	275
8.41.1	Constructor & Destructor Documentation	275

8.41.1.1	LeapYearError()	276
8.42	LUFactorization Class Reference	276
8.42.1	Detailed Description	277
8.42.2	Constructor & Destructor Documentation	277
8.42.2.1	LUFactorization() [1/2]	277
8.42.2.2	LUFactorization() [2/2]	277
8.42.2.3	~LUFactorization()	277
8.42.3	Member Function Documentation	277
8.42.3.1	Determinant()	277
8.42.3.2	Factor()	278
8.42.3.3	Invert()	278
8.42.3.4	operator=()	278
8.42.3.5	SolveSystem()	279
8.43	MatrixFactorization Class Reference	279
8.43.1	Detailed Description	280
8.43.2	Constructor & Destructor Documentation	280
8.43.2.1	MatrixFactorization() [1/2]	280
8.43.2.2	MatrixFactorization() [2/2]	280
8.43.2.3	~MatrixFactorization()	280
8.43.3	Member Function Documentation	280
8.43.3.1	CompressNormalMatrix()	280
8.43.3.2	ExpandNormalMatrixInverse()	281
8.43.3.3	Factor()	281
8.43.3.4	Invert()	282
8.43.3.5	operator=()	282
8.43.3.6	PackedArrayIndex()	282
8.44	MessageInterface Class Reference	282
8.44.1	Detailed Description	283
8.44.2	Member Function Documentation	283
8.44.2.1	ClearMessage()	283

8.44.2.2	ClearMessageQueue()	283
8.44.2.3	GetLogEnable()	284
8.44.2.4	GetLogFileName()	284
8.44.2.5	GetMessageReceiver()	284
8.44.2.6	GetQueuedMessage()	284
8.44.2.7	LogMessage() [1/2]	284
8.44.2.8	LogMessage() [2/2]	284
8.44.2.9	PopupMessage() [1/2]	285
8.44.2.10	PopupMessage() [2/2]	285
8.44.2.11	PutMessage() [1/2]	285
8.44.2.12	PutMessage() [2/2]	286
8.44.2.13	SetLogEnable()	286
8.44.2.14	SetLogFile()	286
8.44.2.15	SetLogPath() [1/2]	286
8.44.2.16	SetLogPath() [2/2]	286
8.44.2.17	SetMessageReceiver()	287
8.44.2.18	ShowMessage() [1/2]	287
8.44.2.19	ShowMessage() [2/2]	287
8.44.3	Member Data Documentation	287
8.44.3.1	MAX_MESSAGE_LENGTH	287
8.45	MessageReceiver Class Reference	288
8.45.1	Detailed Description	288
8.45.2	Constructor & Destructor Documentation	289
8.45.2.1	MessageReceiver()	289
8.45.2.2	~MessageReceiver()	289
8.45.3	Member Function Documentation	289
8.45.3.1	ClearMessage()	289
8.45.3.2	ClearMessageQueue()	289
8.45.3.3	GetLogEnable()	289
8.45.3.4	GetLogFileName()	290

8.45.3.5	GetLogFileText()	290
8.45.3.6	GetMessage()	290
8.45.3.7	IsValidLogFile()	290
8.45.3.8	LogMessage() [1/2]	290
8.45.3.9	LogMessage() [2/2]	290
8.45.3.10	PopupMessage() [1/2]	291
8.45.3.11	PopupMessage() [2/2]	291
8.45.3.12	PutMessage()	291
8.45.3.13	SetLogEnable()	291
8.45.3.14	SetLogFile()	291
8.45.3.15	SetLogPath()	292
8.45.3.16	ShowMessage() [1/2]	292
8.45.3.17	ShowMessage() [2/2]	292
8.46	Rmatrix::NotSquare Class Reference	292
8.46.1	Constructor & Destructor Documentation	293
8.46.1.1	NotSquare()	293
8.47	ArrayTemplateExceptions::OutOfBounds Class Reference	293
8.47.1	Constructor & Destructor Documentation	294
8.47.1.1	OutOfBounds()	294
8.48	TableTemplateExceptions::OutOfBounds Class Reference	294
8.48.1	Constructor & Destructor Documentation	295
8.48.1.1	OutOfBounds()	295
8.49	Gmat::PluginResource Struct Reference	296
8.49.1	Constructor & Destructor Documentation	296
8.49.1.1	PluginResource()	296
8.49.2	Member Data Documentation	296
8.49.2.1	firstId	296
8.49.2.2	handler	296
8.49.2.3	lastId	297
8.49.2.4	nodeName	297

8.49.2.5	parentNodeName	297
8.49.2.6	subtype	297
8.49.2.7	toolkit	297
8.49.2.8	trigger	297
8.49.2.9	type	297
8.49.2.10	widgetType	298
8.50	QRFactorization Class Reference	298
8.50.1	Detailed Description	299
8.50.2	Constructor & Destructor Documentation	299
8.50.2.1	QRFactorization() [1/2]	299
8.50.2.2	QRFactorization() [2/2]	299
8.50.2.3	~QRFactorization()	299
8.50.3	Member Function Documentation	300
8.50.3.1	AddToQR()	300
8.50.3.2	Determinant()	300
8.50.3.3	Factor()	301
8.50.3.4	GetParameterMatrix()	301
8.50.3.5	Invert()	301
8.50.3.6	operator=()	301
8.50.3.7	RemoveFromQR()	302
8.51	GmatRealUtil::RaCodec Struct Reference	302
8.51.1	Member Data Documentation	302
8.51.1.1	coDeclinationD	302
8.51.1.2	radiusD	303
8.51.1.3	rightAscensionD	303
8.52	GmatRealUtil::RaDec Struct Reference	303
8.52.1	Member Data Documentation	303
8.52.1.1	declinationD	303
8.52.1.2	radiusD	303
8.52.1.3	rightAscensionD	304

8.53 RandomNumber Class Reference	304
8.53.1 Detailed Description	304
8.53.2 Constructor & Destructor Documentation	304
8.53.2.1 ~RandomNumber()	304
8.53.3 Member Function Documentation	305
8.53.3.1 Gaussian() [1/2]	305
8.53.3.2 Gaussian() [2/2]	305
8.53.3.3 GaussianArray() [1/2]	305
8.53.3.4 GaussianArray() [2/2]	305
8.53.3.5 Instance()	306
8.53.3.6 SetClockSeed()	306
8.53.3.7 SetSeed()	306
8.53.3.8 Uniform() [1/2]	306
8.53.3.9 Uniform() [2/2]	307
8.53.3.10 UniformArray() [1/2]	307
8.53.3.11 UniformArray() [2/2]	307
8.54 RealUtilitiesExceptions Struct Reference	308
8.54.1 Detailed Description	308
8.55 Rmatrix Class Reference	308
8.55.1 Constructor & Destructor Documentation	311
8.55.1.1 Rmatrix() [1/4]	311
8.55.1.2 Rmatrix() [2/4]	311
8.55.1.3 Rmatrix() [3/4]	311
8.55.1.4 Rmatrix() [4/4]	311
8.55.1.5 ~Rmatrix()	311
8.55.2 Member Function Documentation	311
8.55.2.1 AntiSymmetric()	312
8.55.2.2 Cofactor()	312
8.55.2.3 Determinant()	312
8.55.2.4 Diagonal()	312

8.55.2.5	<code>ElementWiseDivide()</code>	312
8.55.2.6	<code>ElementWiseMultiply()</code>	312
8.55.2.7	<code>GetColumn()</code>	313
8.55.2.8	<code>GetRow()</code>	313
8.55.2.9	<code>GetRowOrColumn()</code>	313
8.55.2.10	<code>GetStringVals()</code>	313
8.55.2.11	<code>Identity()</code>	313
8.55.2.12	<code>Inverse()</code> [1/2]	313
8.55.2.13	<code>Inverse()</code> [2/2]	313
8.55.2.14	<code>IsOrthogonal()</code>	314
8.55.2.15	<code>IsOrthonormal()</code>	314
8.55.2.16	<code>MakeOneColumnMatrix()</code>	314
8.55.2.17	<code>MakeOneRowMatrix()</code>	314
8.55.2.18	<code>operator"!=()"</code>	314
8.55.2.19	<code>operator*()</code> [1/3]	314
8.55.2.20	<code>operator*()</code> [2/3]	315
8.55.2.21	<code>operator*()</code> [3/3]	315
8.55.2.22	<code>operator*=()</code> [1/2]	315
8.55.2.23	<code>operator*=()</code> [2/2]	315
8.55.2.24	<code>operator+()</code> [1/2]	315
8.55.2.25	<code>operator+()</code> [2/2]	315
8.55.2.26	<code>operator+=()</code> [1/2]	315
8.55.2.27	<code>operator+=()</code> [2/2]	316
8.55.2.28	<code>operator-()</code> [1/3]	316
8.55.2.29	<code>operator-()</code> [2/3]	316
8.55.2.30	<code>operator-()</code> [3/3]	316
8.55.2.31	<code>operator-=()</code> [1/2]	316
8.55.2.32	<code>operator-=()</code> [2/2]	316
8.55.2.33	<code>operator/()</code> [1/2]	316
8.55.2.34	<code>operator/()</code> [2/2]	317

8.55.2.35 operator/=() [1/2]	317
8.55.2.36 operator/=() [2/2]	317
8.55.2.37 operator=()	317
8.55.2.38 operator==()	317
8.55.2.39 Pseudoinverse()	317
8.55.2.40 Symmetric()	317
8.55.2.41 ToRowString()	318
8.55.2.42 ToString() [1/2]	318
8.55.2.43 ToString() [2/2]	318
8.55.2.44 Trace()	318
8.55.2.45 Transpose()	318
8.55.3 Friends And Related Function Documentation	319
8.55.3.1 MatrixTimesTranspose	319
8.55.3.2 operator*	319
8.55.3.3 operator+	319
8.55.3.4 operator-	319
8.55.3.5 operator/	319
8.55.3.6 operator<<	320
8.55.3.7 operator>>	320
8.55.3.8 Rvector	320
8.55.3.9 Rvector3	320
8.55.3.10 SkewSymmetric4by4	320
8.55.3.11 TransposeTimesMatrix	320
8.55.3.12 TransposeTimesTranspose	320
8.55.4 Member Data Documentation	321
8.55.4.1 stringVals	321
8.56 Rmatrix33 Class Reference	321
8.56.1 Constructor & Destructor Documentation	323
8.56.1.1 Rmatrix33() [1/4]	323
8.56.1.2 Rmatrix33() [2/4]	323

8.56.1.3	Rmatrix33() [3/4]	323
8.56.1.4	Rmatrix33() [4/4]	323
8.56.1.5	~Rmatrix33()	324
8.56.2	Member Function Documentation	324
8.56.2.1	AntiSymmetric()	324
8.56.2.2	Determinant()	324
8.56.2.3	GetDataDescriptions()	324
8.56.2.4	Inverse()	324
8.56.2.5	IsOrthogonal()	324
8.56.2.6	IsOrthonormal()	325
8.56.2.7	operator!=()	325
8.56.2.8	operator*() [1/3]	325
8.56.2.9	operator*() [2/3]	325
8.56.2.10	operator*() [3/3]	325
8.56.2.11	operator*==() [1/2]	325
8.56.2.12	operator*==() [2/2]	326
8.56.2.13	operator+()	326
8.56.2.14	operator+=()	326
8.56.2.15	operator-() [1/2]	326
8.56.2.16	operator-() [2/2]	326
8.56.2.17	operator-=()	326
8.56.2.18	operator/() [1/2]	326
8.56.2.19	operator/() [2/2]	327
8.56.2.20	operator/=() [1/2]	327
8.56.2.21	operator/=() [2/2]	327
8.56.2.22	operator=()	327
8.56.2.23	operator==()	327
8.56.2.24	Set()	327
8.56.2.25	Symmetric()	328
8.56.2.26	Trace()	328

8.56.2.27	Transpose()	328
8.56.3	Friends And Related Function Documentation	328
8.56.3.1	MatrixTimesTranspose	328
8.56.3.2	operator*	328
8.56.3.3	Rvector3	328
8.56.3.4	SkewSymmetric	329
8.56.3.5	TransposeTimesMatrix	329
8.56.3.6	TransposeTimesTranspose	329
8.57	Rmatrix66 Class Reference	329
8.57.1	Constructor & Destructor Documentation	331
8.57.1.1	Rmatrix66() [1/4]	331
8.57.1.2	Rmatrix66() [2/4]	331
8.57.1.3	Rmatrix66() [3/4]	331
8.57.1.4	Rmatrix66() [4/4]	332
8.57.1.5	~Rmatrix66()	332
8.57.2	Member Function Documentation	332
8.57.2.1	AntiSymmetric()	332
8.57.2.2	Determinant()	332
8.57.2.3	Inverse()	332
8.57.2.4	IsOrthogonal()	332
8.57.2.5	IsOrthonormal()	333
8.57.2.6	LowerLeft()	333
8.57.2.7	LowerRight()	333
8.57.2.8	operator!=(())	333
8.57.2.9	operator*() [1/3]	333
8.57.2.10	operator*() [2/3]	333
8.57.2.11	operator*() [3/3]	333
8.57.2.12	operator*==([1/2]	334
8.57.2.13	operator*==([2/2]	334
8.57.2.14	operator+()	334

8.57.2.15 operator+=()	334
8.57.2.16 operator-() [1/2]	334
8.57.2.17 operator-() [2/2]	334
8.57.2.18 operator-=()	334
8.57.2.19 operator/() [1/2]	335
8.57.2.20 operator/() [2/2]	335
8.57.2.21 operator/=() [1/2]	335
8.57.2.22 operator/=() [2/2]	335
8.57.2.23 operator=()	335
8.57.2.24 operator==(())	335
8.57.2.25 Set()	335
8.57.2.26 SetUndefined()	336
8.57.2.27 Symmetric()	336
8.57.2.28 Trace()	336
8.57.2.29 Transpose()	336
8.57.2.30 UpperLeft()	336
8.57.2.31 UpperRight()	336
8.57.3 Friends And Related Function Documentation	336
8.57.3.1 MatrixTimesTranspose	336
8.57.3.2 operator*	337
8.57.3.3 Rvector6	337
8.57.3.4 SkewSymmetric	337
8.57.3.5 TransposeTimesMatrix	337
8.57.3.6 TransposeTimesTranspose	337
8.58 Rvector Class Reference	338
8.58.1 Constructor & Destructor Documentation	339
8.58.1.1 Rvector() [1/5]	340
8.58.1.2 Rvector() [2/5]	340
8.58.1.3 Rvector() [3/5]	340
8.58.1.4 Rvector() [4/5]	340

8.58.1.5	Rvector()	[5/5]	340
8.58.1.6	~Rvector()		340
8.58.2	Member Function Documentation		340
8.58.2.1	GetMagnitude()		341
8.58.2.2	GetRealArray()		341
8.58.2.3	GetUnitRvector()		341
8.58.2.4	IsZeroVector()		341
8.58.2.5	MakeZeroVector()		341
8.58.2.6	Max()		341
8.58.2.7	Min()		342
8.58.2.8	Norm()		342
8.58.2.9	Normalize()		342
8.58.2.10	operator"!="()		342
8.58.2.11	operator*()	[1/3]	342
8.58.2.12	operator*()	[2/3]	342
8.58.2.13	operator*()	[3/3]	343
8.58.2.14	operator*==()	[1/2]	343
8.58.2.15	operator*==()	[2/2]	343
8.58.2.16	operator+()		343
8.58.2.17	operator+=()		343
8.58.2.18	operator-()	[1/2]	343
8.58.2.19	operator-()	[2/2]	344
8.58.2.20	operator-=()		344
8.58.2.21	operator/()	[1/2]	344
8.58.2.22	operator/()	[2/2]	344
8.58.2.23	operator/=()	[1/2]	344
8.58.2.24	operator/=()	[2/2]	344
8.58.2.25	operator=()		344
8.58.2.26	operator==()		345
8.58.2.27	Set()	[1/2]	345

8.58.2.28 Set() [2/2]	345
8.58.2.29 Sort() [1/2]	345
8.58.2.30 Sort() [2/2]	345
8.58.2.31 ToString() [1/3]	346
8.58.2.32 ToString() [2/3]	346
8.58.2.33 ToString() [3/3]	346
8.58.3 Friends And Related Function Documentation	346
8.58.3.1 operator*	347
8.58.3.2 operator<<	347
8.58.3.3 operator>>	347
8.58.3.4 Outerproduct	347
8.58.3.5 Rmatrix	347
8.59 Rvector3 Class Reference	348
8.59.1 Constructor & Destructor Documentation	349
8.59.1.1 Rvector3() [1/4]	349
8.59.1.2 Rvector3() [2/4]	350
8.59.1.3 Rvector3() [3/4]	350
8.59.1.4 Rvector3() [4/4]	350
8.59.1.5 ~Rvector3()	350
8.59.2 Member Function Documentation	350
8.59.2.1 ComputeLongitudeLatitude()	350
8.59.2.2 Copy()	351
8.59.2.3 Get()	351
8.59.2.4 GetDataDescriptions()	351
8.59.2.5 GetMagnitude()	351
8.59.2.6 GetNumData()	351
8.59.2.7 GetUnitVector()	351
8.59.2.8 Normalize() [1/2]	351
8.59.2.9 Normalize() [2/2]	352
8.59.2.10 operator"!=()	352

8.59.2.11 operator*() [1/3]	352
8.59.2.12 operator*() [2/3]	352
8.59.2.13 operator*() [3/3]	352
8.59.2.14 operator*=() [1/2]	352
8.59.2.15 operator*=() [2/2]	352
8.59.2.16 operator+()	353
8.59.2.17 operator+=()	353
8.59.2.18 operator-() [1/2]	353
8.59.2.19 operator-() [2/2]	353
8.59.2.20 operator-=()	353
8.59.2.21 operator/() [1/2]	353
8.59.2.22 operator/() [2/2]	353
8.59.2.23 operator/=() [1/2]	354
8.59.2.24 operator/=() [2/2]	354
8.59.2.25 operator=()	354
8.59.2.26 operator==()	354
8.59.2.27 Set()	354
8.59.3 Friends And Related Function Documentation	354
8.59.3.1 Cross	354
8.59.3.2 operator*	355
8.59.3.3 Outerproduct	355
8.59.3.4 Rmatrix33	355
8.60 Rvector6 Class Reference	355
8.60.1 Constructor & Destructor Documentation	357
8.60.1.1 Rvector6() [1/6]	357
8.60.1.2 Rvector6() [2/6]	357
8.60.1.3 Rvector6() [3/6]	357
8.60.1.4 Rvector6() [4/6]	358
8.60.1.5 Rvector6() [5/6]	358
8.60.1.6 Rvector6() [6/6]	358

8.60.1.7	<code>~Rvector6()</code>	358
8.60.2	Member Function Documentation	358
8.60.2.1	<code>Clone()</code>	358
8.60.2.2	<code>Get()</code>	358
8.60.2.3	<code>GetDataDescriptions()</code>	359
8.60.2.4	<code>GetNumData()</code>	359
8.60.2.5	<code>GetR()</code> [1/2]	359
8.60.2.6	<code>GetR()</code> [2/2]	359
8.60.2.7	<code>GetV()</code> [1/2]	359
8.60.2.8	<code>GetV()</code> [2/2]	360
8.60.2.9	<code>IsValid()</code>	360
8.60.2.10	<code>operator"!=()</code>	360
8.60.2.11	<code>operator*()</code> [1/3]	360
8.60.2.12	<code>operator*()</code> [2/3]	360
8.60.2.13	<code>operator*()</code> [3/3]	360
8.60.2.14	<code>operator*==()</code> [1/2]	361
8.60.2.15	<code>operator*==()</code> [2/2]	361
8.60.2.16	<code>operator+()</code>	361
8.60.2.17	<code>operator+=()</code>	361
8.60.2.18	<code>operator-()</code> [1/2]	361
8.60.2.19	<code>operator-()</code> [2/2]	361
8.60.2.20	<code>operator-=()</code>	361
8.60.2.21	<code>operator/()</code> [1/2]	361
8.60.2.22	<code>operator/()</code> [2/2]	362
8.60.2.23	<code>operator/=(</code> [1/2]	362
8.60.2.24	<code>operator/=(</code> [2/2]	362
8.60.2.25	<code>operator=()</code>	362
8.60.2.26	<code>operator==(</code>	362
8.60.2.27	<code>Set()</code> [1/2]	363
8.60.2.28	<code>Set()</code> [2/2]	363

8.60.2.29	SetR()	363
8.60.2.30	SetV()	363
8.60.3	Friends And Related Function Documentation	363
8.60.3.1	Rmatrix66	363
8.60.4	Member Data Documentation	363
8.60.4.1	RVECTOR6_UNDEFINED	364
8.60.4.2	UTIL_REAL_UNDEFINED	364
8.61	SchurFactorization Class Reference	364
8.61.1	Detailed Description	365
8.61.2	Constructor & Destructor Documentation	365
8.61.2.1	SchurFactorization() [1/2]	365
8.61.2.2	SchurFactorization() [2/2]	366
8.61.2.3	~SchurFactorization()	366
8.61.3	Member Function Documentation	366
8.61.3.1	Factor()	366
8.61.3.2	Invert() [1/2]	366
8.61.3.3	Invert() [2/2]	367
8.61.3.4	operator=()	367
8.61.3.5	RemoveRowCol()	367
8.61.3.6	RestoreAllRowCols()	368
8.62	StateConversionUtil Class Reference	368
8.62.1	Detailed Description	371
8.62.2	Member Enumeration Documentation	371
8.62.2.1	AnomalyType	371
8.62.2.2	StateType	372
8.62.3	Member Function Documentation	372
8.62.3.1	AltEquinoctialToEquinoctial()	372
8.62.3.2	BrouwerMeanLongToCartesian()	372
8.62.3.3	BrouwerMeanLongToOsculatingElements()	373
8.62.3.4	BrouwerMeanShortToCartesian()	373

8.62.3.5	BrouwerMeanShortToOsculatingElements()	374
8.62.3.6	CalculateEccentricAnomaly()	374
8.62.3.7	CartesianToAngularMomentum()	374
8.62.3.8	CartesianToAOP()	375
8.62.3.9	CartesianToBrouwerMeanLong()	375
8.62.3.10	CartesianToBrouwerMeanShort()	375
8.62.3.11	CartesianToDirOfLineOfNode()	376
8.62.3.12	CartesianToEA()	376
8.62.3.13	CartesianToECC()	376
8.62.3.14	CartesianToEccVector()	376
8.62.3.15	CartesianToEquinoctial()	376
8.62.3.16	CartesianToHA()	377
8.62.3.17	CartesianToINC()	377
8.62.3.18	CartesianToIncomingAsymptote()	377
8.62.3.19	CartesianToKeplerian() ^[1/5]	378
8.62.3.20	CartesianToKeplerian() ^[2/5]	378
8.62.3.21	CartesianToKeplerian() ^[3/5]	378
8.62.3.22	CartesianToKeplerian() ^[4/5]	379
8.62.3.23	CartesianToKeplerian() ^[5/5]	379
8.62.3.24	CartesianToKeplerianDerivativeConversion()	380
8.62.3.25	CartesianToKeplerianDerivativeConversion_FiniteDiff()	380
8.62.3.26	CartesianToKeplerianDerivativeConversionWithKeplInput()	380
8.62.3.27	CartesianToKeplerianDerivativeConversionWithKeplInput_FiniteDiff()	381
8.62.3.28	CartesianToMA()	381
8.62.3.29	CartesianToModEquinoctial()	381
8.62.3.30	CartesianToOutgoingAsymptote()	382
8.62.3.31	CartesianToPlanetodetic()	382
8.62.3.32	CartesianToRAAN()	382
8.62.3.33	CartesianToSMA()	383
8.62.3.34	CartesianToSphericalAZFPA()	383

8.62.3.35 CartesianToSphericalRADEC()	383
8.62.3.36 CartesianToTA()	384
8.62.3.37 Convert() [1/2]	384
8.62.3.38 Convert() [2/2]	384
8.62.3.39 ConvertFromAltEquinoctial()	386
8.62.3.40 ConvertFromBrouwerMeanLong()	386
8.62.3.41 ConvertFromBrouwerMeanShort()	386
8.62.3.42 ConvertFromCartesian()	387
8.62.3.43 ConvertFromDelaunay()	387
8.62.3.44 ConvertFromEquinoctial()	387
8.62.3.45 ConvertFromIncomingAsymptote()	387
8.62.3.46 ConvertFromKeplerian()	388
8.62.3.47 ConvertFromModEquinoctial()	388
8.62.3.48 ConvertFromModKeplerian()	388
8.62.3.49 ConvertFromOutgoingAsymptote()	388
8.62.3.50 ConvertFromPlanetodetic()	389
8.62.3.51 ConvertFromSphericalAZFPA()	389
8.62.3.52 ConvertFromSphericalRADEC()	389
8.62.3.53 ConvertFromTrueAnomaly() [1/2]	389
8.62.3.54 ConvertFromTrueAnomaly() [2/2]	389
8.62.3.55 ConvertToTrueAnomaly() [1/2]	390
8.62.3.56 ConvertToTrueAnomaly() [2/2]	390
8.62.3.57 DelaunayToKeplerian()	390
8.62.3.58 EccentricToTrueAnomaly()	390
8.62.3.59 EquinoctialToAltEquinoctial()	391
8.62.3.60 EquinoctialToCartesian()	392
8.62.3.61 GetAnomalyLongText()	392
8.62.3.62 GetAnomalyShortText()	393
8.62.3.63 GetAnomalyType()	393
8.62.3.64 GetLongTypeNameList()	393

8.62.3.65	GetStateTypeList()	394
8.62.3.66	GetTypeCount()	394
8.62.3.67	HyperbolicToTrueAnomaly()	394
8.62.3.68	IncomingAsymptoteToCartesian()	394
8.62.3.69	IsRvValid()	395
8.62.3.70	IsValidAnomalyType()	395
8.62.3.71	KeplerianToCartesian() [1/2]	395
8.62.3.72	KeplerianToCartesian() [2/2]	396
8.62.3.73	KeplerianToDelaunay()	396
8.62.3.74	KeplerianToModKeplerian()	396
8.62.3.75	MeanToTrueAnomaly()	397
8.62.3.76	ModEquinoctialToCartesian()	397
8.62.3.77	ModKeplerianToKeplerian()	397
8.62.3.78	OutgoingAsymptoteToCartesian()	398
8.62.3.79	PlanetodeticToCartesian()	398
8.62.3.80	RequiresCelestialBodyOrigin()	399
8.62.3.81	RequiresFixedCoordinateSystem()	399
8.62.3.82	SphericalAZFPAToCartesian()	399
8.62.3.83	SphericalRADECToCartesian()	400
8.62.3.84	TrueToEccentricAnomaly()	400
8.62.3.85	TrueToHyperbolicAnomaly()	400
8.62.3.86	TrueToMeanAnomaly()	401
8.62.3.87	ValidateValue()	401
8.63	StringTokenizer Class Reference	401
8.63.1	Detailed Description	402
8.63.2	Constructor & Destructor Documentation	402
8.63.2.1	StringTokenizer() [1/3]	402
8.63.2.2	StringTokenizer() [2/3]	402
8.63.2.3	StringTokenizer() [3/3]	402
8.63.2.4	~StringTokenizer()	403

8.63.3 Member Function Documentation	403
8.63.3.1 CountTokens()	403
8.63.3.2 GetAllTokens()	403
8.63.3.3 GetDelimiters()	404
8.63.3.4 GetToken()	404
8.63.3.5 Set() [1/2]	404
8.63.3.6 Set() [2/2]	404
8.63.3.7 SetDelimiters()	405
8.64 TableTemplateExceptions::TableAlreadySized Class Reference	405
8.64.1 Constructor & Destructor Documentation	406
8.64.1.1 TableAlreadySized()	406
8.65 TableTemplate< T > Class Template Reference	406
8.65.1 Constructor & Destructor Documentation	407
8.65.1.1 TableTemplate() [1/4]	407
8.65.1.2 TableTemplate() [2/4]	408
8.65.1.3 TableTemplate() [3/4]	408
8.65.1.4 TableTemplate() [4/4]	408
8.65.1.5 ~TableTemplate()	408
8.65.2 Member Function Documentation	408
8.65.2.1 ChangeSize()	408
8.65.2.2 GetDataVector()	409
8.65.2.3 GetElement()	409
8.65.2.4 GetNumColumns()	409
8.65.2.5 GetNumRows()	409
8.65.2.6 GetSize()	409
8.65.2.7 init()	409
8.65.2.8 IsSized()	410
8.65.2.9 operator"!=()	410
8.65.2.10 operator()() [1/2]	410
8.65.2.11 operator()() [2/2]	410

8.65.2.12	<code>operator=()</code>	410
8.65.2.13	<code>operator==()</code>	410
8.65.2.14	<code>SetElement()</code>	411
8.65.2.15	<code>SetSize()</code>	411
8.65.3	Member Data Documentation	411
8.65.3.1	<code>colsD</code>	411
8.65.3.2	<code>elementD</code>	411
8.65.3.3	<code>isSizedD</code>	411
8.65.3.4	<code>rowsD</code>	412
8.66	<code>TableTemplateExceptions</code> Class Reference	412
8.66.1	Detailed Description	412
8.67	<code>TimeException</code> Class Reference	412
8.67.1	Constructor & Destructor Documentation	413
8.67.1.1	<code>TimeException()</code>	413
8.68	<code>TimeFileException</code> Class Reference	413
8.68.1	Constructor & Destructor Documentation	414
8.68.1.1	<code>TimeFileException()</code>	414
8.69	<code>TimeFormatException</code> Class Reference	414
8.69.1	Constructor & Destructor Documentation	415
8.69.1.1	<code>TimeFormatException()</code>	415
8.70	<code>Date::TimeRangeError</code> Class Reference	415
8.70.1	Constructor & Destructor Documentation	416
8.70.1.1	<code>TimeRangeError()</code>	416
8.71	<code>TimeSystemConverter</code> Class Reference	417
8.71.1	Member Enumeration Documentation	419
8.71.1.1	<code>TimeSystemTypes</code>	419
8.71.2	Constructor & Destructor Documentation	419
8.71.2.1	<code>TimeSystemConverter()</code> [1/2]	419
8.71.2.2	<code>TimeSystemConverter()</code> [2/2]	419
8.71.3	Member Function Documentation	419

8.71.3.1	Convert() [1/8]	420
8.71.3.2	Convert() [2/8]	420
8.71.3.3	Convert() [3/8]	420
8.71.3.4	Convert() [4/8]	421
8.71.3.5	Convert() [5/8]	421
8.71.3.6	Convert() [6/8]	421
8.71.3.7	Convert() [7/8]	421
8.71.3.8	Convert() [8/8]	422
8.71.3.9	ConvertFromTaiMjd() [1/2]	422
8.71.3.10	ConvertFromTaiMjd() [2/2]	422
8.71.3.11	ConvertGregorianToMjd()	423
8.71.3.12	ConvertGregorianToMjdGT()	423
8.71.3.13	ConvertMjdToGregorian()	423
8.71.3.14	ConvertToTaiMjd() [1/2]	424
8.71.3.15	ConvertToTaiMjd() [2/2]	424
8.71.3.16	GetFirstLeapSecondMJD()	425
8.71.3.17	GetTimeSystemAndFormat()	425
8.71.3.18	GetTimeTypeID()	425
8.71.3.19	GetValidTimeRepresentations()	425
8.71.3.20	Instance()	425
8.71.3.21	IsInLeapSecond() [1/2]	425
8.71.3.22	IsInLeapSecond() [2/2]	425
8.71.3.23	IsValidTimeSystem()	425
8.71.3.24	NumberOfLeapSecondsFrom()	426
8.71.3.25	SetEopFile()	426
8.71.3.26	SetLeapSecsFileReader()	426
8.71.3.27	ValidateTimeFormat()	427
8.71.3.28	ValidateTimeSystem()	427
8.71.4	Member Data Documentation	427
8.71.4.1	L_B	427

8.71.4.2	M_E_COEFF1	427
8.71.4.3	M_E_OFFSET	427
8.71.4.4	NUM_SECS	428
8.71.4.5	T_TT_COEFF1	428
8.71.4.6	T_TT_OFFSET	428
8.71.4.7	TDB_COEFF1	428
8.71.4.8	TDB_COEFF2	428
8.71.4.9	theEopFile	428
8.71.4.10	theLeapSecsFileReader	428
8.71.4.11	theTimeConverter	428
8.71.4.12	TIME_SYSTEM_TEXT	429
8.72	UnimplementedException Class Reference	429
8.72.1	Detailed Description	430
8.72.2	Constructor & Destructor Documentation	430
8.72.2.1	UnimplementedException()	430
8.73	ArrayTemplateExceptions::UnsizeArray Class Reference	430
8.73.1	Constructor & Destructor Documentation	431
8.73.1.1	UnsizeArray()	431
8.74	TableTemplateExceptions::UnsizeTable Class Reference	431
8.74.1	Constructor & Destructor Documentation	432
8.74.1.1	UnsizeTable()	432
8.75	UtcDate Class Reference	433
8.75.1	Constructor & Destructor Documentation	434
8.75.1.1	UtcDate() [1/7]	434
8.75.1.2	UtcDate() [2/7]	434
8.75.1.3	UtcDate() [3/7]	434
8.75.1.4	UtcDate() [4/7]	435
8.75.1.5	UtcDate() [5/7]	435
8.75.1.6	UtcDate() [6/7]	435
8.75.1.7	UtcDate() [7/7]	435
8.75.1.8	~UtcDate()	436
8.75.2	Member Function Documentation	436
8.75.2.1	operator=()	436
8.75.2.2	ToA1Mjd()	436
8.76	UtilityException Class Reference	437
8.76.1	Detailed Description	437
8.76.2	Constructor & Destructor Documentation	437
8.76.2.1	UtilityException()	438
8.77	Rvector::ZeroVector Class Reference	438
8.77.1	Constructor & Destructor Documentation	439
8.77.1.1	ZeroVector()	439

9 File Documentation	441
9.1 GMATsrc/base/ExponentialAtmosphere.cpp File Reference	441
9.2 GMATsrc/base/ExponentialAtmosphere.hpp File Reference	441
9.3 GMATsrc/console/ConsoleAppException.cpp File Reference	442
9.4 GMATsrc/console/ConsoleAppException.hpp File Reference	443
9.5 GMATsrc/console/ConsoleMessageReceiver.cpp File Reference	444
9.6 GMATsrc/console/ConsoleMessageReceiver.hpp File Reference	445
9.7 GMATsrc/include/gmatdefs.hpp File Reference	446
9.7.1 Macro Definition Documentation	448
9.7.1.1 DEFAULT_TO_NO_CLONES	448
9.7.1.2 DEFAULT_TO_NO_REFOBJECTS	448
9.7.1.3 GMAT_API	448
9.7.2 Typedef Documentation	448
9.7.2.1 BooleanArray	449
9.7.2.2 Byte	449
9.7.2.3 ColorMap	449
9.7.2.4 EpochArray	449
9.7.2.5 GEOPARMS	449
9.7.2.6 GmatEpoch	449
9.7.2.7 Integer	449
9.7.2.8 IntegerArray	450
9.7.2.9 IntegerMap	450
9.7.2.10 ObjectArray	450
9.7.2.11 ObjectMap	450
9.7.2.12 ObjectMapStack	450
9.7.2.13 Radians	450
9.7.2.14 Real	450
9.7.2.15 RealArray	450
9.7.2.16 StateArray	451
9.7.2.17 StringArray	451

9.7.2.18	UnsignedInt	451
9.7.2.19	UnsignedIntArray	451
9.7.2.20	WrapperArray	451
9.7.2.21	WrapperMap	451
9.8	GMATsrc/include/utldefs.hpp File Reference	452
9.8.1	Macro Definition Documentation	453
9.8.1.1	GMATUTIL_API	453
9.8.2	Typedef Documentation	454
9.8.2.1	BooleanArray	454
9.8.2.2	Byte	454
9.8.2.3	ColorMap	454
9.8.2.4	EpochArray	454
9.8.2.5	GmatEpoch	454
9.8.2.6	Integer	454
9.8.2.7	IntegerArray	454
9.8.2.8	IntegerMap	455
9.8.2.9	ObjectArray	455
9.8.2.10	ObjectMap	455
9.8.2.11	ObjectMapStack	455
9.8.2.12	ObjectTypeArray	455
9.8.2.13	ObjectTypeArrayMap	455
9.8.2.14	ObjectTypeMap	455
9.8.2.15	Radians	456
9.8.2.16	Real	456
9.8.2.17	RealArray	456
9.8.2.18	StateArray	456
9.8.2.19	StringArray	456
9.8.2.20	UnsignedInt	456
9.8.2.21	UnsignedIntArray	456
9.8.2.22	WrapperArray	457

9.8.2.23	WrapperMap	457
9.8.2.24	WrapperTypeArray	457
9.9	GMATsrc/util/A1Date.cpp File Reference	457
9.10	GMATsrc/util/A1Date.hpp File Reference	458
9.11	GMATsrc/util/A1Mjd.cpp File Reference	458
9.12	GMATsrc/util/A1Mjd.hpp File Reference	459
9.13	GMATsrc/util/ArrayTemplate.cpp File Reference	460
9.14	GMATsrc/util/ArrayTemplate.hpp File Reference	460
9.15	GMATsrc/util/AttitudeConversionUtility.cpp File Reference	461
9.16	GMATsrc/util/AttitudeConversionUtility.hpp File Reference	462
9.17	GMATsrc/util/BaseException.cpp File Reference	463
9.18	GMATsrc/util/BaseException.hpp File Reference	464
9.19	GMATsrc/util/BodyFixedStateConverter.cpp File Reference	464
9.20	GMATsrc/util/BodyFixedStateConverter.hpp File Reference	465
9.21	GMATsrc/util/Date.cpp File Reference	467
9.22	GMATsrc/util/Date.hpp File Reference	467
9.23	GMATsrc/util/DateUtil.cpp File Reference	468
9.23.1	Function Documentation	469
9.23.1.1	IsLeapYear()	469
9.23.1.2	IsValidTime()	469
9.23.1.3	JulianDate()	469
9.23.1.4	ModifiedJulianDate()	470
9.23.1.5	ModifiedJulianDateGT()	470
9.23.1.6	ToDOYFromYearMonthDay()	471
9.23.1.7	ToHMSFromSecondsOfDay()	471
9.23.1.8	ToMonthDayFromYearDOY()	472
9.23.1.9	ToSecondsOfDayFromHMS()	472
9.23.1.10	UnpackDate()	472
9.23.1.11	UnpackDateWithDOY()	473
9.23.1.12	UnpackTime()	473

9.24	GMATsrc/util/DateUtil.hpp File Reference	473
9.24.1	Function Documentation	474
9.24.1.1	IsLeapYear()	474
9.24.1.2	IsValidTime()	475
9.24.1.3	JulianDate()	475
9.24.1.4	ModifiedJulianDate()	475
9.24.1.5	ModifiedJulianDateGT()	476
9.24.1.6	ToDOYFromYearMonthDay()	476
9.24.1.7	ToHMSFromSecondsOfDay()	477
9.24.1.8	ToMonthDayFromYearDOY()	477
9.24.1.9	ToSecondsOfDayFromHMS()	477
9.24.1.10	UnpackDate()	478
9.24.1.11	UnpackDateWithDOY()	478
9.24.1.12	UnpackTime()	478
9.25	GMATsrc/util/ElapsedTime.cpp File Reference	479
9.26	GMATsrc/util/ElapsedTime.hpp File Reference	479
9.27	GMATsrc/util/EopFile.cpp File Reference	480
9.28	GMATsrc/util/EopFile.hpp File Reference	480
9.29	GMATsrc/util/FileManager.cpp File Reference	481
9.30	GMATsrc/util/FileManager.hpp File Reference	482
9.31	GMATsrc/util/FileTypes.hpp File Reference	483
9.32	GMATsrc/util/FileUtil.cpp File Reference	484
9.33	GMATsrc/util/FileUtil.hpp File Reference	484
9.34	GMATsrc/util/GmatConstants.hpp File Reference	486
9.35	GMATsrc/util/GmatDefaults.hpp File Reference	488
9.36	GMATsrc/util/GmatGlobal.cpp File Reference	491
9.37	GMATsrc/util/GmatGlobal.hpp File Reference	492
9.38	GMATsrc/util/GmatTime.cpp File Reference	493
9.39	GMATsrc/util/GmatTime.hpp File Reference	493
9.40	GMATsrc/util/GregorianCalendar.cpp File Reference	494

9.41	GMATsrc/util/GregorianCalendar.cpp File Reference	495
9.42	GMATsrc/util/interpolator/Interpolator.cpp File Reference	496
9.43	GMATsrc/util/interpolator/Interpolator.hpp File Reference	497
9.44	GMATsrc/util/interpolator/InterpolatorException.cpp File Reference	498
9.45	GMATsrc/util/interpolator/InterpolatorException.hpp File Reference	498
9.46	GMATsrc/util/interpolator/LagrangeInterpolator.cpp File Reference	499
9.47	GMATsrc/util/interpolator/LagrangeInterpolator.hpp File Reference	500
9.48	GMATsrc/util/LeapSecsFileReader.cpp File Reference	500
9.49	GMATsrc/util/LeapSecsFileReader.hpp File Reference	501
9.50	GMATsrc/util/Linear.cpp File Reference	502
9.51	GMATsrc/util/Linear.hpp File Reference	503
9.52	GMATsrc/util/matrixoperations/CholeskyFactorization.cpp File Reference	504
9.53	GMATsrc/util/matrixoperations/CholeskyFactorization.hpp File Reference	504
9.54	GMATsrc/util/matrixoperations/LUFactorization.cpp File Reference	505
9.55	GMATsrc/util/matrixoperations/LUFactorization.hpp File Reference	505
9.56	GMATsrc/util/matrixoperations/MatrixFactorization.cpp File Reference	506
9.57	GMATsrc/util/matrixoperations/MatrixFactorization.hpp File Reference	507
9.58	GMATsrc/util/matrixoperations/QRFactorization.cpp File Reference	507
9.59	GMATsrc/util/matrixoperations/QRFactorization.hpp File Reference	508
9.60	GMATsrc/util/matrixoperations/SchurFactorization.cpp File Reference	509
9.61	GMATsrc/util/matrixoperations/SchurFactorization.hpp File Reference	509
9.62	GMATsrc/util/MessageInterface.cpp File Reference	510
9.63	GMATsrc/util/MessageInterface.hpp File Reference	511
9.64	GMATsrc/util/MessageReceiver.cpp File Reference	512
9.65	GMATsrc/util/MessageReceiver.hpp File Reference	513
9.66	GMATsrc/util/RandomNumber.cpp File Reference	514
9.67	GMATsrc/util/RandomNumber.hpp File Reference	514
9.68	GMATsrc/util/RealUtilities.cpp File Reference	515
9.69	GMATsrc/util/RealUtilities.hpp File Reference	516
9.70	GMATsrc/util/Rmatrix.cpp File Reference	518

9.70.1	Function Documentation	519
9.70.1.1	MatrixTimesTranspose()	519
9.70.1.2	operator*()	519
9.70.1.3	operator+()	519
9.70.1.4	operator-()	519
9.70.1.5	operator/()	520
9.70.1.6	operator<<()	520
9.70.1.7	operator>>()	520
9.70.1.8	SkewSymmetric4by4()	520
9.70.1.9	TransposeTimesMatrix()	520
9.70.1.10	TransposeTimesTranspose()	521
9.71	GMATsrc/util/Rmatrix.hpp File Reference	521
9.72	GMATsrc/util/Rmatrix33.cpp File Reference	521
9.72.1	Function Documentation	522
9.72.1.1	MatrixTimesTranspose()	522
9.72.1.2	operator*()	522
9.72.1.3	SkewSymmetric()	523
9.72.1.4	TransposeTimesRmatrix()	523
9.72.1.5	TransposeTimesTranspose()	523
9.73	GMATsrc/util/Rmatrix33.hpp File Reference	523
9.74	GMATsrc/util/Rmatrix66.cpp File Reference	524
9.74.1	Function Documentation	524
9.74.1.1	MatrixTimesTranspose()	524
9.74.1.2	operator*()	525
9.74.1.3	SkewSymmetric()	525
9.74.1.4	TransposeTimesMatrix()	525
9.74.1.5	TransposeTimesTranspose()	525
9.75	GMATsrc/util/Rmatrix66.hpp File Reference	525
9.76	GMATsrc/util/Rvector.cpp File Reference	526
9.76.1	Function Documentation	527

9.76.1.1	operator*()	527
9.76.1.2	operator<()	527
9.76.1.3	operator<<()	527
9.76.1.4	operator>>()	528
9.76.1.5	Outerproduct()	528
9.77	GMATsrc/util/Rvector.hpp File Reference	528
9.77.1	Function Documentation	529
9.77.1.1	operator<()	529
9.78	GMATsrc/util/Rvector3.cpp File Reference	529
9.78.1	Function Documentation	530
9.78.1.1	Cross()	530
9.78.1.2	operator*()	530
9.78.1.3	operator<<()	530
9.78.1.4	operator>>()	531
9.78.1.5	Outerproduct()	531
9.79	GMATsrc/util/Rvector3.hpp File Reference	531
9.80	GMATsrc/util/Rvector6.cpp File Reference	532
9.81	GMATsrc/util/Rvector6.hpp File Reference	532
9.81.1	Macro Definition Documentation	533
9.81.1.1	NUM_DATA_INIT	533
9.82	GMATsrc/util/StateConversionUtil.cpp File Reference	533
9.83	GMATsrc/util/StateConversionUtil.hpp File Reference	534
9.83.1	Macro Definition Documentation	535
9.83.1.1	EARTH_EQ_RADIUS	535
9.83.1.2	EARTH_FLATTENING	535
9.83.1.3	EARTH_MU	536
9.84	GMATsrc/util/StringTokenizer.cpp File Reference	536
9.85	GMATsrc/util/StringTokenizer.hpp File Reference	536
9.86	GMATsrc/util/StringUtil.cpp File Reference	537
9.87	GMATsrc/util/StringUtil.hpp File Reference	538

9.88	GMATsrc/util/TableTemplate.cpp File Reference	542
9.89	GMATsrc/util/TableTemplate.hpp File Reference	543
9.90	GMATsrc/util/TimeSystemConverter.cpp File Reference	544
9.91	GMATsrc/util/TimeSystemConverter.hpp File Reference	545
9.92	GMATsrc/util/TimeTypes.cpp File Reference	546
9.93	GMATsrc/util/TimeTypes.hpp File Reference	546
9.93.1	Typedef Documentation	547
9.93.1.1	DayOfMonth	547
9.93.1.2	DayOfYear	547
9.93.1.3	HourOfDay	548
9.93.1.4	MinuteOfHour	548
9.93.1.5	MonthOfYear	548
9.93.1.6	Ut1Mjd	548
9.93.1.7	UtcMjd	548
9.93.1.8	YearNumber	548
9.94	GMATsrc/util/UtcDate.cpp File Reference	548
9.95	GMATsrc/util/UtcDate.hpp File Reference	549
9.96	GMATsrc/util/UtilityException.hpp File Reference	549
Index		551

Chapter 1

Test List

Class [ExponentialAtmosphere](#)

Check to see if the band discontinuities merit smoothing.

Chapter 2

Todo List

Class **ExponentialAtmosphere**

Replace the spherical Earth model with an oblate Earth model.

Member **FileManager::ReadStartupFile** (const std::string &fileName="")

This code replaces relative paths with absolute. It was implemented to address an issue in R2014a, but the side effects were too severe for the release. It is commented out so that post release, we can assess how to proceed addressing path issues in GMAT.

Member **Gmat::GENERIC_OBJECT**

: DJC - Do we need this for backwards compatibility?

Member **Interpolator::AddPoint** (const Real ind, const Real *data)

Handle memory access violations when the input array is too small.

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

BodyFixedStateConverterUtil	15
Gmat	20
GmatAttitudeConstants	26
GmatEop	26
GmatFile	27
GmatFileUtil	27
GmatIntegerConstants	37
GmatMathConstants	37
GmatMathUtil	38
GmatOrbitConstants	50
GmatPhysicalConstants	50
GmatRealConstants	51
GmatRealUtil	53
GmatSolarSystemDefaults	57
GmatStringUtil	75
GmatTimeConstants	102
GmatTimeUtil	107

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

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

A1Mjd	113
ArrayTemplate< T >	124
ArrayTemplate< Real >	124
Rvector	338
Rvector3	348
Rvector6	355
ArrayTemplateExceptions	129
AttitudeConversionUtility	130
BaseException	137
ArrayTemplateExceptions::ArrayAlreadySized	123
ArrayTemplateExceptions::DimensionError	173
ArrayTemplateExceptions::IllegalSize	247
ArrayTemplateExceptions::OutOfBounds	293
ArrayTemplateExceptions::UnsizeArray	430
ConsoleAppException	147
Date::LeapYearError	275
Date::TimeRangeError	415
GravityFileException	239
GregorianDate::GregorianDateException	244
InterpolatorException	258
InvalidStateRepresentationException	259
InvalidTimeException	261
RealUtilitiesExceptions::ArgumentError	122
RealUtilitiesExceptions::IllegalTime	248
Rmatrix::DivideByZero	174
Rmatrix::IsSingular	262
Rmatrix::NotSquare	292
Rvector::ZeroVector	438
TableTemplateExceptions::DimensionError	172
TableTemplateExceptions::IllegalSize	245
TableTemplateExceptions::OutOfBounds	294
TableTemplateExceptions::TableAlreadySized	405
TableTemplateExceptions::UnsizeTable	431
TimeException	412
TimeFileException	413

TimeFormatException	414
UnimplementedException	429
UtilityException	437
GmatTimeUtil::CalDate	141
Date	155
A1Date	109
UtcDate	433
DateUtil	163
GmatTimeUtil::ElapsedDate	175
ElapsedTime	177
Element	180
EopFile	181
ExponentialAtmosphere	188
FileManager	193
geoparms	211
GmatGlobal	212
GmatTime	231
GregorianDate	240
Interpolator	249
LagrangeInterpolator	263
LeapSecondInformation	271
LeapSecsFileReader	272
MatrixFactorization	279
CholeskyFactorization	144
LUFactorization	276
QRFactorization	298
SchurFactorization	364
MessageInterface	282
MessageReceiver	288
ConsoleMessageReceiver	148
Gmat::PluginResource	296
GmatRealUtil::RaCodec	302
GmatRealUtil::RaDec	303
RandomNumber	304
RealUtilitiesExceptions	308
StateConversionUtil	368
StringTokenizer	401
TableTemplate< T >	406
TableTemplate< Real >	406
Rmatrix	308
Rmatrix33	321
Rmatrix66	329
TableTemplateExceptions	412
TimeSystemConverter	417

Chapter 5

Class Index

5.1 Class List

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

A1Date	109
A1Mjd	113
RealUtilitiesExceptions::ArgumentError	122
ArrayTemplateExceptions::ArrayAlreadySized	123
ArrayTemplate< T >	124
ArrayTemplateExceptions	129
AttitudeConversionUtility	130
BaseException	137
GmatTimeUtil::CalDate	141
CholeskyFactorization	144
ConsoleAppException	147
ConsoleMessageReceiver	148
Date	155
DateUtil	163
TableTemplateExceptions::DimensionError	172
ArrayTemplateExceptions::DimensionError	173
Rmatrix::DivideByZero	174
GmatTimeUtil::ElapsedDate	175
ElapsedTime	177
Element	180
EopFile	181
ExponentialAtmosphere	188
FileManager	193
geoparms	211
GmatGlobal	212
GmatTime	231
GravityFileException	239
GregorianDate	240
GregorianDate::GregorianDateException	244
TableTemplateExceptions::IllegalSize	245
ArrayTemplateExceptions::IllegalSize	247
RealUtilitiesExceptions::IllegalTime	248
Interpolator	249
InterpolatorException	258
InvalidStateRepresentationException	259

InvalidTimeException	261
Rmatrix::IsSingular	262
LagrangeInterpolator	263
LeapSecondInformation	271
LeapSecsFileReader	272
Date::LeapYearError	275
LUFactorization	276
MatrixFactorization	279
MessageInterface	282
MessageReceiver	288
Rmatrix::NotSquare	292
ArrayTemplateExceptions::OutOfBounds	293
TableTemplateExceptions::OutOfBounds	294
Gmat::PluginResource	296
QRFactorization	298
GmatRealUtil::RaCodec	302
GmatRealUtil::RaDec	303
RandomNumber	304
RealUtilitiesExceptions	308
Rmatrix	308
Rmatrix33	321
Rmatrix66	329
Rvector	338
Rvector3	348
Rvector6	355
SchurFactorization	364
StateConversionUtil	368
StringTokenizer	401
TableTemplateExceptions::TableAlreadySized	405
TableTemplate< T >	406
TableTemplateExceptions	412
TimeException	412
TimeFileException	413
TimeFormatException	414
Date::TimeRangeError	415
TimeSystemConverter	417
UnimplementedException	429
ArrayTemplateExceptions::UnsizeArray	430
TableTemplateExceptions::UnsizeTable	431
UtcDate	433
UtilityException	437
Rvector::ZeroVector	438

Chapter 6

File Index

6.1 File List

Here is a list of all files with brief descriptions:

GMATsrc/base/ExponentialAtmosphere.cpp	441
GMATsrc/base/ExponentialAtmosphere.hpp	441
GMATsrc/console/ConsoleAppException.cpp	442
GMATsrc/console/ConsoleAppException.hpp	443
GMATsrc/console/ConsoleMessageReceiver.cpp	444
GMATsrc/console/ConsoleMessageReceiver.hpp	445
GMATsrc/include/gmatdefs.hpp	446
GMATsrc/include/utldefs.hpp	452
GMATsrc/util/A1Date.cpp	457
GMATsrc/util/A1Date.hpp	458
GMATsrc/util/A1Mjd.cpp	458
GMATsrc/util/A1Mjd.hpp	459
GMATsrc/util/ArrayTemplate.cpp	460
GMATsrc/util/ArrayTemplate.hpp	460
GMATsrc/util/AttitudeConversionUtility.cpp	461
GMATsrc/util/AttitudeConversionUtility.hpp	462
GMATsrc/util/BaseException.cpp	463
GMATsrc/util/BaseException.hpp	464
GMATsrc/util/BodyFixedStateConverter.cpp	464
GMATsrc/util/BodyFixedStateConverter.hpp	465
GMATsrc/util/Date.cpp	467
GMATsrc/util/Date.hpp	467
GMATsrc/util/DateUtil.cpp	468
GMATsrc/util/DateUtil.hpp	473
GMATsrc/util/ElapsedTime.cpp	479
GMATsrc/util/ElapsedTime.hpp	479
GMATsrc/util/EopFile.cpp	480
GMATsrc/util/EopFile.hpp	480
GMATsrc/util/FileManager.cpp	481
GMATsrc/util/FileManager.hpp	482
GMATsrc/util/FileTypes.hpp	483
GMATsrc/util/FileUtil.cpp	484
GMATsrc/util/FileUtil.hpp	484
GMATsrc/util/GmatConstants.hpp	486
GMATsrc/util/GmatDefaults.hpp	488

GMATsrc/util/GmatGlobal.cpp	491
GMATsrc/util/GmatGlobal.hpp	492
GMATsrc/util/GmatTime.cpp	493
GMATsrc/util/GmatTime.hpp	493
GMATsrc/util/GregorianCalendar.cpp	494
GMATsrc/util/GregorianCalendar.hpp	495
GMATsrc/util/LeapSecsFileReader.cpp	500
GMATsrc/util/LeapSecsFileReader.hpp	501
GMATsrc/util/Linear.cpp	502
GMATsrc/util/Linear.hpp	503
GMATsrc/util/MessageInterface.cpp	510
GMATsrc/util/MessageInterface.hpp	511
GMATsrc/util/MessageReceiver.cpp	512
GMATsrc/util/MessageReceiver.hpp	513
GMATsrc/util/RandomNumber.cpp	514
GMATsrc/util/RandomNumber.hpp	514
GMATsrc/util/RealUtilities.cpp	515
GMATsrc/util/RealUtilities.hpp	516
GMATsrc/util/Rmatrix.cpp	518
GMATsrc/util/Rmatrix.hpp	521
GMATsrc/util/Rmatrix33.cpp	521
GMATsrc/util/Rmatrix33.hpp	523
GMATsrc/util/Rmatrix66.cpp	524
GMATsrc/util/Rmatrix66.hpp	525
GMATsrc/util/Rvector.cpp	526
GMATsrc/util/Rvector.hpp	528
GMATsrc/util/Rvector3.cpp	529
GMATsrc/util/Rvector3.hpp	531
GMATsrc/util/Rvector6.cpp	532
GMATsrc/util/Rvector6.hpp	532
GMATsrc/util/StateConversionUtil.cpp	533
GMATsrc/util/StateConversionUtil.hpp	534
GMATsrc/util/StringTokenizer.cpp	536
GMATsrc/util/StringTokenizer.hpp	536
GMATsrc/util/StringUtil.cpp	537
GMATsrc/util/StringUtil.hpp	538
GMATsrc/util/TableTemplate.cpp	542
GMATsrc/util/TableTemplate.hpp	543
GMATsrc/util/TimeSystemConverter.cpp	544
GMATsrc/util/TimeSystemConverter.hpp	545
GMATsrc/util/TimeTypes.cpp	546
GMATsrc/util/TimeTypes.hpp	546
GMATsrc/util/UtcDate.cpp	548
GMATsrc/util/UtcDate.hpp	549
GMATsrc/util/UtilityException.hpp	549
GMATsrc/util/interpolator/Interpolator.cpp	496
GMATsrc/util/interpolator/Interpolator.hpp	497
GMATsrc/util/interpolator/InterpolatorException.cpp	498
GMATsrc/util/interpolator/InterpolatorException.hpp	498
GMATsrc/util/interpolator/LagrangeInterpolator.cpp	499
GMATsrc/util/interpolator/LagrangeInterpolator.hpp	500
GMATsrc/util/matrixoperations/CholeskyFactorization.cpp	504
GMATsrc/util/matrixoperations/CholeskyFactorization.hpp	504
GMATsrc/util/matrixoperations/LUFactorization.cpp	505
GMATsrc/util/matrixoperations/LUFactorization.hpp	505
GMATsrc/util/matrixoperations/MatrixFactorization.cpp	506
GMATsrc/util/matrixoperations/MatrixFactorization.hpp	507
GMATsrc/util/matrixoperations/QRFactorization.cpp	507

GMATsrc/util/matrixoperations/ QRFactorization.hpp	508
GMATsrc/util/matrixoperations/ SchurFactorization.cpp	509
GMATsrc/util/matrixoperations/ SchurFactorization.hpp	509

Chapter 7

Namespace Documentation

7.1 BodyFixedStateConverterUtil Namespace Reference

Functions

- [Rvector3 GMATUTIL_API Convert](#) (const [Rvector3](#) &origValue, const std::string &fromType, const std::string &toType, const [Real](#) flattening, const [Real](#) meanRadius)
- [Rvector3 GMATUTIL_API Convert](#) (const [Rvector3](#) &origValue, const std::string &fromType, const std::string &fromHorizon, const std::string &toType, const std::string &toHorizon, const [Real](#) flattening, const [Real](#) meanRadius)
- [Rvector3 GMATUTIL_API CartesianToSpherical](#) (const [Rvector3](#) &cart, const [Real](#) flattening, const [Real](#) meanRadius)
- [Rvector3 GMATUTIL_API SphericalToCartesian](#) (const [Rvector3](#) &spherical, const [Real](#) flattening, const [Real](#) meanRadius)
- [Rvector3 GMATUTIL_API SphericalEllipsoidToCartesian](#) (const [Rvector3](#) &sphEll, const [Real](#) flattening, const [Real](#) meanRadius)
- [Rvector3 GMATUTIL_API CartesianToSphericalEllipsoid](#) (const [Rvector3](#) &cart, const [Real](#) flattening, const [Real](#) meanRadius)
- [Rvector3 GMATUTIL_API SphericalToSphericalEllipsoid](#) (const [Rvector3](#) &spherical, const [Real](#) flattening, const [Real](#) meanRadius)
- [Rvector3 GMATUTIL_API SphericalEllipsoidToSpherical](#) (const [Rvector3](#) &sphEll, const [Real](#) flattening, const [Real](#) meanRadius)
- [bool GMATUTIL_API IsValidStateRepresentation](#) (const std::string &rep)
- [StringArray GMATUTIL_API GetValidRepresentations](#) ()

7.1.1 Function Documentation

7.1.1.1 CartesianToSpherical()

```
Rvector3 BodyFixedStateConverterUtil::CartesianToSpherical (  
    const Rvector3 & cart,  
    const Real flattening,  
    const Real meanRadius )
```

Method to convert from Cartesian to Spherical.

Parameters

< <i>cart</i> >	data in cartesian
< <i>flattening</i> >	flattening coefficient for the body
< <i>meanRadius</i> >	mean radius of the body

Returns

Converted state from the cartesian representation to the spherical representation

7.1.1.2 CartesianToSphericalEllipsoid()

```
Rvector3 BodyFixedStateConverterUtil::CartesianToSphericalEllipsoid (
    const Rvector3 & cart,
    const Real flattening,
    const Real meanRadius )
```

Method to convert from Cartesian to Spherical-Ellipsoid.

Parameters

< <i>cart</i> >	data in cartesian
< <i>flattening</i> >	flattening coefficient for the body
< <i>meanRadius</i> >	mean radius of the body

Returns

Converted state from the cartesian representation to the spherical-ellipsoid representation

7.1.1.3 Convert() [1/2]

```
Rvector3 BodyFixedStateConverterUtil::Convert (
    const Rvector3 & origValue,
    const std::string & fromType,
    const std::string & toType,
    const Real flattening,
    const Real meanRadius )
```

Implements the BodyFixedStateConverter namespace Conversion method.

Parameters

< <i>origValue</i> >	data in given representation
< <i>fromType</i> >	representation from which to convert
< <i>toType</i> >	representation to which to convert
< <i>flattening</i> >	flattening coefficient for the body
< <i>meanRadius</i> >	mean radius of the body

Returns

Converted state from the specified "to" representation to the specified "from" representation

7.1.1.4 Convert() [2/2]

```
Rvector3 BodyFixedStateConverterUtil::Convert (
    const Rvector3 & origValue,
    const std::string & fromType,
    const std::string & fromHorizon,
    const std::string & toType,
    const std::string & toHorizon,
    const Real flattening,
    const Real meanRadius )
```

Conversion method.

Parameters

<origValue>	data in given representation
<fromType>	state type from which to convert
<fromHorizon>	horizon reference value from which to convert
<toType>	state type to which to convert
<toHorizon>	horizon reference to which to convert
<flattening>	flattening coefficient for the body
<meanRadius>	mean radius of the body

Returns

Converted state from the specified "to" representation to the specified "from" representation

7.1.1.5 GetValidRepresentations()

```
StringArray BodyFixedStateConverterUtil::GetValidRepresentations ( )
```

7.1.1.6 IsValidStateRepresentation()

```
bool BodyFixedStateConverterUtil::IsValidStateRepresentation (
    const std::string & rep )
```

7.1.1.7 SphericalEllipsoidToCartesian()

```
Rvector3 BodyFixedStateConverterUtil::SphericalEllipsoidToCartesian (
    const Rvector3 & sphEll,
    const Real flattening,
    const Real meanRadius )
```

Method to convert from Spherical-Ellipsoid to Cartesian.

Parameters

< <i>sphEll</i> >	data in spherical-ellipsoid
< <i>flattening</i> >	flattening coefficient for the body
< <i>meanRadius</i> >	mean radius of the body

Returns

Converted state from the spherical-ellipsoid representation to the cartesian representation

7.1.1.8 SphericalEllipsoidToSpherical()

```
Rvector3 BodyFixedStateConverterUtil::SphericalEllipsoidToSpherical (
    const Rvector3 & sphEll,
    const Real flattening,
    const Real meanRadius )
```

Method to convert from Spherical-Ellipsoid to Spherical.

Parameters

< <i>sphEll</i> >	data in spherical-ellipsoid
< <i>flattening</i> >	flattening coefficient for the body
< <i>meanRadius</i> >	mean radius of the body

Returns

Converted state from the spherical-ellipsoid representation to the spherical representation

7.1.1.9 SphericalToCartesian()

```
Rvector3 BodyFixedStateConverterUtil::SphericalToCartesian (
    const Rvector3 & spherical,
    const Real flattening,
    const Real meanRadius )
```

Method to convert from Spherical to Cartesian.

Parameters

< <i>spherical</i> >	data in spherical
< <i>flattening</i> >	flattening coefficient for the body
< <i>meanRadius</i> >	mean radius of the body

Returns

Converted state from the spherical representation to the cartesian representation

7.1.1.10 SphericalToSphericalEllipsoid()

```
Rvector3 BodyFixedStateConverterUtil::SphericalToSphericalEllipsoid (
    const Rvector3 & spherical,
    const Real flattening,
    const Real meanRadius )
```

Method to convert from Spherical to Spherical-Ellipsoid.

Parameters

< <i>spherical</i> >	data in spherical
< <i>flattening</i> >	flattening coefficient for the body
< <i>meanRadius</i> >	mean radius of the body

Returns

Converted state from the spherical representation to the spherical-ellipsoid representation

7.2 Gmat Namespace Reference**Classes**

- struct [PluginResource](#)

Typedefs

- typedef struct [Gmat::PluginResource](#) [PLUGIN_RESOURCE](#)

Enumerations

- enum [ObjectType](#) {
[SPACECRAFT](#) = 101, [FORMATION](#), [SPACEOBJECT](#), [GROUND_STATION](#),
[BURN](#), [IMPULSIVE_BURN](#), [FINITE_BURN](#), [COMMAND](#),
[PROPAGATOR](#), [ODE_MODEL](#), [PHYSICAL_MODEL](#), [TRANSIENT_FORCE](#),
[INTERPOLATOR](#), [SOLAR_SYSTEM](#), [SPACE_POINT](#), [CELESTIAL_BODY](#),
[CALCULATED_POINT](#), [LIBRATION_POINT](#), [BARYCENTER](#), [ATMOSPHERE](#),
[PARAMETER](#), [VARIABLE](#), [ARRAY](#), [STRING](#),
[STOP_CONDITION](#), [SOLVER](#), [SUBSCRIBER](#), [REPORT_FILE](#),
[XY_PLOT](#), [ORBIT_VIEW](#), [DYNAMIC_DATA_DISPLAY](#), [EPHEMERIS_FILE](#),
[PROP_SETUP](#), [FUNCTION](#), [FUEL_TANK](#), [THRUSTER](#),
[CHEMICAL_THRUSTER](#), [ELECTRIC_THRUSTER](#), [CHEMICAL_FUEL_TANK](#), [ELECTRIC_FUEL_TANK](#),
[POWER_SYSTEM](#), [SOLAR_POWER_SYSTEM](#), [NUCLEAR_POWER_SYSTEM](#), [HARDWARE](#),

- ```

COORDINATE_SYSTEM, AXIS_SYSTEM, ATTITUDE, MATH_NODE,
MATH_TREE, BODY_FIXED_POINT, EVENT, EVENT_LOCATOR,
DATAINTERFACE_SOURCE, MEASUREMENT_MODEL, ERROR_MODEL, DATASTREAM,
DATA_FILE, OBTYPE, DATA_FILTER, INTERFACE,
MEDIA_CORRECTION, SENSOR, RF_HARDWARE, ANTENNA,
USER_DEFINED_OBJECT, USER_OBJECT_ID_NEEDED = USER_DEFINED_OBJECT + 500, GENERI↵
C_OBJECT, UNKNOWN_OBJECT }

```
- enum `WriteMode` {  
`SCRIPTING`, `SHOW_SCRIPT`, `OWNED_OBJECT`, `MATLAB_STRUCT`,  
`EPHEM_HEADER`, `NO_COMMENTS`, `GUI_EDITOR`, `OBJECT_EXPORT` }
  - enum `StateElementId` {  
`UNKNOWN_STATE` = -1, `CARTESIAN_STATE` = 3700, `EQUINOCTIAL_STATE`, `ORBIT_STATE_TRANS↵`  
`ITION_MATRIX`,  
`ORBIT_A_MATRIX`, `MASS_FLOW`, `PREDEFINED_STATE_MAX`, `USER_DEFINED_BEGIN` = 3800,  
`USER_DEFINED_END` = 3999 }
  - enum `ParameterType` {  
`INTEGER_TYPE`, `UNSIGNED_INT_TYPE`, `UNSIGNED_INTARRAY_TYPE`, `INTARRAY_TYPE`,  
`REAL_TYPE`, `REAL_ELEMENT_TYPE`, `STRING_TYPE`, `STRINGARRAY_TYPE`,  
`BOOLEAN_TYPE`, `BOOLEANARRAY_TYPE`, `RVECTOR_TYPE`, `RMATRIX_TYPE`,  
`TIME_TYPE`, `OBJECT_TYPE`, `OBJECTARRAY_TYPE`, `ON_OFF_TYPE`,  
`ENUMERATION_TYPE`, `FILENAME_TYPE`, `COLOR_TYPE`, `GMATTIME_TYPE`,  
`TypeCount`, `UNKNOWN_PARAMETER_TYPE` = -1, `PARAMETER_REMOVED` = -3 }
  - enum `MessageType` {  
`ERROR_` = 10, `WARNING_`, `INFO_`, `DEBUG_`,  
`GENERAL_` }
  - enum `RunState` {  
`IDLE` = 10000, `RUNNING`, `PAUSED`, `TARGETING`,  
`OPTIMIZING`, `ESTIMATING`, `SOLVING`, `SOLVEDPASS`,  
`WAITING` }
  - enum `WrapperDataType` {  
`NUMBER_WT`, `MATRIX_WT`, `STRING_WT`, `STRING_OBJECT_WT`,  
`OBJECT_PROPERTY_WT`, `VARIABLE_WT`, `ARRAY_WT`, `ARRAY_ELEMENT_WT`,  
`PARAMETER_WT`, `OBJECT_WT`, `BOOLEAN_WT`, `INTEGER_WT`,  
`ON_OFF_WT`, `UNKNOWN_WRAPPER_TYPE` = -2 }

## 7.2.1 Typedef Documentation

### 7.2.1.1 PLUGIN\_RESOURCE

```
typedef struct Gmat::PluginResource Gmat::PLUGIN_RESOURCE
```

## 7.2.2 Enumeration Type Documentation

### 7.2.2.1 MessageType

```
enum Gmat::MessageType
```

## Enumerator

|               |  |
|---------------|--|
| ERROR_        |  |
| WARNIN↔<br>G_ |  |
| INFO_         |  |
| DEBUG_        |  |
| GENERA↔<br>L_ |  |

## 7.2.2.2 ObjectType

enum [Gmat::ObjectType](#)

The list of object types

This list needs to be synchronized with the GmatBase::OBJECT\_TYPE\_STRING list found in base/Foundation/↔  
GmatBase.cpp

## Enumerator

|                  |  |
|------------------|--|
| SPACECRAFT       |  |
| FORMATION        |  |
| SPACEOBJECT      |  |
| GROUND_STATION   |  |
| BURN             |  |
| IMPULSIVE_BURN   |  |
| FINITE_BURN      |  |
| COMMAND          |  |
| PROPAGATOR       |  |
| ODE_MODEL        |  |
| PHYSICAL_MODEL   |  |
| TRANSIENT_FORCE  |  |
| INTERPOLATOR     |  |
| SOLAR_SYSTEM     |  |
| SPACE_POINT      |  |
| CELESTIAL_BODY   |  |
| CALCULATED_POINT |  |
| LIBRATION_POINT  |  |
| BARYCENTER       |  |
| ATMOSPHERE       |  |
| PARAMETER        |  |
| VARIABLE         |  |
| ARRAY            |  |
| STRING           |  |
| STOP_CONDITION   |  |
| SOLVER           |  |
| SUBSCRIBER       |  |
| REPORT_FILE      |  |

## Enumerator

|                       |                                                                  |
|-----------------------|------------------------------------------------------------------|
| XY_PLOT               |                                                                  |
| ORBIT_VIEW            |                                                                  |
| DYNAMIC_DATA_DISPLAY  |                                                                  |
| EPHEMERIS_FILE        |                                                                  |
| PROP_SETUP            |                                                                  |
| FUNCTION              |                                                                  |
| FUEL_TANK             |                                                                  |
| THRUSTER              |                                                                  |
| CHEMICAL_THRUSTER     |                                                                  |
| ELECTRIC_THRUSTER     |                                                                  |
| CHEMICAL_FUEL_TANK    |                                                                  |
| ELECTRIC_FUEL_TANK    |                                                                  |
| POWER_SYSTEM          |                                                                  |
| SOLAR_POWER_SYSTEM    |                                                                  |
| NUCLEAR_POWER_SYSTEM  |                                                                  |
| HARDWARE              |                                                                  |
| COORDINATE_SYSTEM     |                                                                  |
| AXIS_SYSTEM           |                                                                  |
| ATTITUDE              |                                                                  |
| MATH_NODE             |                                                                  |
| MATH_TREE             |                                                                  |
| BODY_FIXED_POINT      |                                                                  |
| EVENT                 |                                                                  |
| EVENT_LOCATOR         |                                                                  |
| DATAINTERFACE_SOURCE  |                                                                  |
| MEASUREMENT_MODEL     |                                                                  |
| ERROR_MODEL           |                                                                  |
| DATASTREAM            |                                                                  |
| DATA_FILE             |                                                                  |
| OBTYPE                |                                                                  |
| DATA_FILTER           |                                                                  |
| INTERFACE             |                                                                  |
| MEDIA_CORRECTION      |                                                                  |
| SENSOR                |                                                                  |
| RF_HARDWARE           |                                                                  |
| ANTENNA               |                                                                  |
| USER_DEFINED_OBJECT   |                                                                  |
| USER_OBJECT_ID_NEEDED |                                                                  |
| GENERIC_OBJECT        | <b>Todo</b> : DJC - Do we need this for backwards compatibility? |
| UNKNOWN_OBJECT        |                                                                  |

## 7.2.2.3 ParameterType

```
enum Gmat::ParameterType
```

The list of data types

This list needs to be synchronized with the GmatBase::PARAM\_TYPE\_STRING list found in base/Foundation/↔ GmatBase.cpp

#### Enumerator

|                        |  |
|------------------------|--|
| INTEGER_TYPE           |  |
| UNSIGNED_INT_TYPE      |  |
| UNSIGNED_INTARRAY_TYPE |  |
| INTARRAY_TYPE          |  |
| REAL_TYPE              |  |
| REAL_ELEMENT_TYPE      |  |
| STRING_TYPE            |  |
| STRINGARRAY_TYPE       |  |
| BOOLEAN_TYPE           |  |
| BOOLEANARRAY_TYPE      |  |
| RVECTOR_TYPE           |  |
| RMATRIX_TYPE           |  |
| TIME_TYPE              |  |
| OBJECT_TYPE            |  |
| OBJECTARRAY_TYPE       |  |
| ON_OFF_TYPE            |  |
| ENUMERATION_TYPE       |  |
| FILENAME_TYPE          |  |
| COLOR_TYPE             |  |
| GMATTIME_TYPE          |  |
| TypeCount              |  |
| UNKNOWN_PARAMETER_TYPE |  |
| PARAMETER_REMOVED      |  |

#### 7.2.2.4 RunState

enum [Gmat::RunState](#)

#### Enumerator

|            |  |
|------------|--|
| IDLE       |  |
| RUNNING    |  |
| PAUSED     |  |
| TARGETING  |  |
| OPTIMIZING |  |
| ESTIMATING |  |
| SOLVING    |  |
| SOLVEDPASS |  |
| WAITING    |  |



## 7.2.2.5 StateElementId

```
enum Gmat::StateElementId
```

## Enumerator

|                               |  |
|-------------------------------|--|
| UNKNOWN_STATE                 |  |
| CARTESIAN_STATE               |  |
| EQUINOCTIAL_STATE             |  |
| ORBIT_STATE_TRANSITION_MATRIX |  |
| ORBIT_A_MATRIX                |  |
| MASS_FLOW                     |  |
| PREDEFINED_STATE_MAX          |  |
| USER_DEFINED_BEGIN            |  |
| USER_DEFINED_END              |  |

## 7.2.2.6 WrapperDataType

```
enum Gmat::WrapperDataType
```

## Enumerator

|                      |  |
|----------------------|--|
| NUMBER_WT            |  |
| MATRIX_WT            |  |
| STRING_WT            |  |
| STRING_OBJECT_WT     |  |
| OBJECT_PROPERTY_WT   |  |
| VARIABLE_WT          |  |
| ARRAY_WT             |  |
| ARRAY_ELEMENT_WT     |  |
| PARAMETER_WT         |  |
| OBJECT_WT            |  |
| BOOLEAN_WT           |  |
| INTEGER_WT           |  |
| ON_OFF_WT            |  |
| UNKNOWN_WRAPPER_TYPE |  |

## 7.2.2.7 WriteMode

```
enum Gmat::WriteMode
```

## Enumerator

|             |  |
|-------------|--|
| SCRIPTING   |  |
| SHOW_SCRIPT |  |

**Enumerator**

|               |  |
|---------------|--|
| OWNED_OBJECT  |  |
| MATLAB_STRUCT |  |
| EPHEM_HEADER  |  |
| NO_COMMENTS   |  |
| GUI_EDITOR    |  |
| OBJECT_EXPORT |  |

## 7.3 GmatAttitudeConstants Namespace Reference

**Variables**

- const [Real](#) QUAT\_MIN\_MAG = 1.0e-10
- const [Real](#) EULER\_ANGLE\_TOLERANCE = 1.0E-10
- const [Real](#) DCM\_ORTHONORMALITY\_TOLERANCE = 1.0e-14

### 7.3.1 Variable Documentation

#### 7.3.1.1 DCM\_ORTHONORMALITY\_TOLERANCE

```
const Real GmatAttitudeConstants::DCM_ORTHONORMALITY_TOLERANCE = 1.0e-14
```

#### 7.3.1.2 EULER\_ANGLE\_TOLERANCE

```
const Real GmatAttitudeConstants::EULER_ANGLE_TOLERANCE = 1.0E-10
```

#### 7.3.1.3 QUAT\_MIN\_MAG

```
const Real GmatAttitudeConstants::QUAT_MIN_MAG = 1.0e-10
```

## 7.4 GmatEop Namespace Reference

**Enumerations**

- enum [EopFileType](#) { [EOP\\_C04](#), [FINALS](#) }

### 7.4.1 Enumeration Type Documentation

#### 7.4.1.1 EopFileType

```
enum GmatEop::EopFileType
```

## Enumerator

|         |  |
|---------|--|
| EOP_C04 |  |
| FINALS  |  |

## 7.5 GmatFile Namespace Reference

### 7.5.1 Detailed Description

Provides constants for file types.

## 7.6 GmatFileUtil Namespace Reference

### Functions

- `std::string GMATUTIL_API GetPathSeparator ()`
- `std::string GMATUTIL_API ConvertToOsFileName (const std::string &fileName)`
- `std::string GMATUTIL_API GetCurrentWorkingDirectory ()`
- `bool GMATUTIL_API SetCurrentWorkingDirectory (const std::string &newDir)`
- `std::string GMATUTIL_API GetApplicationPath ()`
- `std::string GMATUTIL_API GetTemporaryDirectory ()`
- `std::string GMATUTIL_API ParseFirstPathName (const std::string &fullPath, bool appendSep=true)`
- `std::string GMATUTIL_API ParsePathName (const char *fullPath, bool appendSep=true)`
- `std::string GMATUTIL_API ParsePathName (const std::string &fullPath, bool appendSep=true)`
- `std::string GMATUTIL_API ParseFileName (const char *fullPath, bool removeExt=false)`
- `std::string GMATUTIL_API ParseFileName (const std::string &fullPath, bool removeExt=false)`
- `std::string GMATUTIL_API ParseFileExtension (const char *fullPath, bool prependDot=false)`
- `std::string GMATUTIL_API ParseFileExtension (const std::string &fullPath, bool prependDot=false)`
- `std::string GMATUTIL_API GetInvalidFileNameMessage (Integer option=1)`
- `bool GMATUTIL_API IsOsWindows ()`
- `bool GMATUTIL_API IsPathRelative (const char *fullPath)`
- `bool GMATUTIL_API IsPathRelative (const std::string &fullPath)`
- `bool GMATUTIL_API IsPathAbsolute (const std::string &fullPath)`
- `bool GMATUTIL_API HasNoPath (const std::string &fullPath)`
- `bool GMATUTIL_API IsValidFileName (const std::string &fname, bool isBlankOk=true)`
- `bool GMATUTIL_API IsSameFileName (const char *fname1, const char *fname2)`
- `bool GMATUTIL_API IsSameFileName (const std::string &fname1, const std::string &fname2)`
- `bool GMATUTIL_API DoesDirectoryExist (const char *dirPath, bool isBlankOk=true)`
- `bool GMATUTIL_API DoesDirectoryExist (const std::string &dirPath, bool isBlankOk=true)`
- `bool GMATUTIL_API DoesFileExist (const char *filename)`
- `bool GMATUTIL_API DoesFileExist (const std::string &filename)`
- `bool GMATUTIL_API GetLine (std::istream *inStream, std::string &line)`
- `bool GMATUTIL_API IsAppInstalled (const std::string &appName, std::string &appLoc)`
- `std::string GetGmatPath ()`
- `WrapperTypeArray GMATUTIL_API GetFunctionOutputTypes (std::istream *is, const StringArray &inputs, const StringArray &outputs, std::string &errMsg, IntegerArray &outputRows, IntegerArray &outputCols)`
- `StringArray GMATUTIL_API GetFileListFromDirectory (const std::string &dirName, bool addPath=false)`
- `StringArray GMATUTIL_API GetTextLines (const std::string &fileName)`

- bool [PrepareCompare](#) ([Integer](#) numDirsToCompare, const std::string &basefilename, const std::string &filename1, const std::string &filename2, const std::string &filename3, std::ifstream &baseln, std::ifstream &in1, std::ifstream &in2, std::ifstream &in3)
- bool [GMATUTIL\\_API CompareLines](#) (const std::string &line1, const std::string &line2, [Real](#) &diff, [Real](#) tol=COMPARE\_TOLERANCE)
- [StringArray](#) [GMATUTIL\\_API](#) & [CompareTextLines](#) ([Integer](#) numDirsToCompare, const char \*basefilename, const char \*filename1, const char \*filename2, const char \*filename3, int &file1DiffCount, int &file2DiffCount, int &file3DiffCount, bool skipBlankLines=false)
- [StringArray](#) [GMATUTIL\\_API](#) & [CompareTextLines](#) ([Integer](#) numDirsToCompare, const std::string &basefilename, const std::string &filename1, const std::string &filename2, const std::string &filename3, int &file1DiffCount, int &file2DiffCount, int &file3DiffCount, bool skipBlankLines=false)
- [StringArray](#) [GMATUTIL\\_API](#) & [CompareNumericLines](#) ([Integer](#) numDirsToCompare, const char \*basefilename, const char \*filename1, const char \*filename2, const char \*filename3, int &file1DiffCount, int &file2DiffCount, int &file3DiffCount, [Real](#) tol=COMPARE\_TOLERANCE)
- [StringArray](#) [GMATUTIL\\_API](#) & [CompareNumericLines](#) ([Integer](#) numDirsToCompare, const std::string &basefilename, const std::string &filename1, const std::string &filename2, const std::string &filename3, int &file1DiffCount, int &file2DiffCount, int &file3DiffCount, [Real](#) tol=COMPARE\_TOLERANCE)
- [StringArray](#) [GMATUTIL\\_API](#) & [CompareNumericColumns](#) ([Integer](#) numDirsToCompare, const char \*basefilename, const char \*filename1, const char \*filename2, const char \*filename3, [Real](#) tol=COMPARE\_TOLERANCE)
- [StringArray](#) [GMATUTIL\\_API](#) & [CompareNumericColumns](#) ([Integer](#) numDirsToCompare, const std::string &basefilename, const std::string &filename1, const std::string &filename2, const std::string &filename3, [Real](#) tol=COMPARE\_TOLERANCE)
- bool [GMATUTIL\\_API SkipHeaderLines](#) (std::ifstream &in, [RealArray](#) &realArray, const std::string &filename)
- bool [GMATUTIL\\_API IsAsciiFile](#) (std::ifstream &file, const std::string &filename)
- bool [GMATUTIL\\_API GetRealColumns](#) (const std::string &line, [RealArray](#) &cols)

## Variables

- const [Integer](#) [BUFFER\\_SIZE](#) = 4096

## 7.6.1 Detailed Description

This file provides methods to query file information and methods to compare two output files. The compare summary is written to the log file.

## 7.6.2 Function Documentation

### 7.6.2.1 CompareLines()

```
bool GmatFileUtil::CompareLines (
 const std::string & line1,
 const std::string & line2,
 Real & diff,
 Real tol = COMPARE_TOLERANCE)
```

Compares numeric values in lines. It skips strings embedded in the lines.

#### Returns

true if all items are numerically same within tolerance false if number of items between two lines are different or if all items are not numerically same within tolerance

## 7.6.2.2 CompareNumericColumns() [1/2]

```
StringArray & GmatFileUtil::CompareNumericColumns (
 Integer numDirsToCompare,
 const char * basefilename,
 const char * filename1,
 const char * filename2,
 const char * filename3,
 Real tol = COMPARE_TOLERANCE)
```

Numerically compares base file with up to three other files. It will compare the smallest number of columns among compare files.

## 7.6.2.3 CompareNumericColumns() [2/2]

```
StringArray & GmatFileUtil::CompareNumericColumns (
 Integer numDirsToCompare,
 const std::string & basefilename,
 const std::string & filename1,
 const std::string & filename2,
 const std::string & filename3,
 Real tol = COMPARE_TOLERANCE)
```

Numerically compares base file with up to three other files. It will compare the smallest number of columns among compare files.

## 7.6.2.4 CompareNumericLines() [1/2]

```
StringArray & GmatFileUtil::CompareNumericLines (
 Integer numDirsToCompare,
 const char * basefilename,
 const char * filename1,
 const char * filename2,
 const char * filename3,
 int & file1DiffCount,
 int & file2DiffCount,
 int & file3DiffCount,
 Real tol = COMPARE_TOLERANCE)
```

Compares files line by line numerically using tolerance. String embedded in a text line or blank line is ignored and continued with next item in the line.

## 7.6.2.5 CompareNumericLines() [2/2]

```
StringArray & GmatFileUtil::CompareNumericLines (
 Integer numDirsToCompare,
 const std::string & basefilename,
 const std::string & filename1,
 const std::string & filename2,
 const std::string & filename3,
 int & file1DiffCount,
 int & file2DiffCount,
 int & file3DiffCount,
 Real tol = COMPARE_TOLERANCE)
```

Compares files line by line numerically using tolerance. String embedded in a text line or blank line is ignored and continued with next item in the line.

#### 7.6.2.6 CompareTextLines() [1/2]

```
StringArray & GmatFileUtil::CompareTextLines (
 Integer numDirsToCompare,
 const char * basefilename,
 const char * filename1,
 const char * filename2,
 const char * filename3,
 int & file1DiffCount,
 int & file2DiffCount,
 int & file3DiffCount,
 bool skipBlankLines = false)
```

Compares ascii files line by line with the same file name in different directory.

#### 7.6.2.7 CompareTextLines() [2/2]

```
StringArray & GmatFileUtil::CompareTextLines (
 Integer numDirsToCompare,
 const std::string & basefilename,
 const std::string & filename1,
 const std::string & filename2,
 const std::string & filename3,
 int & file1DiffCount,
 int & file2DiffCount,
 int & file3DiffCount,
 bool skipBlankLines = false)
```

Compares ascii files line by line with the same file name in different directory.

#### 7.6.2.8 ConvertToOsFileName()

```
std::string GmatFileUtil::ConvertToOsFileName (
 const std::string & fileName)
```

#### 7.6.2.9 DoesDirectoryExist() [1/2]

```
bool GmatFileUtil::DoesDirectoryExist (
 const char * dirPath,
 bool isBlankOk = true)
```

#### 7.6.2.10 DoesDirectoryExist() [2/2]

```
bool GmatFileUtil::DoesDirectoryExist (
 const std::string & dirPath,
 bool isBlankOk = true)
```

**7.6.2.11 DoesFileExist()** [1/2]

```
bool GmatFileUtil::DoesFileExist (
 const char * filename)
```

**7.6.2.12 DoesFileExist()** [2/2]

```
bool GmatFileUtil::DoesFileExist (
 const std::string & filename)
```

**7.6.2.13 GetApplicationPath()**

```
std::string GmatFileUtil::GetApplicationPath ()
```

**7.6.2.14 GetCurrentWorkingDirectory()**

```
std::string GmatFileUtil::GetCurrentWorkingDirectory ()
```

**7.6.2.15 GetFileListFromDirectory()**

```
StringArray GmatFileUtil::GetFileListFromDirectory (
 const std::string & dirName,
 bool addPath = false)
```

**7.6.2.16 GetFunctionOutputTypes()**

```
WrapperTypeArray GmatFileUtil::GetFunctionOutputTypes (
 std::istream * is,
 const StringArray & inputs,
 const StringArray & outputs,
 std::string & errMsg,
 IntegerArray & outputRows,
 IntegerArray & outputCols)
```

#### 7.6.2.17 GetGmatPath()

```
std::string GmatFileUtil::GetGmatPath ()
```

Accessor method to find the location of the GMAT application

This method is currently only coded for Windows, to address path issues found in GMAT R2014a (see GMT-2688 and email reporting issues around May 9 2014)

##### Returns

The path to the GMAT application on Windows machines; an empty string for the others

#### 7.6.2.18 GetInvalidFileNameMessage()

```
std::string GmatFileUtil::GetInvalidFileNameMessage (
 Integer option = 1)
```

Returns invalid file name message.

#### 7.6.2.19 GetLine()

```
bool GmatFileUtil::GetLine (
 std::istream * inStream,
 std::string & line)
```

#### 7.6.2.20 GetPathSeparator()

```
std::string GmatFileUtil::GetPathSeparator ()
```

##### Returns

path separator; "/" or "\\" depends on the platform

#### 7.6.2.21 GetRealColumns()

```
bool GmatFileUtil::GetRealColumns (
 const std::string & line,
 RealArray & cols)
```



#### 7.6.2.22 GetTemporaryDirectory()

```
std::string GmatFileUtil::GetTemporaryDirectory ()
```

#### 7.6.2.23 GetTextLines()

```
StringArray GmatFileUtil::GetTextLines (
 const std::string & fileName)
```

#### 7.6.2.24 HasNoPath()

```
bool GmatFileUtil::HasNoPath (
 const std::string & fullPath)
```

#### 7.6.2.25 IsAppInstalled()

```
bool GmatFileUtil::IsAppInstalled (
 const std::string & appName,
 std::string & appLoc)
```

Asks system if requested application is installed

##### Parameters

|                |                                         |
|----------------|-----------------------------------------|
| <i>appName</i> | Name of the application, such as MATLAB |
|----------------|-----------------------------------------|

##### Returns

true requested application is installed on the system

##### Note

GMAT currently checks for only MATLAB installation

#### 7.6.2.26 IsAsciiFile()

```
bool GmatFileUtil::IsAsciiFile (
 std::ifstream & file,
 const std::string & filename)
```

Checks if file is an ascii file.

#### 7.6.2.27 IsOsWindows()

```
bool GmatFileUtil::IsOsWindows ()
```

##### Returns

true if platform OS is Windows, false if empty buffer from getenv() or other platform

#### 7.6.2.28 IsPathAbsolute()

```
bool GmatFileUtil::IsPathAbsolute (
 const std::string & fullPath)
```

##### Returns

true if given path is absolute path, false otherwise

#### 7.6.2.29 IsPathRelative() [1/2]

```
bool GmatFileUtil::IsPathRelative (
 const char * fullPath)
```

Checks if given path name has relative path.

##### Returns

true if input path has relative path, false otherwise

#### 7.6.2.30 IsPathRelative() [2/2]

```
bool GmatFileUtil::IsPathRelative (
 const std::string & fullPath)
```

Checks if given path name has relative path.

##### Returns

true if input path has relative path, false otherwise

**7.6.2.31 IsSameFileName()** [1/2]

```
bool GmatFileUtil::IsSameFileName (
 const char * fname1,
 const char * fname2)
```

**7.6.2.32 IsSameFileName()** [2/2]

```
bool GmatFileUtil::IsSameFileName (
 const std::string & fname1,
 const std::string & fname2)
```

**7.6.2.33 IsValidFileName()**

```
bool GmatFileUtil::IsValidFileName (
 const std::string & fname,
 bool isBlankOk = true)
```

**7.6.2.34 ParseFileExtension()** [1/2]

```
std::string GmatFileUtil::ParseFileExtension (
 const char * fullPath,
 bool prependDot = false)
```

**7.6.2.35 ParseFileExtension()** [2/2]

```
std::string GmatFileUtil::ParseFileExtension (
 const std::string & fullPath,
 bool prependDot = false)
```

**7.6.2.36 ParseFileName()** [1/2]

```
std::string GmatFileUtil::ParseFileName (
 const char * fullPath,
 bool removeExt = false)
```

#### 7.6.2.37 ParseFileName() [2/2]

```
std::string GmatFileUtil::ParseFileName (
 const std::string & fullPath,
 bool removeExt = false)
```

#### 7.6.2.38 ParseFirstPathName()

```
std::string GmatFileUtil::ParseFirstPathName (
 const std::string & fullPath,
 bool appendSep = true)
```

#### 7.6.2.39 ParsePathName() [1/2]

```
std::string GmatFileUtil::ParsePathName (
 const char * fullPath,
 bool appendSep = true)
```

#### 7.6.2.40 ParsePathName() [2/2]

```
std::string GmatFileUtil::ParsePathName (
 const std::string & fullPath,
 bool appendSep = true)
```

#### 7.6.2.41 PrepareCompare()

```
bool GmatFileUtil::PrepareCompare (
 Integer numDirsToCompare,
 const std::string & basefilename,
 const std::string & filename1,
 const std::string & filename2,
 const std::string & filename3,
 std::ifstream & baseIn,
 std::ifstream & in1,
 std::ifstream & in2,
 std::ifstream & in3)
```

Opens files for comparison. If the same file extension not found for in1, in2, and in3, it will try open with extension .truth.

#### Returns

returns true if all files open successfully.

#### 7.6.2.42 SetCurrentWorkingDirectory()

```
bool GmatFileUtil::SetCurrentWorkingDirectory (
 const std::string & newDir)
```

#### 7.6.2.43 SkipHeaderLines()

```
bool GmatFileUtil::SkipHeaderLines (
 std::ifstream & in,
 RealArray & realArray,
 const std::string & filename)
```

### 7.6.3 Variable Documentation

#### 7.6.3.1 BUFFER\_SIZE

```
const Integer GmatFileUtil::BUFFER_SIZE = 4096
```

## 7.7 GmatIntegerConstants Namespace Reference

### Variables

- const Integer INTEGER\_UNDEFINED = -987654321

#### 7.7.1 Variable Documentation

##### 7.7.1.1 INTEGER\_UNDEFINED

```
const Integer GmatIntegerConstants::INTEGER_UNDEFINED = -987654321
```

## 7.8 GmatMathConstants Namespace Reference

### Enumerations

- enum SIGN { PLUS = 1, MINUS = -1 }

#### 7.8.1 Enumeration Type Documentation

##### 7.8.1.1 SIGN

```
enum GmatMathConstants::SIGN
```

## Enumerator

|       |  |
|-------|--|
| PLUS  |  |
| MINUS |  |

## 7.9 GmatMathUtil Namespace Reference

### Functions

- [Integer GMATUTIL\\_API Abs](#) ([Integer](#) theNumber)
- [Real GMATUTIL\\_API Abs](#) ([Real](#) theNumber)
- [Real GMATUTIL\\_API NearestInt](#) ([Real](#) theNumber)
- [Real GMATUTIL\\_API Round](#) ([Real](#) theNumber)
- [Real GMATUTIL\\_API Floor](#) ([Real](#) theNumber)
- [Real GMATUTIL\\_API Fix](#) ([Real](#) theNumber)
- [Real GMATUTIL\\_API Ceiling](#) ([Real](#) theNumber)
- [Real GMATUTIL\\_API Mod](#) ([Real](#) left, [Real](#) right)
- [Real GMATUTIL\\_API Rem](#) ([Real](#) left, [Real](#) right)
- [void GMATUTIL\\_API Quotient](#) ([Real](#) top, [Real](#) bottom, [Integer](#) &result)
- [void GMATUTIL\\_API Quotient](#) ([Real](#) top, [Real](#) bottom, [Real](#) &result)
- [Real GMATUTIL\\_API Min](#) ([Real](#) left, [Real](#) right)
- [Real GMATUTIL\\_API Max](#) ([Real](#) left, [Real](#) right)
- [bool GMATUTIL\\_API IsPositive](#) ([Real](#) theNumber)
- [bool GMATUTIL\\_API IsNegative](#) ([Real](#) theNumber)
- [bool GMATUTIL\\_API IsNonNegative](#) ([Real](#) theNumber)
- [bool GMATUTIL\\_API IsZero](#) ([Real](#) theNumber, [Real](#) accuracy=[GmatRealConstants::REAL\\_EPSILON](#))
- [bool GMATUTIL\\_API IsEqual](#) ([Real](#) left, [Real](#) right, [Real](#) accuracy=[GmatRealConstants::REAL\\_EPSILON](#))
- [bool GMATUTIL\\_API IsEqual](#) ([GmatTime](#) left, [GmatTime](#) right, [Real](#) accuracy=[GmatRealConstants::REAL\\_EPSILON](#))
- [Integer GMATUTIL\\_API SignOf](#) ([Real](#) theNumber)
- [bool GMATUTIL\\_API IsOdd](#) ([Integer](#) theNumber)
- [bool GMATUTIL\\_API IsEven](#) ([Integer](#) theNumber)
- [Real GMATUTIL\\_API Rad](#) ([Real](#) angleInDeg, [bool](#) modBy2Pi=false)
- [Real GMATUTIL\\_API Deg](#) ([Real](#) angleInRad, [bool](#) modBy360=false)
- [Real GMATUTIL\\_API DegToRad](#) ([Real](#) deg, [bool](#) modBy2Pi=false)
- [Real GMATUTIL\\_API RadToDeg](#) ([Real](#) rad, [bool](#) modBy360=false)
- [Real GMATUTIL\\_API ArcsecToDeg](#) ([Real](#) asec, [bool](#) modBy360=false)
- [Real GMATUTIL\\_API ArcsecToRad](#) ([Real](#) asec, [bool](#) modBy2Pi=false)
- [Real GMATUTIL\\_API Sin](#) ([Real](#) angleInRad, [Real](#) cycleInRad=[GmatMathConstants::TWO\\_PI](#))
- [Real GMATUTIL\\_API SinXOverX](#) ([Real](#) angleInRad, [Real](#) cycleInRad=[GmatMathConstants::TWO\\_PI](#))
- [Real GMATUTIL\\_API Cos](#) ([Real](#) angleInRad, [Real](#) cycleInRad=[GmatMathConstants::TWO\\_PI](#))
- [Real GMATUTIL\\_API Tan](#) ([Real](#) angleInRad, [Real](#) cycleInRad=[GmatMathConstants::TWO\\_PI](#))
- [Real GMATUTIL\\_API Cosh](#) ([Real](#) angleInRad, [Real](#) cycleInRad=[GmatMathConstants::TWO\\_PI](#))
- [Real GMATUTIL\\_API Sinh](#) ([Real](#) angleInRad, [Real](#) cycleInRad=[GmatMathConstants::TWO\\_PI](#))
- [Real GMATUTIL\\_API Tanh](#) ([Real](#) angleInRad, [Real](#) cycleInRad=[GmatMathConstants::TWO\\_PI](#))
- [Real GMATUTIL\\_API ASin](#) ([Real](#) x, [Real](#) tol=[GmatRealConstants::REAL\\_TOL](#), [Real](#) cycleInRad=[GmatMathConstants::TWO\\_PI](#))
- [Real GMATUTIL\\_API ACos](#) ([Real](#) x, [Real](#) tol=[GmatRealConstants::REAL\\_TOL](#), [Real](#) cycleInRad=[GmatMathConstants::TWO\\_PI](#))
- [Real GMATUTIL\\_API ATan](#) ([Real](#) y, [Real](#) x=1.0, [Real](#) cycleInRad=[GmatMathConstants::TWO\\_PI](#))
- [Real GMATUTIL\\_API ATan2](#) ([Real](#) y, [Real](#) x=1.0, [Real](#) cycleInRad=[GmatMathConstants::TWO\\_PI](#))

- [Real GMATUTIL\\_API ASinh \(Real x, Real cycleInRad=GmatMathConstants::TWO\\_PI\)](#)
- [Real GMATUTIL\\_API ACosh \(Real x, Real cycleInRad=GmatMathConstants::TWO\\_PI\)](#)
- [Real GMATUTIL\\_API ATanh \(Real x, Real cycleInRad=GmatMathConstants::TWO\\_PI\)](#)
- [Real GMATUTIL\\_API Ln \(Real x\)](#)
- [Real GMATUTIL\\_API Log \(Real x\)](#)
- [Real GMATUTIL\\_API Log10 \(Real x\)](#)
- [Real GMATUTIL\\_API Log \(Real x, Real base\)](#)
- [Real GMATUTIL\\_API Log \(Real x, Integer base\)](#)
- [Real GMATUTIL\\_API Rand \(Real lowerBound=0.0, Real upperBound=1.0\)](#)
- [Real GMATUTIL\\_API Randn \(Real mean=0.0, Real stdev=1.0\)](#)
- [void GMATUTIL\\_API SetSeed \(UnsignedInt seed\)](#)
- [Real GMATUTIL\\_API Cbrt \(Real x\)](#)
- [Real GMATUTIL\\_API Sqrt \(Real x\)](#)
- [Real GMATUTIL\\_API Exp \(Real x\)](#)
- [Real GMATUTIL\\_API Exp10 \(Real x\)](#)
- [Real GMATUTIL\\_API Pow \(Real x, Real y\)](#)
- [Real GMATUTIL\\_API Pow \(Real x, Integer y\)](#)
- [bool GMATUTIL\\_API IsNaN \(Real x\)](#)
- [bool GMATUTIL\\_API IsInf \(Real x\)](#)

## 7.9.1 Function Documentation

### 7.9.1.1 Abs() [1/2]

```
Integer GmatMathUtil::Abs (
 Integer theNumber)
```

### 7.9.1.2 Abs() [2/2]

```
Real GmatMathUtil::Abs (
 Real theNumber)
```

### 7.9.1.3 ACos()

```
Real GmatMathUtil::ACos (
 Real x,
 Real tol = GmatRealConstants::REAL_TOL,
 Real cycleInRad = GmatMathConstants::TWO_PI)
```

#### 7.9.1.4 ACosh()

```
Real GmatMathUtil::ACosh (
 Real x,
 Real cycleInRad = GmatMathConstants::TWO_PI)
```

#### 7.9.1.5 ArcsecToDeg()

```
Real GmatMathUtil::ArcsecToDeg (
 Real asec,
 bool modBy360 = false)
```

#### 7.9.1.6 ArcsecToRad()

```
Real GmatMathUtil::ArcsecToRad (
 Real asec,
 bool modBy2Pi = false)
```

#### 7.9.1.7 ASin()

```
Real GmatMathUtil::ASin (
 Real x,
 Real tol = GmatRealConstants::REAL_TOL,
 Real cycleInRad = GmatMathConstants::TWO_PI)
```

#### 7.9.1.8 ASinh()

```
Real GmatMathUtil::ASinh (
 Real x,
 Real cycleInRad = GmatMathConstants::TWO_PI)
```

#### 7.9.1.9 ATan()

```
Real GmatMathUtil::ATan (
 Real y,
 Real x = 1.0,
 Real cycleInRad = GmatMathConstants::TWO_PI)
```



#### 7.9.1.10 ATan2()

```
Real GmatMathUtil::ATan2 (
 Real y,
 Real x = 1.0,
 Real cycleInRad = GmatMathConstants::TWO_PI)
```

#### 7.9.1.11 ATanh()

```
Real GmatMathUtil::ATanh (
 Real x,
 Real cycleInRad = GmatMathConstants::TWO_PI)
```

Computes inverse hyperbolic tangent of x.

#### 7.9.1.12 Cbrt()

```
Real GmatMathUtil::Cbrt (
 Real x)
```

#### Note

$30\_031 = 2*3*5*7*11*13+1$   $67\_831 = 2*3*5*7*17*19+1$ ;  
 $211 = 2*3*5*7+1$   $2311 = 2*3*5*7*11 + 1$   $30031 = 2*3*5*7*11*13+1$   $67831 = 2*3*5*7*17*19+1$   
 $2037032760 = (30031 * 67831) -1$

#### 7.9.1.13 Ceiling()

```
Real GmatMathUtil::Ceiling (
 Real theNumber)
```

#### 7.9.1.14 Cos()

```
Real GmatMathUtil::Cos (
 Real angleInRad,
 Real cycleInRad = GmatMathConstants::TWO_PI)
```

#### 7.9.1.15 Cosh()

```
Real GmatMathUtil::Cosh (
 Real angleInRad,
 Real cycleInRad = GmatMathConstants::TWO_PI)
```

#### 7.9.1.16 Deg()

```
Real GmatMathUtil::Deg (
 Real angleInRad,
 bool modBy360 = false)
```

#### 7.9.1.17 DegToRad()

```
Real GmatMathUtil::DegToRad (
 Real deg,
 bool modBy2Pi = false)
```

#### 7.9.1.18 Exp()

```
Real GmatMathUtil::Exp (
 Real x)
```

##### Returns

e raised to the x power

#### 7.9.1.19 Exp10()

```
Real GmatMathUtil::Exp10 (
 Real x)
```

##### Returns

10 raised to the x power

#### 7.9.1.20 Fix()

```
Real GmatMathUtil::Fix (
 Real theNumber)
```

#### 7.9.1.21 Floor()

```
Real GmatMathUtil::Floor (
 Real theNumber)
```

#### 7.9.1.22 IsEqual() [1/2]

```
bool GmatMathUtil::IsEqual (
 Real left,
 Real right,
 Real accuracy = GmatRealConstants::REAL_EPSILON)
```

#### 7.9.1.23 IsEqual() [2/2]

```
bool GmatMathUtil::IsEqual (
 GmatTime left,
 GmatTime right,
 Real accuracy = GmatRealConstants::REAL_EPSILON)
```

#### 7.9.1.24 IsEven()

```
bool GmatMathUtil::IsEven (
 Integer theNumber)
```

#### 7.9.1.25 IsInf()

```
bool GmatMathUtil::IsInf (
 Real x) [inline]
```

Tests if input value is a infinite number.

#### 7.9.1.26 IsNaN()

```
bool GmatMathUtil::IsNaN (
 Real x) [inline]
```

Tests if input value is not a number.

#### 7.9.1.27 IsNegative()

```
bool GmatMathUtil::IsNegative (
 Real theNumber)
```

#### 7.9.1.28 IsNonNegative()

```
bool GmatMathUtil::IsNonNegative (
 Real theNumber)
```

#### 7.9.1.29 IsOdd()

```
bool GmatMathUtil::IsOdd (
 Integer theNumber)
```

#### 7.9.1.30 IsPositive()

```
bool GmatMathUtil::IsPositive (
 Real theNumber)
```

#### 7.9.1.31 IsZero()

```
bool GmatMathUtil::IsZero (
 Real theNumber,
 Real accuracy = GmatRealConstants::REAL_EPSILON)
```

### 7.9.1.32 Ln()

```
Real GmatMathUtil::Ln (
 Real x)
```

#### Returns

Natural logarithm for x

### 7.9.1.33 Log() [1/3]

```
Real GmatMathUtil::Log (
 Real x)
```

#### Returns

Natural logarithm for x

### 7.9.1.34 Log() [2/3]

```
Real GmatMathUtil::Log (
 Real x,
 Real base)
```

#### Returns

Base <base> Logarithm of x

### 7.9.1.35 Log() [3/3]

```
Real GmatMathUtil::Log (
 Real x,
 Integer base)
```

#### Returns

Base <base> Logarithm of x

#### 7.9.1.36 Log10()

```
Real GmatMathUtil::Log10 (
 Real x)
```

##### Returns

Base 10 logarithm for x

#### 7.9.1.37 Max()

```
Real GmatMathUtil::Max (
 Real left,
 Real right)
```

#### 7.9.1.38 Min()

```
Real GmatMathUtil::Min (
 Real left,
 Real right)
```

#### 7.9.1.39 Mod()

```
Real GmatMathUtil::Mod (
 Real left,
 Real right)
```

#### 7.9.1.40 NearestInt()

```
Real GmatMathUtil::NearestInt (
 Real theNumber)
```

#### 7.9.1.41 Pow() <sup>[1/2]</sup>

```
Real GmatMathUtil::Pow (
 Real x,
 Real y)
```

##### Returns

x raised to the y power

#### 7.9.1.42 Pow() [2/2]

```
Real GmatMathUtil::Pow (
 Real x,
 Integer y)
```

##### Returns

x raised to the y power

#### 7.9.1.43 Quotient() [1/2]

```
void GmatMathUtil::Quotient (
 Real top,
 Real bottom,
 Integer & result)
```

#### 7.9.1.44 Quotient() [2/2]

```
void GmatMathUtil::Quotient (
 Real top,
 Real bottom,
 Real & result)
```

#### 7.9.1.45 Rad()

```
Real GmatMathUtil::Rad (
 Real angleInDeg,
 bool modBy2Pi = false)
```

#### 7.9.1.46 RadToDeg()

```
Real GmatMathUtil::RadToDeg (
 Real rad,
 bool modBy360 = false)
```

#### 7.9.1.47 Rand()

```
Real GmatMathUtil::Rand (
 Real lowerBound = 0.0,
 Real upperBound = 1.0)
```

Returns uniformly distributed pseudorandom number

#### 7.9.1.48 Randn()

```
Real GmatMathUtil::Randn (
 Real mean = 0.0,
 Real stdev = 1.0)
```

Returns normally distributed pseudorandom numbers. Actually the normal distribution is the sub form of Gaussian distribution. Gaussian distribution have 2 parameters, mean and variance. When there is zero mean and unit variance the Gaussian distribution becomes normal other wise it is pronounced as Gaussian.

#### 7.9.1.49 Rem()

```
Real GmatMathUtil::Rem (
 Real left,
 Real right)
```

#### 7.9.1.50 Round()

```
Real GmatMathUtil::Round (
 Real theNumber)
```

#### 7.9.1.51 SetSeed()

```
void GmatMathUtil::SetSeed (
 UnsignedInt seed)
```

#### 7.9.1.52 SignOf()

```
Integer GmatMathUtil::SignOf (
 Real theNumber)
```



### 7.9.1.53 Sin()

```
Real GmatMathUtil::Sin (
 Real angleInRad,
 Real cycleInRad = GmatMathConstants::TWO_PI)
```

### 7.9.1.54 Sinh()

```
Real GmatMathUtil::Sinh (
 Real angleInRad,
 Real cycleInRad = GmatMathConstants::TWO_PI)
```

### 7.9.1.55 SinXOverX()

```
Real GmatMathUtil::SinXOverX (
 Real angleInRad,
 Real cycleInRad = GmatMathConstants::TWO_PI)
```

### 7.9.1.56 Sqrt()

```
Real GmatMathUtil::Sqrt (
 Real x)
```

### 7.9.1.57 Tan()

```
Real GmatMathUtil::Tan (
 Real angleInRad,
 Real cycleInRad = GmatMathConstants::TWO_PI)
```

### 7.9.1.58 Tanh()

```
Real GmatMathUtil::Tanh (
 Real angleInRad,
 Real cycleInRad = GmatMathConstants::TWO_PI)
```

## 7.10 GmatOrbitConstants Namespace Reference

## 7.11 GmatPhysicalConstants Namespace Reference

### Variables

- const [Real](#) [SPEED\\_OF\\_LIGHT\\_VACUUM](#) = 299792458.0
- const [Real](#) [c](#) = 299792458.0
- const [Real](#) [UNIVERSAL\\_GRAVITATIONAL\\_CONSTANT](#) = 6.673e-20
- const [Real](#) [ASTRONOMICAL\\_UNIT](#) = 1.49597870e8
- const [Real](#) [ABSOLUTE\\_ZERO\\_K](#) = 0.0
- const [Real](#) [ABSOLUTE\\_ZERO\\_C](#) = -273.15

### 7.11.1 Variable Documentation

#### 7.11.1.1 ABSOLUTE\_ZERO\_C

```
const Real GmatPhysicalConstants::ABSOLUTE_ZERO_C = -273.15
```

#### 7.11.1.2 ABSOLUTE\_ZERO\_K

```
const Real GmatPhysicalConstants::ABSOLUTE_ZERO_K = 0.0
```

#### 7.11.1.3 ASTRONOMICAL\_UNIT

```
const Real GmatPhysicalConstants::ASTRONOMICAL_UNIT = 1.49597870e8
```

#### 7.11.1.4 c

```
const Real GmatPhysicalConstants::c = 299792458.0
```

#### 7.11.1.5 SPEED\_OF\_LIGHT\_VACUUM

```
const Real GmatPhysicalConstants::SPEED_OF_LIGHT_VACUUM = 299792458.0
```

## 7.11.1.6 UNIVERSAL\_GRAVITATIONAL\_CONSTANT

```
const Real GmatPhysicalConstants::UNIVERSAL_GRAVITATIONAL_CONSTANT = 6.673e-20
```

## 7.12 GmatRealConstants Namespace Reference

### Variables

- const Real REAL\_TOL = 1.0e-15
- const Real REAL\_UNDEFINED = -987654321.0123e-45
- const Real REAL\_UNDEFINED\_LARGE = -9876543210.1234
- const Integer INTEGER\_UNDEFINED = -987654321
- const Integer INTEGER\_MAX = std::numeric\_limits<Integer>::max()
- const Integer SHORT\_REAL\_RADIX = 2
- const Real REAL\_EPSILON = 2.2204460492503131e-16
- const Integer REAL\_DIG = 53
- const Integer REAL\_MIN\_EXP = -1021
- const Real REAL\_MIN = 2.2250738585072014e-308
- const Integer REAL\_MIN\_10\_EXP = -307
- const Integer REAL\_MAX\_EXP = 1024
- const Real REAL\_MAX = 1.7976931348623157e+308
- const Integer REAL\_MAX\_10\_EXP = 308

### 7.12.1 Detailed Description

Provides declarations for commonly used physical/computation/time/conversion constants.

### 7.12.2 Variable Documentation

#### 7.12.2.1 INTEGER\_MAX

```
const Integer GmatRealConstants::INTEGER_MAX = std::numeric_limits<Integer>::max()
```

#### 7.12.2.2 INTEGER\_UNDEFINED

```
const Integer GmatRealConstants::INTEGER_UNDEFINED = -987654321
```

#### 7.12.2.3 REAL\_DIG

```
const Integer GmatRealConstants::REAL_DIG = 53
```

#### 7.12.2.4 REAL\_EPSILON

```
const Real GmatRealConstants::REAL_EPSILON = 2.2204460492503131e-16
```

#### 7.12.2.5 REAL\_MAX

```
const Real GmatRealConstants::REAL_MAX = 1.7976931348623157e+308
```

#### 7.12.2.6 REAL\_MAX\_10\_EXP

```
const Integer GmatRealConstants::REAL_MAX_10_EXP = 308
```

#### 7.12.2.7 REAL\_MAX\_EXP

```
const Integer GmatRealConstants::REAL_MAX_EXP = 1024
```

#### 7.12.2.8 REAL\_MIN

```
const Real GmatRealConstants::REAL_MIN = 2.2250738585072014e-308
```

#### 7.12.2.9 REAL\_MIN\_10\_EXP

```
const Integer GmatRealConstants::REAL_MIN_10_EXP = -307
```

#### 7.12.2.10 REAL\_MIN\_EXP

```
const Integer GmatRealConstants::REAL_MIN_EXP = -1021
```

## 7.12.2.11 REAL\_TOL

```
const Real GmatRealConstants::REAL_TOL = 1.0e-15
```

## 7.12.2.12 REAL\_UNDEFINED

```
const Real GmatRealConstants::REAL_UNDEFINED = -987654321.0123e-45
```

## 7.12.2.13 REAL\_UNDEFINED\_LARGE

```
const Real GmatRealConstants::REAL_UNDEFINED_LARGE = -9876543210.1234
```

## 7.12.2.14 SHORT\_REAL\_RADIX

```
const Integer GmatRealConstants::SHORT_REAL_RADIX = 2
```

## 7.13 GmatRealUtil Namespace Reference

## Classes

- struct [RaCodec](#)
- struct [RaDec](#)

## Functions

- [GMATUTIL\\_API RaCodec CartesianToRaCodec](#) (const [Rvector3](#) &r)
- [GMATUTIL\\_API RaDec CartesianToRaDec](#) (const [Rvector3](#) &r)
- [GMATUTIL\\_API Rvector3 RaCodecToCartesian](#) (const [RaCodec](#) &r)
- [GMATUTIL\\_API RaDec RaCodecToRaDec](#) (const [RaCodec](#) &r)
- [GMATUTIL\\_API Rvector3 RaDecToCartesian](#) (const [RaDec](#) &r)
- [GMATUTIL\\_API RaCodec RaDecToRaCodec](#) (const [RaDec](#) &r)
- [GMATUTIL\\_API Real Min](#) (const [Rvector](#) &numbers)
- [GMATUTIL\\_API Real Max](#) (const [Rvector](#) &numbers)
- [GMATUTIL\\_API std::istream & operator>>](#) (std::istream &input, [Rvector](#) &a)
- [GMATUTIL\\_API std::ostream & operator<<](#) (std::ostream &output, const [Rvector](#) &a)
- [GMATUTIL\\_API std::istream & operator>>](#) (std::istream &input, [Rmatrix](#) &a)
- [GMATUTIL\\_API std::ostream & operator<<](#) (std::ostream &output, const [Rmatrix](#) &a)
- [GMATUTIL\\_API std::string RealToString](#) (const [Real](#) &rval, bool useCurrentFormat=true, bool scientific=false, bool showPoint=false, [Integer](#) precision=[GmatGlobal::DATA\\_PRECISION](#), [Integer](#) width=[GmatGlobal::DATA\\_WIDTH](#))
- [GMATUTIL\\_API std::string ToString](#) (const [Real](#) &rval, bool useCurrentFormat=true, bool scientific=false, bool showPoint=false, [Integer](#) precision=[GmatGlobal::DATA\\_PRECISION](#), [Integer](#) width=[GmatGlobal::DATA\\_WIDTH](#))
- [GMATUTIL\\_API std::string ToString](#) (const [Integer](#) &ival, bool useCurrentFormat=true, [Integer](#) width=[GmatGlobal::INTEGER\\_WIDTH](#))

### 7.13.1 Detailed Description

Declares Linear Algebra conversion, Linear I/O, and Linear Math operations.

### 7.13.2 Function Documentation

#### 7.13.2.1 CartesianToRaCodec()

```
GmatRealUtil::RaCodec GmatRealUtil::CartesianToRaCodec (
 const Rvector3 & r)
```

##### Exceptions

|               |                                                                                                                 |
|---------------|-----------------------------------------------------------------------------------------------------------------|
| <i>throws</i> | <a href="#">RealUtilitiesExceptions::ArgumentError</a> if all three Cartesian coordinates to be converted are 0 |
|---------------|-----------------------------------------------------------------------------------------------------------------|

#### 7.13.2.2 CartesianToRaDec()

```
GmatRealUtil::RaDec GmatRealUtil::CartesianToRaDec (
 const Rvector3 & r)
```

##### Exceptions

|               |                                                                                                                 |
|---------------|-----------------------------------------------------------------------------------------------------------------|
| <i>throws</i> | <a href="#">RealUtilitiesExceptions::ArgumentError</a> if all three Cartesian coordinates to be converted are 0 |
|---------------|-----------------------------------------------------------------------------------------------------------------|

#### 7.13.2.3 Max()

```
Real GmatRealUtil::Max (
 const Rvector & numbers)
```

#### 7.13.2.4 Min()

```
Real GmatRealUtil::Min (
 const Rvector & numbers)
```

7.13.2.5 `operator<<()` [1/2]

```
std::ostream & GmatRealUtil::operator<< (
 std::ostream & output,
 const Rvector & a)
```

Formats [Rvector](#) value using global format and sends to output stream. Once global format is set, it remains the same format until it is reset by `global->SetActualFormat()`.

## Parameters

|               |                                      |
|---------------|--------------------------------------|
| <i>output</i> | Output stream                        |
| <i>a</i>      | <a href="#">Rvector</a> to write out |

return Output stream

7.13.2.6 `operator<<()` [2/2]

```
std::ostream & GmatRealUtil::operator<< (
 std::ostream & output,
 const Rmatrix & a)
```

Formats [Rmatrix](#) value using global format and sends to output stream. Once global format is set, it remains the same format until it is reset by `global->SetActualFormat()`.

## Parameters

|               |                                      |
|---------------|--------------------------------------|
| <i>output</i> | Output stream                        |
| <i>a</i>      | <a href="#">Rmatrix</a> to write out |

return Output stream

7.13.2.7 `operator>>()` [1/2]

```
std::istream & GmatRealUtil::operator>> (
 std::istream & input,
 Rvector & a)
```

7.13.2.8 `operator>>()` [2/2]

```
std::istream & GmatRealUtil::operator>> (
 std::istream & input,
 Rmatrix & a)
```

#### 7.13.2.9 RaCodecToCartesian()

```
Rvector3 GmatRealUtil::RaCodecToCartesian (
 const RaCodec & r)
```

#### 7.13.2.10 RaCodecToRaDec()

```
GmatRealUtil::RaDec GmatRealUtil::RaCodecToRaDec (
 const RaCodec & r)
```

#### 7.13.2.11 RaDecToCartesian()

```
Rvector3 GmatRealUtil::RaDecToCartesian (
 const RaDec & r)
```

#### 7.13.2.12 RaDecToRaCodec()

```
GmatRealUtil::RaCodec GmatRealUtil::RaDecToRaCodec (
 const RaDec & r)
```

#### 7.13.2.13 RealToString()

```
std::string GmatRealUtil::RealToString (
 const Real & rval,
 bool useCurrentFormat = true,
 bool scientific = false,
 bool showPoint = false,
 Integer precision = GmatGlobal::DATA_PRECISION,
 Integer width = GmatGlobal::DATA_WIDTH)
```

#### 7.13.2.14 ToString() [1/2]

```
std::string GmatRealUtil::ToString (
 const Real & rval,
 bool useCurrentFormat = true,
 bool scientific = false,
 bool showPoint = false,
 Integer precision = GmatGlobal::DATA_PRECISION,
 Integer width = GmatGlobal::DATA_WIDTH)
```



## 7.13.2.15 ToString() [2/2]

```
std::string GmatRealUtil::ToString (
 const Integer & ival,
 bool useCurrentFormat = true,
 Integer width = GmatGlobal::INTEGER_WIDTH)
```

## 7.14 GmatSolarSystemDefaults Namespace Reference

## Enumerations

- enum `DefaultPlanets` {  
`MERCURY` = 0, `VENUS`, `EARTH`, `MARS`,  
`JUPITER`, `SATURN`, `URANUS`, `NEPTUNE`,  
`PLUTO`, `NumberOfDefaultPlanets` }  
*Default planet data ----- planets -----.*
- enum `DefaultMoons` { `LUNA` = 0, `NumberOfDefaultMoons` }  
*Default planet data ----- moons -----.*

## Variables

- const std::string `SOLAR_SYSTEM_BARYCENTER_NAME` = "SolarSystemBarycenter"  
*default names for each of the possible celestial bodies in the solar system*
- const std::string `SUN_NAME` = "Sun"
- const std::string `MERCURY_NAME` = "Mercury"
- const std::string `VENUS_NAME` = "Venus"
- const std::string `EARTH_NAME` = "Earth"
- const std::string `MOON_NAME` = "Luna"
- const std::string `MARS_NAME` = "Mars"
- const std::string `PHOBOS_NAME` = "Phobos"
- const std::string `DEIMOS_NAME` = "Deimos"
- const std::string `JUPITER_NAME` = "Jupiter"
- const std::string `METIS_NAME` = "Metis"
- const std::string `ADRASTEIA_NAME` = "Adrastea"
- const std::string `AMALTHEA_NAME` = "Amalthea"
- const std::string `THEBE_NAME` = "Thebe"
- const std::string `IO_NAME` = "Io"
- const std::string `EUROPA_NAME` = "Europa"
- const std::string `GANYMEDE_NAME` = "Ganymede"
- const std::string `CALLISTO_NAME` = "Callisto"
- const std::string `SATURN_NAME` = "Saturn"
- const std::string `PAN_NAME` = "Pan"
- const std::string `ATLAS_NAME` = "Atlas"
- const std::string `PROMETHEUS_NAME` = "Prometheus"
- const std::string `PANDORA_NAME` = "Pandora"
- const std::string `EPIMETHEUS_NAME` = "Epimetheus"
- const std::string `JANUS_NAME` = "Janus"
- const std::string `MIMAS_NAME` = "Mimas"
- const std::string `ENCELADUS_NAME` = "Enceladus"
- const std::string `TETHYS_NAME` = "Tethys"
- const std::string `TELESTO_NAME` = "Telesto"

- const std::string CALYPSO\_NAME = "Calypso"
- const std::string DIONE\_NAME = "Dione"
- const std::string HELENE\_NAME = "Helene"
- const std::string RHEA\_NAME = "Rhea"
- const std::string TITAN\_NAME = "Titan"
- const std::string IAPETUS\_NAME = "Iapetus"
- const std::string PHOEBE\_NAME = "Phoebe"
- const std::string URANUS\_NAME = "Uranus"
- const std::string CORDELIA\_NAME = "Cordelia"
- const std::string OPHELIA\_NAME = "Ophelia"
- const std::string BIANCA\_NAME = "Bianca"
- const std::string CRESSIDA\_NAME = "Cressida"
- const std::string DESDEMONA\_NAME = "Desdemona"
- const std::string JULIET\_NAME = "Juliet"
- const std::string PORTIA\_NAME = "Portia"
- const std::string ROSALIND\_NAME = "Rosalind"
- const std::string BELINDA\_NAME = "Belinda"
- const std::string PUCK\_NAME = "Puck"
- const std::string MIRANDA\_NAME = "Miranda"
- const std::string ARIEL\_NAME = "Ariel"
- const std::string UMBRIEL\_NAME = "Umbriel"
- const std::string TITANIA\_NAME = "Titania"
- const std::string OBERON\_NAME = "Oberon"
- const std::string NEPTUNE\_NAME = "Neptune"
- const std::string NAIAD\_NAME = "Naiad"
- const std::string THALASSA\_NAME = "Thalassa"
- const std::string DESPINA\_NAME = "Despina"
- const std::string GALATEA\_NAME = "Galatea"
- const std::string LARISSA\_NAME = "Larissa"
- const std::string PROTEUS\_NAME = "Proteus"
- const std::string TRITON\_NAME = "Triton"
- const std::string PLUTO\_NAME = "Pluto"
- const std::string CHARON\_NAME = "Charon"
- const Integer SSB\_NAIF\_ID = 0

*Default barycenter data.*

- const Real SSB\_MU = 0.0
- const std::string PLANET\_NAMES [NumberOfDefaultPlanets]

*Default planet data.*

- const Real PLANET\_EQUATORIAL\_RADIUS [NumberOfDefaultPlanets]
- const Real PLANET\_FLATTENING [NumberOfDefaultPlanets]
- const Real PLANET\_MU [NumberOfDefaultPlanets]
- const Real PLANET\_TWO\_BODY\_EPOCH [NumberOfDefaultPlanets]
- const Rvector6 PLANET\_TWO\_BODY\_ELEMENTS [NumberOfDefaultPlanets]
- const Rvector6 PLANET\_ORIENTATION\_PARAMETERS [NumberOfDefaultPlanets]
- const Integer PLANET\_NAIF\_IDS [NumberOfDefaultPlanets]
- const std::string PLANET\_SPICE\_FRAME\_ID [NumberOfDefaultPlanets]
- const std::string MOON\_NAMES [NumberOfDefaultMoons]
- const std::string MOON\_CENTRAL\_BODIES [NumberOfDefaultMoons]
- const Real MOON\_EQUATORIAL\_RADIUS [NumberOfDefaultMoons]
- const Real MOON\_FLATTENING [NumberOfDefaultMoons]
- const Real MOON\_MU [NumberOfDefaultMoons]
- const Real MOON\_TWO\_BODY\_EPOCH [NumberOfDefaultMoons]
- const Rvector6 MOON\_TWO\_BODY\_ELEMENTS [NumberOfDefaultMoons]
- const Rvector6 MOON\_ORIENTATION\_PARAMETERS [NumberOfDefaultMoons]

- const Integer MOON\_NAIF\_IDS [NumberOfDefaultMoons]
  - const std::string MOON\_SPICE\_FRAME\_ID [NumberOfDefaultMoons]
  - const Real STAR\_EQUATORIAL\_RADIUS = 695990.0000
- Default star data ----- the Sun -----.*
- const Real STAR\_FLATTENING = 0.0
  - const Real STAR\_MU = 132712440017.99
  - const Real STAR\_TWO\_BODY\_EPOCH = 21544.500370768266
  - const Rvector6 STAR\_TWO\_BODY\_ELEMENTS
  - const Rvector6 STAR\_ORIENTATION\_PARAMETERS
  - const Integer STAR\_NAIF\_IDS = 10
  - const std::string STAR\_SPICE\_FRAME\_ID = "IAU\_SUN"
  - const Real STAR\_RADIANT\_POWER = 1358.0
  - const Real STAR\_REFERENCE\_DISTANCE = GmatPhysicalConstants::ASTRONOMICAL\_UNIT
  - const Real STAR\_PHOTOSPHERE\_RADIUS = 695990000.0

### 7.14.1 Detailed Description

Provides commonly used default values.

### 7.14.2 Enumeration Type Documentation

#### 7.14.2.1 DefaultMoons

```
enum GmatSolarSystemDefaults::DefaultMoons
```

Default planet data ----- moons -----.

Enumerator

|                      |  |
|----------------------|--|
| LUNA                 |  |
| NumberOfDefaultMoons |  |

#### 7.14.2.2 DefaultPlanets

```
enum GmatSolarSystemDefaults::DefaultPlanets
```

Default planet data ----- planets -----.

Enumerator

|         |  |
|---------|--|
| MERCURY |  |
| VENUS   |  |
| EARTH   |  |

## Enumerator

|                        |  |
|------------------------|--|
| MARS                   |  |
| JUPITER                |  |
| SATURN                 |  |
| URANUS                 |  |
| NEPTUNE                |  |
| PLUTO                  |  |
| NumberOfDefaultPlanets |  |

### 7.14.3 Variable Documentation

#### 7.14.3.1 ADRASTEIA\_NAME

```
const std::string GmatSolarSystemDefaults::ADRASTEIA_NAME = "Adrastea"
```

#### 7.14.3.2 AMALTHEA\_NAME

```
const std::string GmatSolarSystemDefaults::AMALTHEA_NAME = "Amalthea"
```

#### 7.14.3.3 ARIEL\_NAME

```
const std::string GmatSolarSystemDefaults::ARIEL_NAME = "Ariel"
```

#### 7.14.3.4 ATLAS\_NAME

```
const std::string GmatSolarSystemDefaults::ATLAS_NAME = "Atlas"
```

#### 7.14.3.5 BELINDA\_NAME

```
const std::string GmatSolarSystemDefaults::BELINDA_NAME = "Belinda"
```

#### 7.14.3.6 BIANCA\_NAME

```
const std::string GmatSolarSystemDefaults::BIANCA_NAME = "Bianca"
```

#### 7.14.3.7 CALLISTO\_NAME

```
const std::string GmatSolarSystemDefaults::CALLISTO_NAME = "Callisto"
```

#### 7.14.3.8 CALYPSO\_NAME

```
const std::string GmatSolarSystemDefaults::CALYPSO_NAME = "Calypso"
```

#### 7.14.3.9 CHARON\_NAME

```
const std::string GmatSolarSystemDefaults::CHARON_NAME = "Charon"
```

#### 7.14.3.10 CORDELIA\_NAME

```
const std::string GmatSolarSystemDefaults::CORDELIA_NAME = "Cordelia"
```

#### 7.14.3.11 CRESSIDA\_NAME

```
const std::string GmatSolarSystemDefaults::CRESSIDA_NAME = "Cressida"
```

#### 7.14.3.12 DEIMOS\_NAME

```
const std::string GmatSolarSystemDefaults::DEIMOS_NAME = "Deimos"
```

#### 7.14.3.13 DESDEMONA\_NAME

```
const std::string GmatSolarSystemDefaults::DESDEMONA_NAME = "Desdemona"
```

#### 7.14.3.14 DESPINA\_NAME

```
const std::string GmatSolarSystemDefaults::DESPINA_NAME = "Despina"
```

#### 7.14.3.15 DIONE\_NAME

```
const std::string GmatSolarSystemDefaults::DIONE_NAME = "Dione"
```

#### 7.14.3.16 EARTH\_NAME

```
const std::string GmatSolarSystemDefaults::EARTH_NAME = "Earth"
```

#### 7.14.3.17 ENCELADUS\_NAME

```
const std::string GmatSolarSystemDefaults::ENCELADUS_NAME = "Enceladus"
```

#### 7.14.3.18 EPIMETHEUS\_NAME

```
const std::string GmatSolarSystemDefaults::EPIMETHEUS_NAME = "Epimetheus"
```

#### 7.14.3.19 EUROPA\_NAME

```
const std::string GmatSolarSystemDefaults::EUROPA_NAME = "Europa"
```

#### 7.14.3.20 GALATEA\_NAME

```
const std::string GmatSolarSystemDefaults::GALATEA_NAME = "Galatea"
```

#### 7.14.3.21 GANYMEDE\_NAME

```
const std::string GmatSolarSystemDefaults::GANYMEDE_NAME = "Ganymede"
```

#### 7.14.3.22 HELENE\_NAME

```
const std::string GmatSolarSystemDefaults::HELENE_NAME = "Helene"
```

#### 7.14.3.23 IAPETUS\_NAME

```
const std::string GmatSolarSystemDefaults::IAPETUS_NAME = "Iapetus"
```

#### 7.14.3.24 IO\_NAME

```
const std::string GmatSolarSystemDefaults::IO_NAME = "Io"
```

#### 7.14.3.25 JANUS\_NAME

```
const std::string GmatSolarSystemDefaults::JANUS_NAME = "Janus"
```

#### 7.14.3.26 JULIET\_NAME

```
const std::string GmatSolarSystemDefaults::JULIET_NAME = "Juliet"
```

#### 7.14.3.27 JUPITER\_NAME

```
const std::string GmatSolarSystemDefaults::JUPITER_NAME = "Jupiter"
```

#### 7.14.3.28 LARISSA\_NAME

```
const std::string GmatSolarSystemDefaults::LARISSA_NAME = "Larissa"
```

#### 7.14.3.29 MARS\_NAME

```
const std::string GmatSolarSystemDefaults::MARS_NAME = "Mars"
```

#### 7.14.3.30 MERCURY\_NAME

```
const std::string GmatSolarSystemDefaults::MERCURY_NAME = "Mercury"
```

#### 7.14.3.31 METIS\_NAME

```
const std::string GmatSolarSystemDefaults::METIS_NAME = "Metis"
```

#### 7.14.3.32 MIMAS\_NAME

```
const std::string GmatSolarSystemDefaults::MIMAS_NAME = "Mimas"
```

#### 7.14.3.33 MIRANDA\_NAME

```
const std::string GmatSolarSystemDefaults::MIRANDA_NAME = "Miranda"
```

#### 7.14.3.34 MOON\_CENTRAL\_BODIES

```
const std::string GmatSolarSystemDefaults::MOON_CENTRAL_BODIES[NumberOfDefaultMoons]
```

**Initial value:**

```
=
{
 EARTH_NAME,
}
```

#### 7.14.3.35 MOON\_EQUATORIAL\_RADIUS

```
const Real GmatSolarSystemDefaults::MOON_EQUATORIAL_RADIUS[NumberOfDefaultMoons]
```

**Initial value:**

```
=
{
 1738.2000,
}
```



## 7.14.3.36 MOON\_FLATTENING

```
const Real GmatSolarSystemDefaults::MOON_FLATTENING[NumberOfDefaultMoons]
```

**Initial value:**

```
=
{
 0.0,

}
```

## 7.14.3.37 MOON\_MU

```
const Real GmatSolarSystemDefaults::MOON_MU[NumberOfDefaultMoons]
```

**Initial value:**

```
=
{
 4902.8005821478,

}
```

## 7.14.3.38 MOON\_NAIF\_IDS

```
const Integer GmatSolarSystemDefaults::MOON_NAIF_IDS[NumberOfDefaultMoons]
```

**Initial value:**

```
=
{
 301,

}
```

## 7.14.3.39 MOON\_NAME

```
const std::string GmatSolarSystemDefaults::MOON_NAME = "Luna"
```

#### 7.14.3.40 MOON\_NAMES

```
const std::string GmatSolarSystemDefaults::MOON_NAMES[NumberOfDefaultMoons]
```

**Initial value:**

```
=
{
 MOON_NAME,

}
```

#### 7.14.3.41 MOON\_ORIENTATION\_PARAMETERS

```
const Rvector6 GmatSolarSystemDefaults::MOON_ORIENTATION_PARAMETERS[NumberOfDefaultMoons]
```

**Initial value:**

```
=
{
 Rvector6(0.0, 0.0, 0.0, 0.0, 0.0, 0.0),

}
```

#### 7.14.3.42 MOON\_SPICE\_FRAME\_ID

```
const std::string GmatSolarSystemDefaults::MOON_SPICE_FRAME_ID[NumberOfDefaultMoons]
```

**Initial value:**

```
=
{
 "MOON_PA",

}
```

#### 7.14.3.43 MOON\_TWO\_BODY\_ELEMENTS

```
const Rvector6 GmatSolarSystemDefaults::MOON_TWO_BODY_ELEMENTS[NumberOfDefaultMoons]
```

**Initial value:**

```
=
{
 Rvector6(
 385494.90434829952, 0.055908943292024992, 20.940245433093748,
 12.233244412716252, 68.004298803147648, 137.94325682926458),

}
```

#### 7.14.3.44 MOON\_TWO\_BODY\_EPOCH

```
const Real GmatSolarSystemDefaults::MOON_TWO_BODY_EPOCH[NumberOfDefaultMoons]
```

**Initial value:**

```
=
{
 21544.500370768266,
}
```

#### 7.14.3.45 NAIAD\_NAME

```
const std::string GmatSolarSystemDefaults::NAIAD_NAME = "Naiad"
```

#### 7.14.3.46 NEPTUNE\_NAME

```
const std::string GmatSolarSystemDefaults::NEPTUNE_NAME = "Neptune"
```

#### 7.14.3.47 OBERON\_NAME

```
const std::string GmatSolarSystemDefaults::OBERON_NAME = "Oberon"
```

#### 7.14.3.48 OPHELIA\_NAME

```
const std::string GmatSolarSystemDefaults::OPHELIA_NAME = "Ophelia"
```

#### 7.14.3.49 PAN\_NAME

```
const std::string GmatSolarSystemDefaults::PAN_NAME = "Pan"
```

#### 7.14.3.50 PANDORA\_NAME

```
const std::string GmatSolarSystemDefaults::PANDORA_NAME = "Pandora"
```

#### 7.14.3.51 PHOBOS\_NAME

```
const std::string GmatSolarSystemDefaults::PHOBOS_NAME = "Phobos"
```

#### 7.14.3.52 PHOEBE\_NAME

```
const std::string GmatSolarSystemDefaults::PHOEBE_NAME = "Phoebe"
```

#### 7.14.3.53 PLANET\_EQUATORIAL\_RADIUS

```
const Real GmatSolarSystemDefaults::PLANET_EQUATORIAL_RADIUS[NumberOfDefaultPlanets]
```

**Initial value:**

```
=
{
 2.439700000000000e+003,
 6.051900000000000e+003,
 6.3781363E3,
 3.397000000000000e+003,
 7.149200000000000e+004,
 6.026800000000000e+004,
 2.555900000000000e+004,
 2.526900000000000e+004,
 1162.0
}
```

#### 7.14.3.54 PLANET\_FLATTENING

```
const Real GmatSolarSystemDefaults::PLANET_FLATTENING[NumberOfDefaultPlanets]
```

**Initial value:**

```
=
{
 0.0,
 0.0,
 0.00335270,
 0.00647630,
 0.06487439,
 0.09796243,
 0.02292734,
 0.01856029,
 0.0
}
```

### 7.14.3.55 PLANET\_MU

```
const Real GmatSolarSystemDefaults::PLANET_MU[NumberOfDefaultPlanets]
```

**Initial value:**

```
=
{
 22032.080486418,
 324858.59882646,
 398600.4415,
 42828.314258067,
 126712767.85780,
 37940626.061137,
 5794549.0070719,
 6836534.0638793,
 981.60088770700
}
```

### 7.14.3.56 PLANET\_NAIF\_IDS

```
const Integer GmatSolarSystemDefaults::PLANET_NAIF_IDS[NumberOfDefaultPlanets]
```

**Initial value:**

```
=
{
 199,
 299,
 399,
 499,
 599,
 699,
 799,
 899,
 999,
}
```

### 7.14.3.57 PLANET\_NAMES

```
const std::string GmatSolarSystemDefaults::PLANET_NAMES[NumberOfDefaultPlanets]
```

**Initial value:**

```
=
{
 MERCURY_NAME,
 VENUS_NAME,
 EARTH_NAME,
 MARS_NAME,
 JUPITER_NAME,
 SATURN_NAME,
 URANUS_NAME,
 NEPTUNE_NAME,
 PLUTO_NAME,
}
```

Default planet data.

### 7.14.3.58 PLANET\_ORIENTATION\_PARAMETERS

```
const Rvector6 GmatSolarSystemDefaults::PLANET_ORIENTATION_PARAMETERS[NumberOfDefaultPlanets]
```

**Initial value:**

```
=
{
 Rvector6(281.01, -0.033, 61.45, -0.005, 329.548, 6.1385025),
 Rvector6(272.76, 0.0, 67.16, 0.0, 160.20, -1.4813688),
 Rvector6(0.0, -0.641, 90.00, -0.557, 190.147, 360.9856235),
 Rvector6(317.68143, -0.1061, 52.88650, -0.0609, 176.630, 350.89198226),
 Rvector6(268.05, -0.009, 64.49, 0.003, 284.95, 870.5366420),
 Rvector6(40.589, -0.036, 83.537, -0.004, 38.90, 810.7939024),

 Rvector6(257.311, 0.0, -15.175, 0.0, 203.81, -501.1600928),
 Rvector6(299.36, 0.70, 43.46, -0.51, 253.18, 536.3128492),
 Rvector6(313.02, 0.0, 9.09, 0.0, 236.77, -56.3623195)
}
```

### 7.14.3.59 PLANET\_SPICE\_FRAME\_ID

```
const std::string GmatSolarSystemDefaults::PLANET_SPICE_FRAME_ID[NumberOfDefaultPlanets]
```

**Initial value:**

```
=
{
 "IAU_MERCURY",
 "IAU_VENUS",
 "ITRF93",
 "IAU_MARS",
 "IAU_JUPITER",
 "IAU_SATURN",
 "IAU_URANUS",
 "IAU_NEPTUNE",
 "IAU_PLUTO",
}
```

### 7.14.3.60 PLANET\_TWO\_BODY\_ELEMENTS

```
const Rvector6 GmatSolarSystemDefaults::PLANET_TWO_BODY_ELEMENTS[NumberOfDefaultPlanets]
```

**Initial value:**

```
=
{
 Rvector6(57909212.938567216, 0.20562729774965544, 28.551674963293556,
 10.99100758149257, 67.548689584103984, 175.10396761800456),
 Rvector6(108208423.76486244, 0.0067572911404369688, 24.433051334216176,
 8.007373221205856, 124.55871063212626, 49.889845117140576),
 Rvector6(149653978.9783766, 0.01704556707314489, 23.439034090426388,
 0.00018646554487906264, 101.7416388084352, 358.12708491129),
 Rvector6(227939100.16983532, 0.093314935483163344, 24.677089965042784,
 3.3736838414054472, 333.01849018562076, 23.020633424007744),
 Rvector6(779362950.5867208, 0.049715759324379896, 23.235170252934984,
 3.253166212922, 12.959463238924978, 20.296667207322848),
 Rvector6(1433895241.1645338, 0.055944006117351672, 22.551333377462712,
 5.9451029086964872, 83.977808941927856, 316.23400767222348),
 Rvector6(2876804054.239868, 0.044369079419761096, 23.663364175915172,
 1.850441916938424, 168.86875273062818, 145.8502865552013),
 Rvector6(4503691751.2342816, 0.011211871260687014, 22.29780590076114,
 3.47555654789392, 33.957145210261132, 266.76236610390636),
 Rvector6(5909627293.567856, 0.24928777871911536, 23.4740184346088,
 43.998303104440304, 183.03164997859696, 25.513664216653164)
}
```

#### 7.14.3.61 PLANET\_TWO\_BODY\_EPOCH

```
const Real GmatSolarSystemDefaults::PLANET_TWO_BODY_EPOCH[NumberOfDefaultPlanets]
```

**Initial value:**

```
=
{
 21544.500370768266, 21544.500370768266, 21544.500370768266,
 21544.500370768266, 21544.500370768266, 21544.500370768266,
 21544.500370768266, 21544.500370768266, 21544.500370768266
}
```

#### 7.14.3.62 PLUTO\_NAME

```
const std::string GmatSolarSystemDefaults::PLUTO_NAME = "Pluto"
```

#### 7.14.3.63 PORTIA\_NAME

```
const std::string GmatSolarSystemDefaults::PORTIA_NAME = "Portia"
```

#### 7.14.3.64 PROMETHEUS\_NAME

```
const std::string GmatSolarSystemDefaults::PROMETHEUS_NAME = "Promethus"
```

#### 7.14.3.65 PROTEUS\_NAME

```
const std::string GmatSolarSystemDefaults::PROTEUS_NAME = "Proteus"
```

#### 7.14.3.66 PUCK\_NAME

```
const std::string GmatSolarSystemDefaults::PUCK_NAME = "Puck"
```

#### 7.14.3.67 RHEA\_NAME

```
const std::string GmatSolarSystemDefaults::RHEA_NAME = "Rhea"
```

#### 7.14.3.68 ROSALIND\_NAME

```
const std::string GmatSolarSystemDefaults::ROSALIND_NAME = "Rosalind"
```

#### 7.14.3.69 SATURN\_NAME

```
const std::string GmatSolarSystemDefaults::SATURN_NAME = "Saturn"
```

#### 7.14.3.70 SOLAR\_SYSTEM\_BARYCENTER\_NAME

```
const std::string GmatSolarSystemDefaults::SOLAR_SYSTEM_BARYCENTER_NAME = "SolarSystemBarycenter"
```

default names for each of the possible celestial bodies in the solar system

#### 7.14.3.71 SSB\_MU

```
const Real GmatSolarSystemDefaults::SSB_MU = 0.0
```

#### 7.14.3.72 SSB\_NAIF\_ID

```
const Integer GmatSolarSystemDefaults::SSB_NAIF_ID = 0
```

Default barycenter data.

#### 7.14.3.73 STAR\_EQUATORIAL\_RADIUS

```
const Real GmatSolarSystemDefaults::STAR_EQUATORIAL_RADIUS = 695990.0000
```

Default star data ----- the Sun -----.



#### 7.14.3.74 STAR\_FLATTENING

```
const Real GmatSolarSystemDefaults::STAR_FLATTENING = 0.0
```

#### 7.14.3.75 STAR\_MU

```
const Real GmatSolarSystemDefaults::STAR_MU = 132712440017.99
```

#### 7.14.3.76 STAR\_NAIF\_IDS

```
const Integer GmatSolarSystemDefaults::STAR_NAIF_IDS = 10
```

#### 7.14.3.77 STAR\_ORIENTATION\_PARAMETERS

```
const Rvector6 GmatSolarSystemDefaults::STAR_ORIENTATION_PARAMETERS
```

**Initial value:**

```
= Rvector6(
 286.13, 0.0, 63.87, 0.0, 84.10, 14.1844000)
```

#### 7.14.3.78 STAR\_PHOTOSPHERE\_RADIUS

```
const Real GmatSolarSystemDefaults::STAR_PHOTOSPHERE_RADIUS = 695990000.0
```

#### 7.14.3.79 STAR\_RADIANT\_POWER

```
const Real GmatSolarSystemDefaults::STAR_RADIANT_POWER = 1358.0
```

#### 7.14.3.80 STAR\_REFERENCE\_DISTANCE

```
const Real GmatSolarSystemDefaults::STAR_REFERENCE_DISTANCE = GmatPhysicalConstants::ASTRONOMICAL_UNIT
```

#### 7.14.3.81 STAR\_SPICE\_FRAME\_ID

```
const std::string GmatSolarSystemDefaults::STAR_SPICE_FRAME_ID = "IAU_SUN"
```

#### 7.14.3.82 STAR\_TWO\_BODY\_ELEMENTS

```
const Rvector6 GmatSolarSystemDefaults::STAR_TWO_BODY_ELEMENTS
```

**Initial value:**

```
= Rvector6(
 149653978.9783766, 0.01704556707314489, 23.439034090426388,
 0.00018646554487906264, 281.7416388084352, 358.12708491129)
```

#### 7.14.3.83 STAR\_TWO\_BODY\_EPOCH

```
const Real GmatSolarSystemDefaults::STAR_TWO_BODY_EPOCH = 21544.500370768266
```

#### 7.14.3.84 SUN\_NAME

```
const std::string GmatSolarSystemDefaults::SUN_NAME = "Sun"
```

#### 7.14.3.85 TELESTO\_NAME

```
const std::string GmatSolarSystemDefaults::TELESTO_NAME = "Telesto"
```

#### 7.14.3.86 TETHYS\_NAME

```
const std::string GmatSolarSystemDefaults::TETHYS_NAME = "Tethys"
```

#### 7.14.3.87 THALASSA\_NAME

```
const std::string GmatSolarSystemDefaults::THALASSA_NAME = "Thalassa"
```

#### 7.14.3.88 THEBE\_NAME

```
const std::string GmatSolarSystemDefaults::THEBE_NAME = "Thebe"
```

#### 7.14.3.89 TITAN\_NAME

```
const std::string GmatSolarSystemDefaults::TITAN_NAME = "Titan"
```

#### 7.14.3.90 TITANIA\_NAME

```
const std::string GmatSolarSystemDefaults::TITANIA_NAME = "Titania"
```

#### 7.14.3.91 TRITON\_NAME

```
const std::string GmatSolarSystemDefaults::TRITON_NAME = "Triton"
```

#### 7.14.3.92 UMBRIEL\_NAME

```
const std::string GmatSolarSystemDefaults::UMBRIEL_NAME = "Umbriel"
```

#### 7.14.3.93 URANUS\_NAME

```
const std::string GmatSolarSystemDefaults::URANUS_NAME = "Uranus"
```

#### 7.14.3.94 VENUS\_NAME

```
const std::string GmatSolarSystemDefaults::VENUS_NAME = "Venus"
```

## 7.15 GmatStringUtil Namespace Reference

### Enumerations

- enum [StripType](#) { [LEADING](#) = 1, [TRAILING](#) = 2, [BOTH](#) = 3 }
- enum [AlignmentType](#) { [LEFT](#), [RIGHT](#), [CENTER](#) }

## Functions

- [GMATUTIL\\_API](#) `std::string RemoveAll` (const `std::string` &str, char ch, [Integer](#) start=0)
- [GMATUTIL\\_API](#) `std::string RemoveAll` (const `std::string` &str, const `std::string` &removeStr, [Integer](#) start=0)
- [GMATUTIL\\_API](#) `std::string RemoveAllBlanks` (const `std::string` &str, bool ignoreSingleQuotes=false)
- [GMATUTIL\\_API](#) `std::string RemoveLastNumber` (const `std::string` &str, [Integer](#) &lastNumber)
- [GMATUTIL\\_API](#) `std::string RemoveLastString` (const `std::string` &str, const `std::string` &lastStr, bool remove↵  
All=false)
- [GMATUTIL\\_API](#) `std::string RemoveSpaceInBrackets` (const `std::string` &str, const `std::string` &bracketPair)
- [GMATUTIL\\_API](#) `std::string RemoveMultipleSpaces` (const `std::string` &str)
- [GMATUTIL\\_API](#) `std::string RemoveTrailingZeros` ([Real](#) val, const `std::string` &valStr, [Integer](#) iterCount=0)
- [GMATUTIL\\_API](#) `std::string RemoveScientificNotation` (const `std::string` &str)
- [GMATUTIL\\_API](#) `std::string RemoveMathSymbols` (const `std::string` &str, bool removeMathOperator=false)
- [GMATUTIL\\_API](#) `std::string PadWithBlanks` (const `std::string` &str, [Integer](#) toSize, [StripType](#) whichEnd=TR↵  
AILING)
- [GMATUTIL\\_API](#) `std::string BuildNumber` ([Real](#) value, bool useExp=false, [Integer](#) length=17)
- [GMATUTIL\\_API](#) `std::string Trim` (const `std::string` &str, [StripType](#) stype=BOTH, bool removeSemicolon=false, bool removeEol=false)
- [GMATUTIL\\_API](#) `std::string Strip` (const `std::string` &str, [StripType](#) stype=BOTH)
- [GMATUTIL\\_API](#) `std::string ToUpper` (const `std::string` &str, bool firstLetterOnly=false)
- [GMATUTIL\\_API](#) `std::string ToLower` (const `std::string` &str, bool firstLetterOnly=false)
- [GMATUTIL\\_API](#) `std::string Capitalize` (const `std::string` &str)
- [GMATUTIL\\_API](#) `std::string ReplaceFirst` (const `std::string` &str, const `std::string` &from, const `std::string` &to, `std::string::size_type` startIdx=0)
- [GMATUTIL\\_API](#) `std::string Replace` (const `std::string` &str, const `std::string` &from, const `std::string` &to, `std::string::size_type` startIdx=0)
- [GMATUTIL\\_API](#) `std::string ReplaceName` (const `std::string` &str, const `std::string` &from, const `std::string` &to)
- [GMATUTIL\\_API](#) `std::string ReplaceNumber` (const `std::string` &str, const `std::string` &from, const `std::string` &to)
- [GMATUTIL\\_API](#) `std::string ReplaceChainedUnaryOperators` (const `std::string` &str)
- [GMATUTIL\\_API](#) `std::string RealToString` (const [Real](#) &val, bool useCurrentFormat=true, bool scientific=false, bool showPoint=true, [Integer](#) precision=GmatGlobal::DATA\_PRECISION, [Integer](#) width=GmatGlobal::DAT↵  
A\_WIDTH)
- [GMATUTIL\\_API](#) `std::string RealToString` (const [Real](#) &val, [Integer](#) precision, bool showPoint=false, [Integer](#) width=1)
- [GMATUTIL\\_API](#) `std::string ToString` (const bool &val)
- [GMATUTIL\\_API](#) `std::string ToString` (const [Real](#) &val, [Integer](#) precision, bool showPoint=false, [Integer](#) width=1)
- [GMATUTIL\\_API](#) `std::string ToString` (const [Integer](#) &val, [Integer](#) width)
- [GMATUTIL\\_API](#) `std::string ToString` (const [Real](#) &val, bool useCurrentFormat=true, bool scientific=false, bool showPoint=true, [Integer](#) precision=GmatGlobal::DATA\_PRECISION, [Integer](#) width=GmatGlobal::DATA\_W↵  
IDTH)
- [GMATUTIL\\_API](#) `std::string ToString` (const [Integer](#) &val, bool useCurrentFormat=true, [Integer](#) width=Gmat↵  
Global::INTEGER\_WIDTH)
- [GMATUTIL\\_API](#) `std::string ToStringNoZeros` (const [Real](#) &val)
- [GMATUTIL\\_API](#) `std::string ToOrdinal` ([Integer](#) i, bool textOnly=false)
- [GMATUTIL\\_API](#) `std::string RemoveExtraParen` (const `std::string` &str, bool ignoreComma=false, bool ignoreSingleQuotes=false)
- [GMATUTIL\\_API](#) `std::string RemoveOuterParen` (const `std::string` &str)
- [GMATUTIL\\_API](#) `std::string RemoveOuterString` (const `std::string` &str, const `std::string` &start, const `std::string` &end)
- [GMATUTIL\\_API](#) `std::string RemoveEnclosingString` (const `std::string` &str, const `std::string` &enStr)
- [GMATUTIL\\_API](#) `std::string RemoveInlineComment` (const `std::string` &str, const `std::string` &cmStr)
- [GMATUTIL\\_API](#) `std::string MakeCommentLines` (const `std::string` &str, bool breakAtCr=false)
- [GMATUTIL\\_API](#) `std::string ParseFunctionName` (const `std::string` &str, `std::string` &argStr)
- [GMATUTIL\\_API](#) `StringArray ParseFunctionCall` (const `std::string` &str)

- [GMATUTIL\\_API](#) std::string [AddEnclosingString](#) (const std::string &str, const std::string &enStr)
- [GMATUTIL\\_API](#) std::string [GetInvalidNameMessageFormat](#) ()
- [GMATUTIL\\_API](#) char [GetClosingBracket](#) (const char &openBracket)
- [GMATUTIL\\_API](#) [StringArray](#) [SeparateBrackets](#) (const std::string &chunk, const std::string &bracketPair, const std::string &delim, bool checkOuterBracket=true)
- [GMATUTIL\\_API](#) [StringArray](#) [SeparateBy](#) (const std::string &str, const std::string &delim, bool putBrackets← Together=false, bool insertDelim=false, bool insertComma=true)
- [GMATUTIL\\_API](#) [StringArray](#) [SeparateByComma](#) (const std::string &str, bool checkSingleQuote=true)
- [GMATUTIL\\_API](#) [StringArray](#) [SeparateDots](#) (const std::string &str)
- [GMATUTIL\\_API](#) [StringArray](#) [DecomposeBy](#) (const std::string &str, const std::string &delim)
- [GMATUTIL\\_API](#) bool [IsNumber](#) (const std::string &str)
- [GMATUTIL\\_API](#) bool [IsValidReal](#) (const std::string &str, [Real](#) &value, [Integer](#) &errorCode, bool trim← Parens=false, bool allowOverflow=true)
- [GMATUTIL\\_API](#) bool [ToReal](#) (const char \*str, [Real](#) \*value, bool trimParens=false, bool allowOverflow=true)
- [GMATUTIL\\_API](#) bool [ToReal](#) (const std::string &str, [Real](#) \*value, bool trimParens=false, bool allow← Overflow=true)
- [GMATUTIL\\_API](#) bool [ToReal](#) (const char \*str, [Real](#) &value, bool trimParens=false, bool allowOverflow=true)
- [GMATUTIL\\_API](#) bool [ToReal](#) (const std::string &str, [Real](#) &value, bool trimParens=false, bool allow← Overflow=true)
- [GMATUTIL\\_API](#) bool [ToInteger](#) (const std::string &str, [Integer](#) \*value, bool trimParens=false, bool allow← Overflow=true)
- [GMATUTIL\\_API](#) bool [ToInteger](#) (const char \*str, [Integer](#) &value, bool trimParens=false, bool allow← Overflow=true)
- [GMATUTIL\\_API](#) bool [ToInteger](#) (const std::string &str, [Integer](#) &value, bool trimParens=false, bool allow← Overflow=true)
- [GMATUTIL\\_API](#) bool [ToUnsignedInt](#) (const std::string &str, [UnsignedInt](#) \*value, bool trimParens=false, bool allowOverflow=true)
- [GMATUTIL\\_API](#) bool [ToUnsignedInt](#) (const std::string &str, [UnsignedInt](#) &value, bool trimParens=false, bool allowOverflow=true)
- [GMATUTIL\\_API](#) bool [ToBoolean](#) (const std::string &str, bool \*value, bool trimParens=false)
- [GMATUTIL\\_API](#) bool [ToBoolean](#) (const std::string &str, bool &value, bool trimParens=false)
- [GMATUTIL\\_API](#) bool [ToOnOff](#) (const std::string &str, std::string &value, bool trimParens=false)
- [GMATUTIL\\_API](#) [RealArray](#) [ToRealArray](#) (const std::string &str, bool allowOverflow=true, bool allow← Semicolon=false)
- [GMATUTIL\\_API](#) [IntegerArray](#) [ToIntegerArray](#) (const char \*str, bool allowOverflow=true)
- [GMATUTIL\\_API](#) [IntegerArray](#) [ToIntegerArray](#) (const std::string &str, bool allowOverflow=true)
- [GMATUTIL\\_API](#) [UnsignedIntArray](#) [ToUnsignedIntArray](#) (const std::string &str, bool allowOverflow=true)
- [GMATUTIL\\_API](#) [StringArray](#) [ToStringArray](#) (const std::string &str)
- [GMATUTIL\\_API](#) [BooleanArray](#) [ToBooleanArray](#) (const std::string &str)
- [GMATUTIL\\_API](#) void [ParseParameter](#) (const std::string &str, std::string &type, std::string &owner, std::string &dep)
- [GMATUTIL\\_API](#) void [GetArrayCommaIndex](#) (const std::string &str, [Integer](#) &comma, const std::string &bracketPair="()")
- [GMATUTIL\\_API](#) void [GetArrayIndexVar](#) (const std::string &str, std::string &rowStr, std::string &colStr, std← ::string &name, const std::string &bracketPair="()")
- [GMATUTIL\\_API](#) void [GetArrayIndex](#) (const std::string &str, [Integer](#) &row, [Integer](#) &col, std::string &name, const std::string &bracketPair="()")
- [GMATUTIL\\_API](#) void [GetArrayIndex](#) (const std::string &str, std::string &rowStr, std::string &colStr, [Integer](#) &row, [Integer](#) &col, std::string &name, const std::string &bracketPair="()")
- [GMATUTIL\\_API](#) std::string [GetArrayName](#) (const std::string &str, const std::string &bracketPair="()")
- [GMATUTIL\\_API](#) bool [IsOneElementArray](#) (const std::string &str)
- [GMATUTIL\\_API](#) bool [IsSimpleArrayElement](#) (const std::string &str)
- [GMATUTIL\\_API](#) void [FindFirstAndLast](#) (const std::string &str, char ch, [Integer](#) &first, [Integer](#) &last)
- [GMATUTIL\\_API](#) void [FindParenMatch](#) (const std::string &str, [Integer](#) &open, [Integer](#) &close, bool &isOuter← Paren)

- [GMATUTIL\\_API](#) void [FindMatchingParen](#) (const std::string &str, [Integer](#) &openParen, [Integer](#) &closeParen, bool &isOuterParen, [Integer](#) start=0)
- [GMATUTIL\\_API](#) void [FindMatchingBracket](#) (const std::string &str, [Integer](#) &openBracket, [Integer](#) &closeBracket, bool &isOuterBracket, const std::string &bracket, [Integer](#) start=0)
- [GMATUTIL\\_API](#) void [FindLastParenMatch](#) (const std::string &str, [Integer](#) &openParen, [Integer](#) &closeParen, [Integer](#) start=0)
- [GMATUTIL\\_API](#) bool [IsEnclosedWith](#) (const std::string &str, const std::string &enclosingStr)
- [GMATUTIL\\_API](#) bool [IsEnclosedWithExtraParen](#) (const std::string &str, bool checkOps=true, bool ignoreComma=false)
- [GMATUTIL\\_API](#) bool [IsEnclosedWithBraces](#) (const std::string &str)
- [GMATUTIL\\_API](#) bool [IsEnclosedWithBrackets](#) (const std::string &str)
- [GMATUTIL\\_API](#) bool [IsBracketBalanced](#) (const std::string &str, const std::string &bracketPair)
- [GMATUTIL\\_API](#) bool [IsParenBalanced](#) (const std::string &str)
- [GMATUTIL\\_API](#) bool [IsParenEmpty](#) (const std::string &str)
- [GMATUTIL\\_API](#) bool [AreAllBracketsBalanced](#) (const std::string &str, const std::string &allPairs)
- [GMATUTIL\\_API](#) bool [AreAllNamesValid](#) (const std::string &str, bool blankNamesOk=false)
- [GMATUTIL\\_API](#) bool [IsValidFunctionCall](#) (const std::string &str)
- [GMATUTIL\\_API](#) bool [IsOuterParen](#) (const std::string &str)
- [GMATUTIL\\_API](#) bool [IsCommaPartOfArray](#) (const std::string &str, [Integer](#) start=0)
- [GMATUTIL\\_API](#) bool [IsBracketPartOfArray](#) (const std::string &str, const std::string &bracketPairs, bool checkOnlyFirst)
- [GMATUTIL\\_API](#) bool [IsParenPartOfArray](#) (const std::string &str)
- [GMATUTIL\\_API](#) bool [IsThereEqualSign](#) (const std::string &str)
- [GMATUTIL\\_API](#) bool [IsThereMathSymbol](#) (const std::string &str)
- [GMATUTIL\\_API](#) bool [HasNoBrackets](#) (const std::string &str, bool parensForArraysAllowed=true)
- [GMATUTIL\\_API](#) bool [IsSingleItem](#) (const std::string &str)
- [GMATUTIL\\_API](#) bool [StartsWith](#) (const std::string &str, const std::string &value)
- [GMATUTIL\\_API](#) bool [EndsWith](#) (const std::string &str, const std::string &value)
- [GMATUTIL\\_API](#) bool [EndsWithPathSeparator](#) (const std::string &str)
- [GMATUTIL\\_API](#) bool [IsValidNumber](#) (const std::string &str, bool allowOverflow=true)
- [GMATUTIL\\_API](#) bool [IsValidName](#) (const char \*str, bool ignoreBracket=false, bool blankNamesOk=false)
- [GMATUTIL\\_API](#) bool [IsValidName](#) (const std::string &str, bool ignoreBracket=false, bool blankNamesOk=false)
- [GMATUTIL\\_API](#) bool [IsValidParameterName](#) (const std::string &str)
- [GMATUTIL\\_API](#) bool [IsLastNumberPartOfName](#) (const std::string &str)
- [GMATUTIL\\_API](#) bool [IsBlank](#) (const std::string &str, bool ignoreEol=false)
- [GMATUTIL\\_API](#) bool [HasMissingQuote](#) (const std::string &str, const std::string &quote, bool ignoreSpaceAfterQuote=true)
- [GMATUTIL\\_API](#) bool [IsStringInsideSymbols](#) (const std::string &str, const std::string &reqStr, const std::string &symbol, std::string::size\_type &reqStrPos)
- [GMATUTIL\\_API](#) bool [IsMathEquation](#) (const std::string &str, bool checkInvalidOpOnly=false, bool blankNamesOk=false)
- [GMATUTIL\\_API](#) bool [IsMathOperator](#) (const char &ch)
- [GMATUTIL\\_API](#) [Integer](#) [NumberOfOccurrences](#) (const std::string &str, const char c)
- [GMATUTIL\\_API](#) [Integer](#) [NumberOfScientificNotation](#) (const std::string &str)
- [GMATUTIL\\_API](#) [StringArray](#) [GetVarNames](#) (const std::string &str)
- [GMATUTIL\\_API](#) void [WriteStringArray](#) (const [StringArray](#) &strArray, const std::string &desc="", const std::string &prefix="")
- [GMATUTIL\\_API](#) bool [IsValidIdentity](#) (const std::string &str)
- [GMATUTIL\\_API](#) bool [IsValidExtendedIdentity](#) (const std::string &str)
- [GMATUTIL\\_API](#) bool [IsValidFileName](#) (const std::string &str)
- [GMATUTIL\\_API](#) bool [IsValidFullFileName](#) (const std::string &str, [Integer](#) &error)
- [GMATUTIL\\_API](#) [StringArray](#) [ParseName](#) (const std::string &extendedName)
- [GMATUTIL\\_API](#) std::string [GetAlignmentString](#) (const std::string &inputString, [UnsignedInt](#) len, [AlignmentType](#) alignment=LEFT)
- [GMATUTIL\\_API](#) std::wstring [StringToWideString](#) (const std::string &str)
- [GMATUTIL\\_API](#) std::string [WideStringToString](#) (const std::wstring &wstr)
- [GMATUTIL\\_API](#) std::string [WideStringToString](#) (const wchar\_t \*wchar)

### 7.15.1 Detailed Description

This file defines string utility functions.

### 7.15.2 Enumeration Type Documentation

#### 7.15.2.1 AlignmentType

```
enum GmatStringUtil::AlignmentType
```

Enumerator

|        |  |
|--------|--|
| LEFT   |  |
| RIGHT  |  |
| CENTER |  |

#### 7.15.2.2 StripType

```
enum GmatStringUtil::StripType
```

Enumerator

|          |  |
|----------|--|
| LEADING  |  |
| TRAILING |  |
| BOTH     |  |

### 7.15.3 Function Documentation

#### 7.15.3.1 AddEnclosingString()

```
std::string GmatStringUtil::AddEnclosingString (
 const std::string & str,
 const std::string & enStr)
```

Put string in enclosing string

Parameters

|              |                  |
|--------------|------------------|
| <i>str</i>   | Input string     |
| <i>enStr</i> | Enclosing string |

Generated by Doxygen

**Returns**

String with enclosing string

**7.15.3.2 AreAllBracketsBalanced()**

```
bool GmatStringUtil::AreAllBracketsBalanced (
 const std::string & str,
 const std::string & allPairs)
```

**7.15.3.3 AreAllNamesValid()**

```
bool GmatStringUtil::AreAllNamesValid (
 const std::string & str,
 bool blankNameIsOk = false)
```

Checks if names or number are valid in the math equation.

**7.15.3.4 BuildNumber()**

```
std::string GmatStringUtil::BuildNumber (
 Real value,
 bool useExp = false,
 Integer length = 17)
```

Builds a formatted string containing a Real, so the Real can be serialized to the display

**Parameters**

|               |                                      |
|---------------|--------------------------------------|
| <i>value</i>  | The Real that needs to be serialized |
| <i>useExp</i> | Use scientific notation              |
| <i>length</i> | The size of the desired string       |

**Returns**

The formatted string

**Note**

This was moved from GmatCommand



### 7.15.3.5 Capitalize()

```
std::string GmatStringUtil::Capitalize (
 const std::string & str)
```

Capitalize the first letter of the string

### 7.15.3.6 DecomposeBy()

```
StringArray GmatStringUtil::DecomposeBy (
 const std::string & str,
 const std::string & delim)
```

Returns the first token and the rest e.g) "cd c:/my test directory" returns if delimiter is " ". cd c:/my test directory

#### Parameters

|              |              |
|--------------|--------------|
| <i>str</i>   | Input string |
| <i>delim</i> | Delimiter    |

#### Returns

First token and the rest or empty string array if token not found

### 7.15.3.7 EndsWith()

```
bool GmatStringUtil::EndsWith (
 const std::string & str,
 const std::string & value)
```

### 7.15.3.8 EndsWithPathSeparator()

```
bool GmatStringUtil::EndsWithPathSeparator (
 const std::string & str)
```

### 7.15.3.9 FindFirstAndLast()

```
void GmatStringUtil::FindFirstAndLast (
 const std::string & str,
 char ch,
 Integer & first,
 Integer & last)
```

#### 7.15.3.10 FindLastParenMatch()

```
void GmatStringUtil::FindLastParenMatch (
 const std::string & str,
 Integer & openParen,
 Integer & closeParen,
 Integer start = 0)
```

#### 7.15.3.11 FindMatchingBracket()

```
void GmatStringUtil::FindMatchingBracket (
 const std::string & str,
 Integer & openBracket,
 Integer & closeBracket,
 bool & isOuterBracket,
 const std::string & bracket,
 Integer start = 0)
```

#### 7.15.3.12 FindMatchingParen()

```
void GmatStringUtil::FindMatchingParen (
 const std::string & str,
 Integer & openParen,
 Integer & closeParen,
 bool & isOuterParen,
 Integer start = 0)
```

#### 7.15.3.13 FindParenMatch()

```
void GmatStringUtil::FindParenMatch (
 const std::string & str,
 Integer & open,
 Integer & close,
 bool & isOuterParen)
```

#### 7.15.3.14 GetAlignmentString()

```
std::string GmatStringUtil::GetAlignmentString (
 const std::string inputString,
 UnsignedInt len,
 AlignmentType alignment = LEFT)
```

**7.15.3.15 GetArrayCommaIndex()**

```
void GmatStringUtil::GetArrayCommaIndex (
 const std::string & str,
 Integer & comma,
 const std::string & bracketPair = "()")
```

**7.15.3.16 GetArrayIndex()** [1/2]

```
void GmatStringUtil::GetArrayIndex (
 const std::string & str,
 Integer & row,
 Integer & col,
 std::string & name,
 const std::string & bracketPair = "()")
```

**7.15.3.17 GetArrayIndex()** [2/2]

```
void GmatStringUtil::GetArrayIndex (
 const std::string & str,
 std::string & rowStr,
 std::string & colStr,
 Integer & row,
 Integer & col,
 std::string & name,
 const std::string & bracketPair = "()")
```

**7.15.3.18 GetArrayIndexVar()**

```
void GmatStringUtil::GetArrayIndexVar (
 const std::string & str,
 std::string & rowStr,
 std::string & colStr,
 std::string & name,
 const std::string & bracketPair = "()")
```

**7.15.3.19 GetArrayName()**

```
std::string GmatStringUtil::GetArrayName (
 const std::string & str,
 const std::string & bracketPair = "()")
```

Extracts array name form array elememnt string, ie, MyArray(1,2) returns MyArray.

str Array element string to be parsed for array name

**Parameters**

|                    |                                                             |
|--------------------|-------------------------------------------------------------|
| <i>bracketPair</i> | bracket pair used in the input array, such as [], () ["()"] |
|--------------------|-------------------------------------------------------------|

**Returns**

Return array name if input string is array element form; input string if input string is not a form of array

**7.15.3.20 GetClosingBracket()**

```
char GmatStringUtil::GetClosingBracket (
 const char & openBracket)
```

**7.15.3.21 GetInvalidNameMessageFormat()**

```
std::string GmatStringUtil::GetInvalidNameMessageFormat ()
```

Returns invalid object name message.

**7.15.3.22 GetVarNames()**

```
StringArray GmatStringUtil::GetVarNames (
 const std::string & str)
```

**7.15.3.23 HasMissingQuote()**

```
bool GmatStringUtil::HasMissingQuote (
 const std::string & str,
 const std::string & quote,
 bool ignoreSpaceAfterQuote = true)
```

**7.15.3.24 HasNoBrackets()**

```
bool GmatStringUtil::HasNoBrackets (
 const std::string & str,
 bool parensForArraysAllowed = true)
```

#### 7.15.3.25 IsBlank()

```
bool GmatStringUtil::IsBlank (
 const std::string & str,
 bool ignoreEol = false)
```

#### 7.15.3.26 IsBracketBalanced()

```
bool GmatStringUtil::IsBracketBalanced (
 const std::string & str,
 const std::string & bracketPair)
```

#### 7.15.3.27 IsBracketPartOfArray()

```
bool GmatStringUtil::IsBracketPartOfArray (
 const std::string & str,
 const std::string & bracketPairs,
 bool checkOnlyFirst)
```

#### 7.15.3.28 IsCommaPartOfArray()

```
bool GmatStringUtil::IsCommaPartOfArray (
 const std::string & str,
 Integer start = 0)
```

#### 7.15.3.29 IsEnclosedWith()

```
bool GmatStringUtil::IsEnclosedWith (
 const std::string & str,
 const std::string & enclosingStr)
```

#### 7.15.3.30 IsEnclosedWithBraces()

```
bool GmatStringUtil::IsEnclosedWithBraces (
 const std::string & str)
```

**7.15.3.31 IsEnclosedWithBrackets()**

```
bool GmatStringUtil::IsEnclosedWithBrackets (
 const std::string & str)
```

**7.15.3.32 IsEnclosedWithExtraParen()**

```
bool GmatStringUtil::IsEnclosedWithExtraParen (
 const std::string & str,
 bool checkOps = true,
 bool ignoreComma = false)
```

**7.15.3.33 IsLastNumberPartOfName()**

```
bool GmatStringUtil::IsLastNumberPartOfName (
 const std::string & str)
```

**7.15.3.34 IsMathEquation()**

```
bool GmatStringUtil::IsMathEquation (
 const std::string & str,
 bool checkInvalidOpOnly = false,
 bool blankNameIsOk = false)
```

**7.15.3.35 IsMathOperator()**

```
bool GmatStringUtil::IsMathOperator (
 const char & ch)
```

**7.15.3.36 IsNumber()**

```
bool GmatStringUtil::IsNumber (
 const std::string & str)
```

**7.15.3.37 IsOneElementArray()**

```
bool GmatStringUtil::IsOneElementArray (
 const std::string & str)
```

**7.15.3.38 IsOuterParen()**

```
bool GmatStringUtil::IsOuterParen (
 const std::string & str)
```

**7.15.3.39 IsParenBalanced()**

```
bool GmatStringUtil::IsParenBalanced (
 const std::string & str)
```

**7.15.3.40 IsParenEmpty()**

```
bool GmatStringUtil::IsParenEmpty (
 const std::string & str)
```

Checks if there is nothing inside parenthesis such as () or (()).

**7.15.3.41 IsParenPartOfArray()**

```
bool GmatStringUtil::IsParenPartOfArray (
 const std::string & str)
```

**7.15.3.42 IsSimpleArrayElement()**

```
bool GmatStringUtil::IsSimpleArrayElement (
 const std::string & str)
```

Checks if string is a simple array element such a(1,1), b(c,d). It will return false for a(b(1,1), c(1,1)) or a(1+1, 2\*2).

**7.15.3.43 IsSingleItem()**

```
bool GmatStringUtil::IsSingleItem (
 const std::string & str)
```

#### 7.15.3.44 IsStringInsideSymbols()

```
bool GmatStringUtil::IsStringInsideSymbols (
 const std::string & str,
 const std::string & reqStr,
 const std::string & symbol,
 std::string::size_type & reqStrPos)
```

Checks if reqStr is inside symbols. Return true if reqStr found and it is inside symbols. It updates position of reqStr.

If symbol has two characters, it will check reqStr between first and second characters. for example: IsStringInsideSymbols(inputStr, ";", "[", strPos);

#### Returns

true if reqStr found and it is inside symbols false if reqStr not found or reqStr is not between symbols

#### 7.15.3.45 IsThereEqualSign()

```
bool GmatStringUtil::IsThereEqualSign (
 const std::string & str)
```

#### 7.15.3.46 IsThereMathSymbol()

```
bool GmatStringUtil::IsThereMathSymbol (
 const std::string & str)
```

#### 7.15.3.47 IsValidExtendedIdentity()

```
bool GmatStringUtil::IsValidExtendedIdentity (
 const std::string & str)
```

#### 7.15.3.48 IsValidFileName()

```
bool GmatStringUtil::IsValidFileName (
 const std::string & str)
```



#### 7.15.3.49 IsValidFullFileName()

```
bool GmatStringUtil::IsValidFullFileName (
 const std::string & str,
 Integer & error)
```

#### 7.15.3.50 IsValidFunctionCall()

```
bool GmatStringUtil::IsValidFunctionCall (
 const std::string & str)
```

Checks if function arguments has valid names or string literal. If input is not a function call, it returns false.

#### 7.15.3.51 IsValidIdentity()

```
bool GmatStringUtil::IsValidIdentity (
 const std::string & str)
```

#### 7.15.3.52 IsValidName() [1/2]

```
bool GmatStringUtil::IsValidName (
 const char * str,
 bool ignoreBracket = false,
 bool blankNameIsOk = false)
```

#### 7.15.3.53 IsValidName() [2/2]

```
bool GmatStringUtil::IsValidName (
 const std::string & str,
 bool ignoreBracket = false,
 bool blankNameIsOk = false)
```

#### 7.15.3.54 IsValidNumber()

```
bool GmatStringUtil::IsValidNumber (
 const std::string & str,
 bool allowOverflow = true)
```

#### 7.15.3.55 IsValidParameterName()

```
bool GmatStringUtil::IsValidParameterName (
 const std::string & str)
```

Checks if input string a valid Parameter name syntax such as sat.X or sat.Earth.TA. It does not check for the actual Parameter name such as X or TA.

#### 7.15.3.56 IsValidReal()

```
bool GmatStringUtil::IsValidReal (
 const std::string & str,
 Real & value,
 Integer & errorCode,
 bool trimParens = false,
 bool allowOverflow = true)
```

#### 7.15.3.57 MakeCommentLines()

```
std::string GmatStringUtil::MakeCommentLines (
 const std::string & str,
 bool breakAtCr = false)
```

Converts input string to comment lines by adding % at the beginning of non blank lines.

#### 7.15.3.58 NumberOfOccurrences()

```
Integer GmatStringUtil::NumberOfOccurrences (
 const std::string & str,
 const char c)
```

#### 7.15.3.59 NumberOfScientificNotation()

```
Integer GmatStringUtil::NumberOfScientificNotation (
 const std::string & str)
```

Checks if input string contains scientific notation and returns number of occurrence.

#### 7.15.3.60 PadWithBlanks()

```
std::string GmatStringUtil::PadWithBlanks (
 const std::string & str,
 Integer toSize,
 StripType whichEnd = TRAILING)
```

#### 7.15.3.61 ParseFunctionCall()

```
StringArray GmatStringUtil::ParseFunctionCall (
 const std::string & str)
```

Parses function call from the following syntax and return function name and function arguments. [out] = Function1(in); Function2(in); Function3; out = Function4(in, 'string literal')

**Parameters**

|            |              |
|------------|--------------|
| <i>str</i> | Input string |
|------------|--------------|

**Returns**

Function name and arguments

**7.15.3.62 ParseFunctionName()**

```
std::string GmatStringUtil::ParseFunctionName (
 const std::string & str,
 std::string & argStr)
```

Parses function name from the following syntax: [out] = Function1(in); Function2(in); Function3;

**Parameters**

|               |                                       |
|---------------|---------------------------------------|
| <i>str</i>    | Input string                          |
| <i>argStr</i> | output function argument including () |

**Returns**

Function name or blank if name not found

**7.15.3.63 ParseName()**

```
StringArray GmatStringUtil::ParseName (
 const std::string & extendedName)
```

**7.15.3.64 ParseParameter()**

```
void GmatStringUtil::ParseParameter (
 const std::string & str,
 std::string & type,
 std::string & owner,
 std::string & dep)
```

**7.15.3.65 RealToString()** [1/2]

```
std::string GmatStringUtil::RealToString (
 const Real & val,
 bool useCurrentFormat = true,
 bool scientific = false,
 bool showPoint = true,
 Integer precision = GmatGlobal::DATA_PRECISION,
 Integer width = GmatGlobal::DATA_WIDTH)
```

**7.15.3.66 RealToString()** [2/2]

```
std::string GmatStringUtil::RealToString (
 const Real & val,
 Integer precision,
 bool showPoint = false,
 Integer width = 1)
```

**7.15.3.67 RemoveAll()** [1/2]

```
std::string GmatStringUtil::RemoveAll (
 const std::string & str,
 char ch,
 Integer start = 0)
```

This file provides string utility functions. Removes all occurrence of input ch starting from index start.

**7.15.3.68 RemoveAll()** [2/2]

```
std::string GmatStringUtil::RemoveAll (
 const std::string & str,
 const std::string & removeStr,
 Integer start = 0)
```

Removes all occurrence of characters in removeStr starting at start.

**7.15.3.69 RemoveAllBlanks()**

```
std::string GmatStringUtil::RemoveAllBlanks (
 const std::string & str,
 bool ignoreSingleQuotes = false)
```

#### 7.15.3.70 RemoveEnclosingString()

```
std::string GmatStringUtil::RemoveEnclosingString (
 const std::string & str,
 const std::string & enStr)
```

#### 7.15.3.71 RemoveExtraParen()

```
std::string GmatStringUtil::RemoveExtraParen (
 const std::string & str,
 bool ignoreComma = false,
 bool ignoreSingleQuotes = false)
```

#### 7.15.3.72 RemoveInlineComment()

```
std::string GmatStringUtil::RemoveInlineComment (
 const std::string & str,
 const std::string & cmStr)
```

#### 7.15.3.73 RemoveLastNumber()

```
std::string GmatStringUtil::RemoveLastNumber (
 const std::string & str,
 Integer & lastNumber)
```

#### 7.15.3.74 RemoveLastString()

```
std::string GmatStringUtil::RemoveLastString (
 const std::string & str,
 const std::string & lastStr,
 bool removeAll = false)
```

#### 7.15.3.75 RemoveMathSymbols()

```
std::string GmatStringUtil::RemoveMathSymbols (
 const std::string & str,
 bool removeMathOperator = false)
```

**7.15.3.76 RemoveMultipleSpaces()**

```
std::string GmatStringUtil::RemoveMultipleSpaces (
 const std::string & str)
```

**7.15.3.77 RemoveOuterParen()**

```
std::string GmatStringUtil::RemoveOuterParen (
 const std::string & str)
```

Removes outer pair of parenthesis if it has one.

**7.15.3.78 RemoveOuterString()**

```
std::string GmatStringUtil::RemoveOuterString (
 const std::string & str,
 const std::string & start,
 const std::string & end)
```

**7.15.3.79 RemoveScientificNotation()**

```
std::string GmatStringUtil::RemoveScientificNotation (
 const std::string & str)
```

Replaces scientific notations with zeros. If string contains equal sign, it replaces scientific notations in right hand side of the first equal sign to zeros. For example, 1.5e+10 to 1.50010, 1.5E-11 to 1.50011.

**7.15.3.80 RemoveSpaceInBrackets()**

```
std::string GmatStringUtil::RemoveSpaceInBrackets (
 const std::string & str,
 const std::string & bracketPair)
```

**7.15.3.81 RemoveTrailingZeros()**

```
std::string GmatStringUtil::RemoveTrailingZeros (
 Real val,
 const std::string & valStr,
 Integer iterCount = 0)
```

Removes trailing zeros from real number string. It goes through another iteration using less precision format for numbers ending .999999.

## Parameters

|               |                                                |
|---------------|------------------------------------------------|
| <i>val</i>    | Real number                                    |
| <i>valStr</i> | String of real number to remove trailing zeros |

**7.15.3.82 Replace()**

```
std::string GmatStringUtil::Replace (
 const std::string & str,
 const std::string & from,
 const std::string & to,
 std::string::size_type startIndex = 0)
```

**7.15.3.83 ReplaceChainedUnaryOperators()**

```
std::string GmatStringUtil::ReplaceChainedUnaryOperators (
 const std::string & str)
```

Replaces repeated plus (+) or minus (-) signs with one sign. For example "+--+abc+---def+--+ghi" will give "+abc-def\_ghi".

**7.15.3.84 ReplaceFirst()**

```
std::string GmatStringUtil::ReplaceFirst (
 const std::string & str,
 const std::string & from,
 const std::string & to,
 std::string::size_type startIndex = 0)
```

**7.15.3.85 ReplaceName()**

```
std::string GmatStringUtil::ReplaceName (
 const std::string & str,
 const std::string & from,
 const std::string & to)
```

**7.15.3.86 ReplaceNumber()**

```
std::string GmatStringUtil::ReplaceNumber (
 const std::string & str,
 const std::string & from,
 const std::string & to)
```

#### 7.15.3.87 SeparateBrackets()

```
StringArray GmatStringUtil::SeparateBrackets (
 const std::string & chunk,
 const std::string & bracketPair,
 const std::string & delim,
 bool checkOuterBracket = true)
```

#### 7.15.3.88 SeparateBy()

```
StringArray GmatStringUtil::SeparateBy (
 const std::string & str,
 const std::string & delim,
 bool putBracketsTogether = false,
 bool insertDelim = false,
 bool insertComma = true)
```

#### 7.15.3.89 SeparateByComma()

```
StringArray GmatStringUtil::SeparateByComma (
 const std::string & str,
 bool checkSingleQuote = true)
```

#### 7.15.3.90 SeparateDots()

```
StringArray GmatStringUtil::SeparateDots (
 const std::string & str)
```

#### 7.15.3.91 StartsWith()

```
bool GmatStringUtil::StartsWith (
 const std::string & str,
 const std::string & value)
```

#### 7.15.3.92 StringToWideString()

```
std::wstring GmatStringUtil::StringToWideString (
 const std::string & str)
```

Converts narrow string (std::string) to wide string (std::wstring).



#### 7.15.3.93 Strip()

```
std::string GmatStringUtil::Strip (
 const std::string & str,
 StripType stype = BOTH)
```

#### 7.15.3.94 ToBoolean() [1/2]

```
bool GmatStringUtil::ToBoolean (
 const std::string & str,
 bool * value,
 bool trimParens = false)
```

#### 7.15.3.95 ToBoolean() [2/2]

```
bool GmatStringUtil::ToBoolean (
 const std::string & str,
 bool & value,
 bool trimParens = false)
```

#### 7.15.3.96 ToBooleanArray()

```
BooleanArray GmatStringUtil::ToBooleanArray (
 const std::string & str)
```

#### 7.15.3.97 ToInteger() [1/3]

```
bool GmatStringUtil::ToInteger (
 const std::string & str,
 Integer * value,
 bool trimParens = false,
 bool allowOverflow = true)
```

#### 7.15.3.98 ToInteger() [2/3]

```
bool GmatStringUtil::ToInteger (
 const char * str,
 Integer & value,
 bool trimParens = false,
 bool allowOverflow = true)
```

**7.15.3.99 ToInteger()** [3/3]

```
bool GmatStringUtil::ToInteger (
 const std::string & str,
 Integer & value,
 bool trimParens = false,
 bool allowOverflow = true)
```

**7.15.3.100 ToIntegerArray()** [1/2]

```
IntegerArray GmatStringUtil::ToIntegerArray (
 const char * str,
 bool allowOverflow = true)
```

**7.15.3.101 ToIntegerArray()** [2/2]

```
IntegerArray GmatStringUtil::ToIntegerArray (
 const std::string & str,
 bool allowOverflow = true)
```

**7.15.3.102 ToLower()**

```
std::string GmatStringUtil::ToLower (
 const std::string & str,
 bool firstLetterOnly = false)
```

Makes whole string or first letter lower case.

**7.15.3.103 ToOnOff()**

```
bool GmatStringUtil::ToOnOff (
 const std::string & str,
 std::string & value,
 bool trimParens = false)
```

**7.15.3.104 ToOrdinal()**

```
std::string GmatStringUtil::ToOrdinal (
 Integer i,
 bool textOnly = false)
```

Returns the ordinal number, given the input integer.

**7.15.3.105 ToReal()** [1/4]

```
bool GmatStringUtil::ToReal (
 const char * str,
 Real * value,
 bool trimParens = false,
 bool allowOverflow = true)
```

**7.15.3.106 ToReal()** [2/4]

```
bool GmatStringUtil::ToReal (
 const std::string & str,
 Real * value,
 bool trimParens = false,
 bool allowOverflow = true)
```

**7.15.3.107 ToReal()** [3/4]

```
bool GmatStringUtil::ToReal (
 const char * str,
 Real & value,
 bool trimParens = false,
 bool allowOverflow = true)
```

**7.15.3.108 ToReal()** [4/4]

```
bool GmatStringUtil::ToReal (
 const std::string & str,
 Real & value,
 bool trimParens = false,
 bool allowOverflow = true)
```

**7.15.3.109 ToRealArray()**

```
RealArray GmatStringUtil::ToRealArray (
 const std::string & str,
 bool allowOverflow = true,
 bool allowSemicolon = false)
```

**7.15.3.110 ToString()** [1/5]

```
std::string GmatStringUtil::ToString (
 const bool & val)
```

**7.15.3.111 ToString()** [2/5]

```
std::string GmatStringUtil::ToString (
 const Real & val,
 Integer precision,
 bool showPoint = false,
 Integer width = 1)
```

**7.15.3.112 ToString()** [3/5]

```
std::string GmatStringUtil::ToString (
 const Integer & val,
 Integer width)
```

**7.15.3.113 ToString()** [4/5]

```
std::string GmatStringUtil::ToString (
 const Real & val,
 bool useCurrentFormat = true,
 bool scientific = false,
 bool showPoint = true,
 Integer precision = GmatGlobal::DATA_PRECISION,
 Integer width = GmatGlobal::DATA_WIDTH)
```

**7.15.3.114 ToString()** [5/5]

```
std::string GmatStringUtil::ToString (
 const Integer & val,
 bool useCurrentFormat = true,
 Integer width = GmatGlobal::INTEGER_WIDTH)
```

**7.15.3.115 ToStringArray()**

```
StringArray GmatStringUtil::ToStringArray (
 const std::string & str)
```

**7.15.3.116 ToStringNoZeros()**

```
std::string GmatStringUtil::ToStringNoZeros (
 const Real & val)
```

Formats real number to string without trailing zeros.

**7.15.3.117 ToUnsignedInt()** [1/2]

```
bool GmatStringUtil::ToUnsignedInt (
 const std::string & str,
 UnsignedInt * value,
 bool trimParens = false,
 bool allowOverflow = true)
```

**7.15.3.118 ToUnsignedInt()** [2/2]

```
bool GmatStringUtil::ToUnsignedInt (
 const std::string & str,
 UnsignedInt & value,
 bool trimParens = false,
 bool allowOverflow = true)
```

**7.15.3.119 ToUnsignedIntArray()**

```
UnsignedIntArray GmatStringUtil::ToUnsignedIntArray (
 const std::string & str,
 bool allowOverflow = true)
```

**7.15.3.120 ToUpper()**

```
std::string GmatStringUtil::ToUpper (
 const std::string & str,
 bool firstLetterOnly = false)
```

Makes whole string or first letter upper case.

**7.15.3.121 Trim()**

```
std::string GmatStringUtil::Trim (
 const std::string & str,
 StripType stype = BOTH,
 bool removeSemicolon = false,
 bool removeEol = false)
```

**7.15.3.122 WideStringToString()** [1/2]

```
std::string GmatStringUtil::WideStringToString (
 const std::wstring & wstr)
```

Converts wide string (std::wstring) to narrow string (std::string).

**7.15.3.123 WideStringToString()** [2/2]

```
std::string GmatStringUtil::WideStringToString (
 const wchar_t * wchar)
```

Converts wide string (wchar\_t\*) to narrow string (std::string).

**7.15.3.124 WriteStringArray()**

```
void GmatStringUtil::WriteStringArray (
 const StringArray & strArray,
 const std::string & desc = "",
 const std::string & prefix = "")
```

**7.16 GmatTimeConstants Namespace Reference****Enumerations**

- enum [DayName](#) {  
[SUNDAY](#), [MONDAY](#), [TUESDAY](#), [WEDNESDAY](#),  
[THURSDAY](#), [FRIDAY](#), [SATURDAY](#) }
- enum [MonthName](#) {  
[JANUARY](#) = 1, [FEBRUARY](#), [MARCH](#), [APRIL](#),  
[MAY](#), [JUNE](#), [JULY](#), [AUGUST](#),  
[SEPTEMBER](#), [OCTOBER](#), [NOVEMBER](#), [DECEMBER](#) }

**Variables**

- const [Real](#) [SECS\\_PER\\_DAY](#) = 86400.0
- const [Real](#) [SECS\\_PER\\_HOUR](#) = 3600.0
- const [Real](#) [SECS\\_PER\\_MINUTE](#) = 60.0
- const [Real](#) [DAYS\\_PER\\_YEAR](#) = 365.25
- const [Real](#) [DAYS\\_PER\\_JULIAN\\_CENTURY](#) = 36525.00
- const [Real](#) [DAYS\\_PER\\_SEC](#) = 1.1574074074074074074074074074e-5
- const [Real](#) [TIME\\_OF\\_J2000](#) = 883655990.850000
- const [Real](#) [JD\\_OF\\_J2000](#) = 2451545.0
- const [Real](#) [MJD\\_OF\\_J2000](#) = 21545.00000000
- const [Real](#) [A1MJD\\_OF\\_J2000](#) = 21545.00000000
- const [Real](#) [JD\\_MJD\\_OFFSET](#) = 2400000.5
- const [Real](#) [TT\\_TAI\\_OFFSET](#) = 32.184
- const [Real](#) [A1\\_TAI\\_OFFSET](#) = 0.0343817
- const [Real](#) [JD\\_JAN\\_5\\_1941](#) = 2430000.0
- const [Real](#) [JD\\_NOV\\_17\\_1858](#) = 2400000.5
- const [Integer](#) [DAYS\\_BEFORE\\_MONTH](#) [12]
- const [Integer](#) [LEAP\\_YEAR\\_DAYS\\_BEFORE\\_MONTH](#) [12]
- const [Integer](#) [DAYS\\_IN\\_MONTH](#) [12]
- const [Integer](#) [LEAP\\_YEAR\\_DAYS\\_IN\\_MONTH](#) [12]
- const [Integer](#) [JULIAN\\_DATE\\_OF\\_010541](#) = 2430000
- const [Real](#) [MJD\\_EPOCH\\_PRECISION](#) = 7.27e-12

## 7.16.1 Enumeration Type Documentation

### 7.16.1.1 DayName

enum `GmatTimeConstants::DayName`

#### Enumerator

|           |  |
|-----------|--|
| SUNDAY    |  |
| MONDAY    |  |
| TUESDAY   |  |
| WEDNESDAY |  |
| THURSDAY  |  |
| FRIDAY    |  |
| SATURDAY  |  |

### 7.16.1.2 MonthName

enum `GmatTimeConstants::MonthName`

#### Enumerator

|           |  |
|-----------|--|
| JANUARY   |  |
| FEBRUARY  |  |
| MARCH     |  |
| APRIL     |  |
| MAY       |  |
| JUNE      |  |
| JULY      |  |
| AUGUST    |  |
| SEPTEMBER |  |
| OCTOBER   |  |
| NOVEMBER  |  |
| DECEMBER  |  |

## 7.16.2 Variable Documentation

### 7.16.2.1 A1\_TAI\_OFFSET

const `Real` `GmatTimeConstants::A1_TAI_OFFSET` = 0.0343817

#### 7.16.2.2 A1MJD\_OF\_J2000

```
const Real GmatTimeConstants::A1MJD_OF_J2000 = 21545.00000000
```

#### 7.16.2.3 DAYS\_BEFORE\_MONTH

```
const Integer GmatTimeConstants::DAYS_BEFORE_MONTH[12]
```

**Initial value:**

```
=
{0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334}
```

#### 7.16.2.4 DAYS\_IN\_MONTH

```
const Integer GmatTimeConstants::DAYS_IN_MONTH[12]
```

**Initial value:**

```
=
{31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}
```

#### 7.16.2.5 DAYS\_PER\_JULIAN\_CENTURY

```
const Real GmatTimeConstants::DAYS_PER_JULIAN_CENTURY = 36525.00
```

#### 7.16.2.6 DAYS\_PER\_SEC

```
const Real GmatTimeConstants::DAYS_PER_SEC = 1.1574074074074074074074074e-5
```

#### 7.16.2.7 DAYS\_PER\_YEAR

```
const Real GmatTimeConstants::DAYS_PER_YEAR = 365.25
```



### 7.16.2.8 JD\_JAN\_5\_1941

```
const Real GmatTimeConstants::JD_JAN_5_1941 = 2430000.0
```

### 7.16.2.9 JD\_MJD\_OFFSET

```
const Real GmatTimeConstants::JD_MJD_OFFSET = 2400000.5
```

### 7.16.2.10 JD\_NOV\_17\_1858

```
const Real GmatTimeConstants::JD_NOV_17_1858 = 2400000.5
```

### 7.16.2.11 JD\_OF\_J2000

```
const Real GmatTimeConstants::JD_OF_J2000 = 2451545.0
```

### 7.16.2.12 JULIAN\_DATE\_OF\_010541

```
const Integer GmatTimeConstants::JULIAN_DATE_OF_010541 = 2430000
```

### 7.16.2.13 LEAP\_YEAR\_DAYS\_BEFORE\_MONTH

```
const Integer GmatTimeConstants::LEAP_YEAR_DAYS_BEFORE_MONTH[12]
```

**Initial value:**

```
=
{0, 31, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335}
```

#### 7.16.2.14 LEAP\_YEAR\_DAYS\_IN\_MONTH

```
const Integer GmatTimeConstants::LEAP_YEAR_DAYS_IN_MONTH[12]
```

**Initial value:**

```
=
{31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}
```

#### 7.16.2.15 MJD\_EPOCH\_PRECISION

```
const Real GmatTimeConstants::MJD_EPOCH_PRECISION = 7.27e-12
```

#### 7.16.2.16 MJD\_OF\_J2000

```
const Real GmatTimeConstants::MJD_OF_J2000 = 21545.00000000
```

#### 7.16.2.17 SECS\_PER\_DAY

```
const Real GmatTimeConstants::SECS_PER_DAY = 86400.0
```

#### 7.16.2.18 SECS\_PER\_HOUR

```
const Real GmatTimeConstants::SECS_PER_HOUR = 3600.0
```

#### 7.16.2.19 SECS\_PER\_MINUTE

```
const Real GmatTimeConstants::SECS_PER_MINUTE = 60.0
```

#### 7.16.2.20 TIME\_OF\_J2000

```
const Real GmatTimeConstants::TIME_OF_J2000 = 883655990.850000
```

## 7.16.2.21 TT\_TAI\_OFFSET

```
const Real GmatTimeConstants::TT_TAI_OFFSET = 32.184
```

## 7.17 GmatTimeUtil Namespace Reference

### Classes

- class [CalDate](#)
- class [ElapsedDate](#)

### Functions

- bool [GMATUTIL\\_API IsValidMonthName](#) (const std::string &str)
- std::string [GMATUTIL\\_API GetMonthName](#) ([Integer](#) month)
- [Integer](#) [GMATUTIL\\_API GetMonth](#) (const std::string &monthName)
- std::string [GMATUTIL\\_API FormatCurrentTime](#) ([Integer](#) format=1)
- std::string [GMATUTIL\\_API GetGregorianFormat](#) ()

### 7.17.1 Function Documentation

#### 7.17.1.1 FormatCurrentTime()

```
std::string GmatTimeUtil::FormatCurrentTime (
 Integer format = 1)
```

Add microseconds of the current time

#### 7.17.1.2 GetGregorianFormat()

```
std::string GmatTimeUtil::GetGregorianFormat ()
```

Return gregorian time format

#### 7.17.1.3 GetMonth()

```
Integer GmatTimeUtil::GetMonth (
 const std::string & monthName)
```

#### 7.17.1.4 GetMonthName()

```
std::string GmatTimeUtil::GetMonthName (
 Integer month)
```

#### 7.17.1.5 IsValidMonthName()

```
bool GmatTimeUtil::IsValidMonthName (
 const std::string & str)
```

Provides declarations for date & time types.



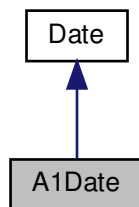
## Chapter 8

# Class Documentation

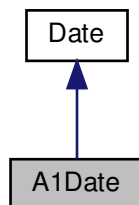
### 8.1 A1Date Class Reference

```
#include <A1Date.hpp>
```

Inheritance diagram for A1Date:



Collaboration diagram for A1Date:



## Public Member Functions

- [A1Date](#) ()
- [A1Date](#) ([Integer](#) year, [Integer](#) month, [Integer](#) day, [Integer](#) hour, [Integer](#) minute, [Real](#) second)
- [A1Date](#) ([Integer](#) year, [Integer](#) day, [Integer](#) hour, [Integer](#) minute, [Real](#) second)
- [A1Date](#) ([Integer](#) year, [Integer](#) month, [Integer](#) day, [Real](#) secondsOfDay)
- [A1Date](#) (const [GmatTimeUtil::CalDate](#) &date)
- [A1Date](#) (const std::string &dateString)
- [A1Date](#) (const [A1Date](#) &a1date)
- [A1Date](#) operator= (const [A1Date](#) &a1date)
- bool operator> (const [A1Date](#) &a1date) const
- bool operator< (const [A1Date](#) &a1date) const
- ~[A1Date](#) ()

## Additional Inherited Members

### 8.1.1 Constructor & Destructor Documentation

#### 8.1.1.1 [A1Date](#)() [1/7]

```
A1Date::A1Date ()
```

Provides conversions among various ways representing A1 calendar dates and times. Default constructor.

#### Note

Calls [A1Mjd](#) default constructor which creates an object with 0 second from reference.

#### 8.1.1.2 [A1Date](#)() [2/7]

```
A1Date::A1Date (
 Integer year,
 Integer month,
 Integer day,
 Integer hour,
 Integer minute,
 Real second)
```

Constructor.

#### Note

Assumes input date is in A1 time system.

### 8.1.1.3 A1Date() [3/7]

```
A1Date::A1Date (
 Integer year,
 Integer dayOfYear,
 Integer hour,
 Integer minute,
 Real second)
```

Constructor.

#### Note

Assumes input date is in A1 time system.

### 8.1.1.4 A1Date() [4/7]

```
A1Date::A1Date (
 Integer year,
 Integer month,
 Integer day,
 Real secondsOfDay)
```

Constructor.

#### Note

Assumes input date is in A1 time system.

### 8.1.1.5 A1Date() [5/7]

```
A1Date::A1Date (
 const GmatTimeUtil::CalDate & date)
```

Constructor.

#### Note

assumes input date is in A1 time system.

### 8.1.1.6 A1Date() [6/7]

```
A1Date::A1Date (
 const std::string & dateString)
```

Constructor.

**Parameters**

|                 |                                   |
|-----------------|-----------------------------------|
| < <i>time</i> > | Time in "YYMMDD.hhmmssnnn" format |
|-----------------|-----------------------------------|

**Note**

Assumes input date is in A1 time system.

**8.1.1.7 A1Date()** [7/7]

```
A1Date::A1Date (
 const A1Date & a1date)
```

Copy constructor.

**8.1.1.8 ~A1Date()**

```
A1Date::~~A1Date ()
```

Destructor.

**8.1.2 Member Function Documentation****8.1.2.1 operator<()**

```
bool A1Date::operator< (
 const A1Date & a1date) const
```

Comparison operator <

**8.1.2.2 operator=()**

```
A1Date A1Date::operator= (
 const A1Date & a1date)
```

Assignment operator.



## 8.1.2.3 operator&gt;()

```
bool A1Date::operator> (
 const A1Date & a1date) const
```

Comparison operator >

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

- GMATsrc/util/A1Date.hpp
- GMATsrc/util/A1Date.cpp

## 8.2 A1Mjd Class Reference

```
#include <A1Mjd.hpp>
```

Collaboration diagram for A1Mjd:



## Public Member Functions

- [A1Mjd](#) ()
- [A1Mjd](#) (const [Real](#) &val)
- [A1Mjd](#) (const [A1Mjd](#) &a1mjd)
- [A1Mjd](#) & [operator=](#) (const [A1Mjd](#) &right)
- [A1Mjd](#) \* [Clone](#) () const
- [~A1Mjd](#) ()
- [A1Mjd](#) [operator+](#) (const [A1Mjd](#) &right) const
- [A1Mjd](#) [operator-](#) (const [A1Mjd](#) &right) const
- const [A1Mjd](#) & [operator+=](#) (const [A1Mjd](#) &right)
- const [A1Mjd](#) & [operator-=](#) (const [A1Mjd](#) &right)
- [A1Mjd](#) [operator+](#) (const [Real](#) &right) const
- [A1Mjd](#) [operator-](#) (const [Real](#) &right) const
- const [A1Mjd](#) & [operator+=](#) (const [Real](#) &right)
- const [A1Mjd](#) & [operator-=](#) (const [Real](#) &right)
- bool [operator<](#) (const [A1Mjd](#) &right) const
- bool [operator>](#) (const [A1Mjd](#) &right) const
- bool [operator==](#) (const [A1Mjd](#) &right) const
- bool [operator!=](#) (const [A1Mjd](#) &right) const
- bool [operator>=](#) (const [A1Mjd](#) &right) const
- bool [operator<=](#) (const [A1Mjd](#) &right) const
- [Real](#) [Subtract](#) (const [A1Mjd](#) &right) const

- [Real Subtract](#) (const [Real](#) &right) const
- [Real Get](#) () const
- void [Set](#) ([Real](#) val)
- [Real GetReal](#) () const
- void [SetReal](#) ([Real](#) val)
- [UtcDate ToUtcDate](#) ()
- [A1Date ToA1Date](#) (bool handleLeapSecond=false)
- [Real UtcMjdToA1Mjd](#) (const [Real](#) utcMjd)
- [UtcMjd ToUtcMjd](#) ()
- [Integer GetNumData](#) () const
- const std::string \* [GetDataDescriptions](#) () const
- std::string \* [ToValueStrings](#) ()

## Static Public Attributes

- static const [A1Mjd J2000](#) = [A1Mjd](#)(A1MJD\_OF\_J2000)

## 8.2.1 Detailed Description

This class provides A1 Modified Julian [Date\(MJD\)](#). The zero date of the MJD 12 noon on January 5th, 1941.

## 8.2.2 Constructor & Destructor Documentation

### 8.2.2.1 [A1Mjd\(\)](#) [1/3]

```
A1Mjd::A1Mjd ()
```

Default constructor. Creates [A1Mjd](#) time of J2000.

### 8.2.2.2 [A1Mjd\(\)](#) [2/3]

```
A1Mjd::A1Mjd (
 const Real & val)
```

Constructor. Creates [A1Mjd](#) time from Real value.

#### Parameters

|       |                                |
|-------|--------------------------------|
| <val> | the value to create an object. |
|-------|--------------------------------|

### 8.2.2.3 A1Mjd() [3/3]

```
A1Mjd::A1Mjd (
 const A1Mjd & a1mjd)
```

Copy constructor.

#### Parameters

|                                   |                          |
|-----------------------------------|--------------------------|
| <code>&lt;<i>a1mjd</i>&gt;</code> | the object to be copied. |
|-----------------------------------|--------------------------|

### 8.2.2.4 ~A1Mjd()

```
A1Mjd::~~A1Mjd ()
```

Destructor.

## 8.2.3 Member Function Documentation

### 8.2.3.1 Clone()

```
A1Mjd * A1Mjd::Clone () const
```

### 8.2.3.2 Get()

```
Real A1Mjd::Get () const [inline]
```

### 8.2.3.3 GetDataDescriptions()

```
const std::string * A1Mjd::GetDataDescriptions () const
```

#### Returns

data description pointer.

#### 8.2.3.4 GetNumData()

```
Integer AlMjd::GetNumData () const
```

##### Returns

number of data elements.

#### 8.2.3.5 GetReal()

```
Real AlMjd::GetReal () const [inline]
```

#### 8.2.3.6 operator!=(())

```
bool AlMjd::operator!= (
 const AlMjd & right) const
```

Determins if this object value is not equal to another object value.

##### Parameters

|                      |                 |
|----------------------|-----------------|
| <i>&lt;right&gt;</i> | another object. |
|----------------------|-----------------|

##### Returns

true if this object value is not equal to another object value; false otherwise.

#### 8.2.3.7 operator+() [1/2]

```
AlMjd AlMjd::operator+ (
 const AlMjd & right) const
```

Adds an object and return a new object.

##### Returns

a new object.

**8.2.3.8 operator+()** [2/2]

```
A1Mjd A1Mjd::operator+ (
 const Real & right) const
```

Adds a Real value and return a new object.

**8.2.3.9 operator+=()** [1/2]

```
const A1Mjd & A1Mjd::operator+= (
 const A1Mjd & right)
```

Adds an object and return the same object.

**8.2.3.10 operator+=()** [2/2]

```
const A1Mjd & A1Mjd::operator+= (
 const Real & right)
```

Adds a Real value and return the same object.

**8.2.3.11 operator-()** [1/2]

```
A1Mjd A1Mjd::operator- (
 const A1Mjd & right) const
```

Subtracts a object and return a new object.

**Returns**

a new object.

**8.2.3.12 operator-()** [2/2]

```
A1Mjd A1Mjd::operator- (
 const Real & right) const
```

Subtracts a Real value and return a new object.

**8.2.3.13 operator-=()** [1/2]

```
const A1Mjd & A1Mjd::operator-= (
 const A1Mjd & right)
```

Subtracts an object and return the same object.

**8.2.3.14 operator-=()** [2/2]

```
const AlMjd & AlMjd::operator-= (
 const Real & right)
```

Subtracts a Real value and return the same object.

**8.2.3.15 operator<()**

```
bool AlMjd::operator< (
 const AlMjd & right) const [inline]
```

**8.2.3.16 operator<=()**

```
bool AlMjd::operator<= (
 const AlMjd & right) const [inline]
```

**8.2.3.17 operator=()**

```
AlMjd & AlMjd::operator= (
 const AlMjd & right)
```

Assignment operator.

**Parameters**

|         |                          |
|---------|--------------------------|
| <right> | the object to be copied. |
|---------|--------------------------|

**Returns**

reference to this object.

**8.2.3.18 operator==(**

```
bool AlMjd::operator==(
 const AlMjd & right) const
```

Determines if this object value is less than another object value.

**Parameters**

|         |                 |
|---------|-----------------|
| <right> | another object. |
|---------|-----------------|

**Returns**

true if this object value is less than other object value; false otherwise. Determines if this object value is greater than another object value.

**Parameters**

|                      |                 |
|----------------------|-----------------|
| <i>&lt;right&gt;</i> | another object. |
|----------------------|-----------------|

**Returns**

true if this object value is greater than other object value; false otherwise. Determines if this object value is equal to another object value.

**Parameters**

|                      |                 |
|----------------------|-----------------|
| <i>&lt;right&gt;</i> | another object. |
|----------------------|-----------------|

**Returns**

true if this object value is equal to another object value; false otherwise.

**8.2.3.19 operator>()**

```
bool A1Mjd::operator> (
 const A1Mjd & right) const [inline]
```

**8.2.3.20 operator>=()**

```
bool A1Mjd::operator>= (
 const A1Mjd & right) const [inline]
```

**8.2.3.21 Set()**

```
void A1Mjd::Set (
 Real val) [inline]
```

#### 8.2.3.22 SetReal()

```
void AlMjd::SetReal (
 Real val) [inline]
```

#### 8.2.3.23 Subtract() [1/2]

```
Real AlMjd::Subtract (
 const AlMjd & right) const [inline]
```

#### 8.2.3.24 Subtract() [2/2]

```
Real AlMjd::Subtract (
 const Real & right) const [inline]
```

#### 8.2.3.25 ToA1Date()

```
AlDate AlMjd::ToA1Date (
 bool handleLeapSecond = false)
```

Converts from a A1 modified Julian date to [A1Date](#) (no leap seconds)

#### 8.2.3.26 ToUtcDate()

```
UtcDate AlMjd::ToUtcDate ()
```

Determins if this object value is greater than or equal to another object value.

##### Parameters

|                            |                 |
|----------------------------|-----------------|
| <code>&lt;right&gt;</code> | another object. |
|----------------------------|-----------------|

##### Returns

true if this object value is greater than or equal to another object value; false otherwise. Determines if this object value is less than or equal to another object value.

##### Parameters

|                            |                 |
|----------------------------|-----------------|
| <code>&lt;right&gt;</code> | another object. |
|----------------------------|-----------------|



**Returns**

true if this object value is less than or equal to another object value; false otherwise. Subtracts a object and return a Real value.

**8.2.3.27 ToUtcMjd()**

```
UtcMjd A1Mjd::ToUtcMjd ()
```

**8.2.3.28 ToValueStrings()**

```
std::string * A1Mjd::ToValueStrings ()
```

**Returns**

data value string pointer.

**8.2.3.29 UtcMjdToA1Mjd()**

```
Real A1Mjd::UtcMjdToA1Mjd (
 const Real utcMjd)
```

Converts from UTC modified Julian date to a A1 modified Julian date

**8.2.4 Member Data Documentation****8.2.4.1 J2000**

```
const A1Mjd A1Mjd::J2000 = A1Mjd(A1MJD_OF_J2000) [static]
```

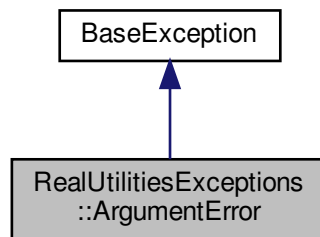
The documentation for this class was generated from the following files:

- GMATsrc/util/A1Mjd.hpp
- GMATsrc/util/A1Mjd.cpp

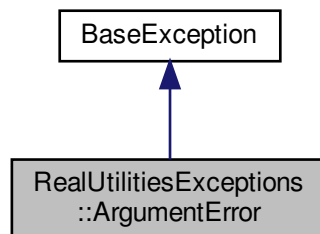
### 8.3 RealUtilitiesExceptions::ArgumentError Class Reference

```
#include <RealUtilities.hpp>
```

Inheritance diagram for RealUtilitiesExceptions::ArgumentError:



Collaboration diagram for RealUtilitiesExceptions::ArgumentError:



#### Public Member Functions

- [ArgumentError](#) (const std::string &message="")

#### Additional Inherited Members

##### 8.3.1 Constructor & Destructor Documentation

### 8.3.1.1 ArgumentError()

```
RealUtilitiesExceptions::ArgumentError::ArgumentError (
 const std::string & message = "") [inline]
```

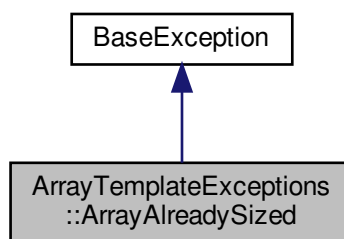
The documentation for this class was generated from the following file:

- GMATsrc/util/[RealUtilities.hpp](#)

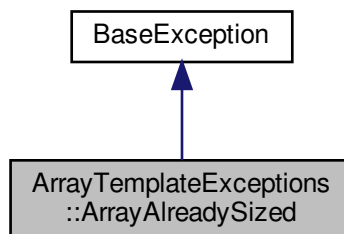
## 8.4 ArrayTemplateExceptions::ArrayAlreadySized Class Reference

```
#include <ArrayTemplate.hpp>
```

Inheritance diagram for ArrayTemplateExceptions::ArrayAlreadySized:



Collaboration diagram for ArrayTemplateExceptions::ArrayAlreadySized:



### Public Member Functions

- [ArrayAlreadySized](#) (const std::string &message="ArrayTemplate error : array already sized.")

## Additional Inherited Members

### 8.4.1 Constructor & Destructor Documentation

#### 8.4.1.1 ArrayAlreadySized()

```
ArrayTemplateExceptions::ArrayAlreadySized::ArrayAlreadySized (
 const std::string & message = "ArrayTemplate error : array already sized.")
[inline]
```

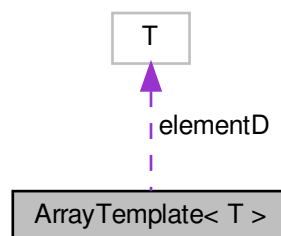
The documentation for this class was generated from the following file:

- GMATsrc/util/[ArrayTemplate.hpp](#)

## 8.5 ArrayTemplate< T > Class Template Reference

```
#include <ArrayTemplate.hpp>
```

Collaboration diagram for ArrayTemplate< T >:



## Public Member Functions

- [ArrayTemplate](#) ()
- [ArrayTemplate](#) ([Integer](#) sizeOfArray)
- [ArrayTemplate](#) ([Integer](#) sizeOfArray, const T \*array)
- [ArrayTemplate](#) (const [ArrayTemplate](#)< T > &array)
- virtual [~ArrayTemplate](#) ()
- const [ArrayTemplate](#)< T > & [operator=](#) (const [ArrayTemplate](#)< T > &array)
- bool [operator==](#) (const [ArrayTemplate](#)< T > &array) const
- bool [operator!=](#) (const [ArrayTemplate](#)< T > &array) const
- virtual T & [operator\(\)](#) ([Integer](#) index)
- virtual const T & [operator\(\)](#) ([Integer](#) index) const

- virtual T & [operator\[\]](#) ([Integer](#) index)
- virtual const T & [operator\[\]](#) ([Integer](#) index) const
- virtual bool [IsSized](#) () const
- virtual void [SetSize](#) ([Integer](#) size)
- virtual [Integer](#) [GetSize](#) () const
- virtual void [Resize](#) ([Integer](#) size)
- virtual T [GetElement](#) ([Integer](#) index) const
- virtual void [SetElement](#) ([Integer](#) index, const T &value)
- const T \* [GetDataVector](#) () const

### Protected Member Functions

- void [init](#) ([Integer](#) s)

### Protected Attributes

- T \* [elementD](#)
- [Integer](#) [sizeD](#)
- bool [isSizedD](#)

## 8.5.1 Constructor & Destructor Documentation

### 8.5.1.1 ArrayTemplate() [1/4]

```
template<class T >
ArrayTemplate< T >::ArrayTemplate ()
```

#### Note

Assumptions about template parameter types: Type has appropriate initializers and operators (constructors, "=", "==", "!=" operators) The exceptions are declared in a separate class because the current HP compiler cannot properly handle exceptions declared a template class and thrown in another template class.

Ccontains the declarations for the [ArrayTemplate](#) array container class (see Notes below for assumptions about parameter T)

### 8.5.1.2 ArrayTemplate() [2/4]

```
template<class T >
ArrayTemplate< T >::ArrayTemplate (
 Integer sizeOfArray)
```

#### 8.5.1.3 ArrayTemplate() [3/4]

```
template<class T>
ArrayTemplate< T >::ArrayTemplate (
 Integer sizeOfArray,
 const T * array)
```

#### 8.5.1.4 ArrayTemplate() [4/4]

```
template<class T>
ArrayTemplate< T >::ArrayTemplate (
 const ArrayTemplate< T > & array)
```

#### 8.5.1.5 ~ArrayTemplate()

```
template<class T >
ArrayTemplate< T >::~~ArrayTemplate () [virtual]
```

### 8.5.2 Member Function Documentation

#### 8.5.2.1 GetDataVector()

```
template<class T>
const T* ArrayTemplate< T >::GetDataVector () const [inline]
```

#### 8.5.2.2 GetElement()

```
template<class T >
T ArrayTemplate< T >::GetElement (
 Integer index) const [virtual]
```

#### 8.5.2.3 GetSize()

```
template<class T >
int ArrayTemplate< T >::GetSize () const [virtual]
```

#### 8.5.2.4 init()

```
template<class T >
void ArrayTemplate< T >::init (
 Integer s) [protected]
```

#### 8.5.2.5 IsSized()

```
template<class T >
bool ArrayTemplate< T >::IsSized () const [virtual]
```

#### 8.5.2.6 operator!=(())

```
template<class T>
bool ArrayTemplate< T >::operator!= (
 const ArrayTemplate< T > & array) const
```

#### 8.5.2.7 operator()( ) [1/2]

```
template<class T >
T & ArrayTemplate< T >::operator() (
 Integer index) [virtual]
```

#### 8.5.2.8 operator()( ) [2/2]

```
template<class T >
const T & ArrayTemplate< T >::operator() (
 Integer index) const [virtual]
```

#### 8.5.2.9 operator=()

```
template<class T>
const ArrayTemplate< T > & ArrayTemplate< T >::operator= (
 const ArrayTemplate< T > & array)
```

#### 8.5.2.10 operator==(

```
template<class T>
bool ArrayTemplate< T >::operator== (
 const ArrayTemplate< T > & array) const
```

#### 8.5.2.11 operator[]() [1/2]

```
template<class T >
T & ArrayTemplate< T >::operator[] (
 Integer index) [virtual]
```

#### 8.5.2.12 operator[]() [2/2]

```
template<class T >
const T & ArrayTemplate< T >::operator[] (
 Integer index) const [virtual]
```

#### 8.5.2.13 Resize()

```
template<class T >
void ArrayTemplate< T >::Resize (
 Integer size) [virtual]
```

#### 8.5.2.14 SetElement()

```
template<class T>
void ArrayTemplate< T >::SetElement (
 Integer index,
 const T & value) [virtual]
```

#### 8.5.2.15 SetSize()

```
template<class T >
void ArrayTemplate< T >::SetSize (
 Integer size) [virtual]
```



### 8.5.3 Member Data Documentation

#### 8.5.3.1 elementD

```
template<class T>
T* ArrayTemplate< T >::elementD [protected]
```

#### 8.5.3.2 isSizedD

```
template<class T>
bool ArrayTemplate< T >::isSizedD [protected]
```

#### 8.5.3.3 sizeD

```
template<class T>
Integer ArrayTemplate< T >::sizeD [protected]
```

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

- GMATsrc/util/[ArrayTemplate.hpp](#)
- GMATsrc/util/[ArrayTemplate.cpp](#)

## 8.6 ArrayTemplateExceptions Class Reference

```
#include <ArrayTemplate.hpp>
```

### Classes

- class [ArrayAlreadySized](#)
- class [DimensionError](#)
- class [IllegalSize](#)
- class [OutOfBounds](#)
- class [UnsizeArray](#)

#### 8.6.1 Detailed Description

Contains the declarations for the [ArrayTemplate](#) array container class (see Notes below for assumptions about parameter T)

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

- GMATsrc/util/[ArrayTemplate.hpp](#)

## 8.7 AttitudeConversionUtility Class Reference

```
#include <AttitudeConversionUtility.hpp>
```

### Static Public Member Functions

- static [Rmatrix33 ToCosineMatrix](#) (const [Rvector](#) &quat1)
- static [Rmatrix33 ToCosineMatrix](#) (const [Rvector3](#) &eulerAngles, [Integer](#) seq1, [Integer](#) seq2, [Integer](#) seq3)
- static [Rmatrix33 ToCosineMatrix](#) (const [Real](#) \*eulerAngles, [Integer](#) seq1, [Integer](#) seq2, [Integer](#) seq3)
- static [Rvector3 ToEulerAngles](#) (const [Rvector](#) &quat1, [Integer](#) seq1, [Integer](#) seq2, [Integer](#) seq3)
- static [Rvector3 ToEulerAngles](#) (const [Rmatrix33](#) &cosMat, [Integer](#) seq1, [Integer](#) seq2, [Integer](#) seq3)
- static [Rvector ToQuaternion](#) (const [Rvector3](#) &eulerAngles, [Integer](#) seq1, [Integer](#) seq2, [Integer](#) seq3)
- static [Rvector ToQuaternion](#) (const [Rmatrix33](#) &cosMat)
- static [Rvector ToQuaternion](#) (const [Rvector3](#) &MRPs)
- static [Rvector3 ToMRPs](#) (const [Rvector](#) &quat1)
- static [Rvector3 ToEulerAngleRates](#) (const [Rvector3](#) &angularVel, const [Rvector3](#) &eulerAngles, [Integer](#) seq1, [Integer](#) seq2, [Integer](#) seq3)
- static [Rvector3 ToAngularVelocity](#) (const [Rvector3](#) &eulerRates, const [Rvector3](#) &eulerAngles, [Integer](#) seq1, [Integer](#) seq2, [Integer](#) seq3)
- static [Rmatrix33 EulerAxisAndAngleToDCM](#) (const [Rvector3](#) &eAxis, [Real](#) eAngle)
- static void [DCMToEulerAxisAndAngle](#) (const [Rmatrix33](#) &cosMat, [Rvector3](#) &eAxis, [Real](#) &eAngle)
- static bool [IsValidEulerSequence](#) (const std::string &theSeq)

### 8.7.1 Detailed Description

Definition of the static class containing methods to convert between attitude state representations. This is a static class: No instances of this class may be declared.

### 8.7.2 Member Function Documentation

#### 8.7.2.1 DCMToEulerAxisAndAngle()

```
void AttitudeConversionUtility::DCMToEulerAxisAndAngle (
 const Rmatrix33 & cosMat,
 Rvector3 & eAxis,
 Real & eAngle) [static]
```

This method computes the euler axis and angle given the input cosine matrix.

#### Parameters

|          |                       |
|----------|-----------------------|
| <cosmat> | cosine matrix.        |
| <eAxis>  | euler axis (output).  |
| <eAngle> | euler angle (output). |

**Returns**

Euler Axis/Angle representation of the attitude.

**8.7.2.2 EulerAxisAndAngleToDCM()**

```
Rmatrix33 AttitudeConversionUtility::EulerAxisAndAngleToDCM (
 const Rvector3 & eAxis,
 Real eAngle) [static]
```

This method computes the direction cosine matrix given the input euler axis and angle.

**Parameters**

|          |              |
|----------|--------------|
| <eAxis>  | euler axis.  |
| <eAngle> | euler angle. |

**Returns**

Cosine matrix representation of the attitude.

**8.7.2.3 IsValidEulerSequence()**

```
bool AttitudeConversionUtility::IsValidEulerSequence (
 const std::string & theSeq) [static]
```

This method determines if the input string represents a valid Euler Rotation Sequence.

**Parameters**

|          |                       |
|----------|-----------------------|
| <theSeq> | euler sequence string |
|----------|-----------------------|

**Returns**

true if input is a valid euler sequence; false otherwise

**8.7.2.4 ToAngularVelocity()**

```
Rvector3 AttitudeConversionUtility::ToAngularVelocity (
 const Rvector3 & eulerRates,
 const Rvector3 & eulerAngles,
 Integer seq1,
```

```
Integer seq2,
Integer seq3) [static]
```

Converts the input euler angle rates to an angular velocity, using the euler sequence provided.

## Parameters

|                    |                                           |
|--------------------|-------------------------------------------|
| <i>eulerRates</i>  | the input euler angle rates (radians/sec) |
| <i>eulerAngles</i> | the input euler angles (radians)          |
| <i>seq1</i>        | first entry of the euler sequence         |
| <i>seq2</i>        | second entry of the euler sequence        |
| <i>seq3</i>        | third entry of the euler sequence         |

## Returns

the angular velocity representation (radians/second).

## Note

Obviously, the euler rates and euler angles passed in must have been computed at the same time.

## 8.7.2.5 ToCosineMatrix() [1/3]

```
Rmatrix33 AttitudeConversionUtility::ToCosineMatrix (
 const Rvector & quat1) [static]
```

Converts the input quaternion to a direction cosine matrix.

## Parameters

|              |                       |
|--------------|-----------------------|
| <i>quat1</i> | the input quaternion. |
|--------------|-----------------------|

## Returns

the cosine direction matrix representation of the input attitude.

## 8.7.2.6 ToCosineMatrix() [2/3]

```
Rmatrix33 AttitudeConversionUtility::ToCosineMatrix (
 const Rvector3 & eulerAngles,
 Integer seq1,
 Integer seq2,
 Integer seq3) [static]
```

Converts the input euler angles and sequence to a direction cosine matrix.

## Parameters

|                    |                                    |
|--------------------|------------------------------------|
| <i>eulerAngles</i> | the input euler angles (radians)   |
| <i>seq1</i>        | first entry of the euler sequence  |
| <i>seq2</i>        | second entry of the euler sequence |
| <i>seq3</i>        | third entry of the euler sequence  |

**Returns**

the cosine direction matrix representation of the input attitude.

**8.7.2.7 ToCosineMatrix()** [3/3]

```
Rmatrix33 AttitudeConversionUtility::ToCosineMatrix (
 const Real * eulerAngles,
 Integer seq1,
 Integer seq2,
 Integer seq3) [static]
```

Converts the input euler angles and sequence to a direction cosine matrix.

**Parameters**

|                    |                                    |
|--------------------|------------------------------------|
| <i>eulerAngles</i> | the input euler angles (radians)   |
| <i>seq1</i>        | first entry of the euler sequence  |
| <i>seq2</i>        | second entry of the euler sequence |
| <i>seq3</i>        | third entry of the euler sequence  |

**Returns**

the cosine direction matrix representation of the input attitude.

**8.7.2.8 ToEulerAngleRates()**

```
Rvector3 AttitudeConversionUtility::ToEulerAngleRates (
 const Rvector3 & angularVel,
 const Rvector3 & eulerAngles,
 Integer seq1,
 Integer seq2,
 Integer seq3) [static]
```

Converts the input angular velocity to a set of euler angle rates, using the euler sequence provided.

**Parameters**

|                    |                                          |
|--------------------|------------------------------------------|
| <i>angVel</i>      | the input angular velocity (radians/sec) |
| <i>eulerAngles</i> | the input euler angles (radians)         |
| <i>seq1</i>        | first entry of the euler sequence        |
| <i>seq2</i>        | second entry of the euler sequence       |
| <i>seq3</i>        | third entry of the euler sequence        |

**Returns**

the euler angle rates representation (radians/second).

**Note**

Obviously, the angular velocity and euler angles passed in must have been computed at the same time.

**8.7.2.9 ToEulerAngles() [1/2]**

```
Rvector3 AttitudeConversionUtility::ToEulerAngles (
 const Rvector & quat1,
 Integer seq1,
 Integer seq2,
 Integer seq3) [static]
```

Converts the input quaternion to a set of euler angles, using the euler sequence provided.

**Parameters**

|              |                                    |
|--------------|------------------------------------|
| <i>quat1</i> | the input quaternion.              |
| <i>seq1</i>  | first entry of the euler sequence  |
| <i>seq2</i>  | second entry of the euler sequence |
| <i>seq3</i>  | third entry of the euler sequence  |

**Returns**

the euler angles representation of the input attitude (radians)

**8.7.2.10 ToEulerAngles() [2/2]**

```
Rvector3 AttitudeConversionUtility::ToEulerAngles (
 const Rmatrix33 & cosMat,
 Integer seq1,
 Integer seq2,
 Integer seq3) [static]
```

Converts the input cosine matrix to a set of euler angles, using the euler sequence provided.

**Parameters**

|               |                                    |
|---------------|------------------------------------|
| <i>cosMat</i> | the input cosine matrix.           |
| <i>seq1</i>   | first entry of the euler sequence  |
| <i>seq2</i>   | second entry of the euler sequence |
| <i>seq3</i>   | third entry of the euler sequence  |

**Returns**

the euler angles representation (radians) of the input attitude.

**8.7.2.11 ToMRPs()**

```
Rvector3 AttitudeConversionUtility::ToMRPs (
 const Rvector & quat1) [static]
```

Converts the input quaternion vector into the Modified Rodriguez Parameters. Note that we are now using the CCSDS definition of quaternions where qc = q4.

**Parameters**

|              |                      |
|--------------|----------------------|
| <i>quat1</i> | the input quaternion |
|--------------|----------------------|

**Returns**

the MRP representation of the input attitude.

**8.7.2.12 ToQuaternion()** [1/3]

```
Rvector AttitudeConversionUtility::ToQuaternion (
 const Rvector3 & eulerAngles,
 Integer seq1,
 Integer seq2,
 Integer seq3) [static]
```

Converts the input set of euler angles to a quaternion, using the euler sequence provided.

**Parameters**

|                    |                                    |
|--------------------|------------------------------------|
| <i>eulerAngles</i> | the input euler angles (radians)   |
| <i>seq1</i>        | first entry of the euler sequence  |
| <i>seq2</i>        | second entry of the euler sequence |
| <i>seq3</i>        | third entry of the euler sequence  |

**Returns**

the quaternion representation of the input attitude.



## 8.7.2.13 ToQuaternion() [2/3]

```
Rvector AttitudeConversionUtility::ToQuaternion (
 const Rmatrix33 & cosMat) [static]
```

Converts the input cosine matrix to a quaternion.

## Parameters

|               |                          |
|---------------|--------------------------|
| <i>cosMat</i> | the input cosine matrix. |
|---------------|--------------------------|

## Returns

the quaternion representation of the input attitude.

## 8.7.2.14 ToQuaternion() [3/3]

```
Rvector AttitudeConversionUtility::ToQuaternion (
 const Rvector3 & MRPs) [static]
```

Converts the input Modified Rodriguez Parameters to a quaternion vector. Note that we are now using the CCSDS definition of quaternions where  $q_c = q_4$ .

## Parameters

|             |                       |
|-------------|-----------------------|
| <i>MRPs</i> | the input MRP vector. |
|-------------|-----------------------|

## Returns

the quaternion representation of the input attitude.

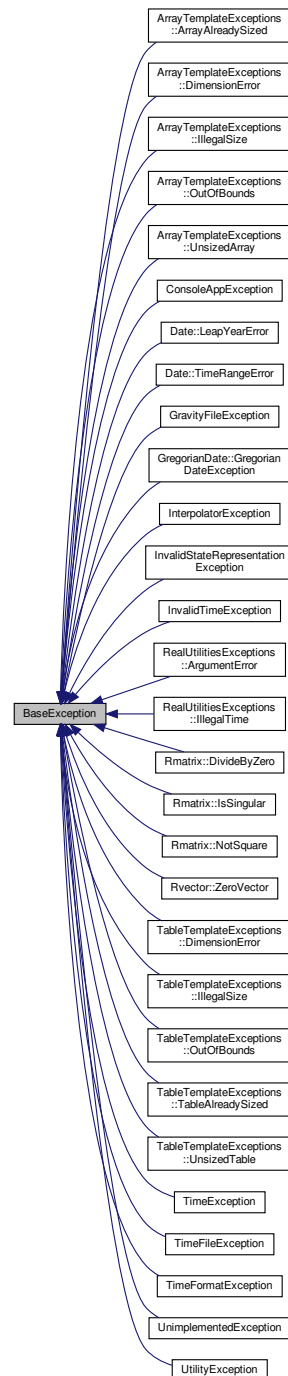
The documentation for this class was generated from the following files:

- GMATsrc/util/[AttitudeConversionUtility.hpp](#)
- GMATsrc/util/[AttitudeConversionUtility.cpp](#)

## 8.8 BaseException Class Reference

```
#include <BaseException.hpp>
```

Inheritance diagram for BaseException:



## Public Member Functions

- virtual std::string [GetFullMessage](#) () const
- virtual std::string [GetDetails](#) () const
- virtual bool [IsFatal](#) () const
- virtual void [SetMessage](#) (const std::string &message)
- virtual void [SetDetails](#) (const std::string &details)

- virtual void [SetFatal](#) (bool fatal)
- virtual void [SetDetails](#) (const char \*details,...)
- virtual [Gmat::MessageType](#) [GetMessageType](#) ()
- virtual void [SetMessageType](#) ([Gmat::MessageType](#) mt)
- const [BaseException](#) & [operator=](#) (const std::string &newMessage)

### Static Public Attributes

- static const int [MAX\\_MESSAGE\\_LENGTH](#) = 3000

### Protected Member Functions

- [BaseException](#) (const std::string &message="", const std::string &details="", [Gmat::MessageType](#) mt=[Gmat::GENERAL\\_](#))
- [BaseException](#) (const [BaseException](#) &be)
- virtual [~BaseException](#) ()
- const [BaseException](#) & [operator=](#) (const [BaseException](#) &be)

#### 8.8.1 Detailed Description

This class provides base exception class, from which all GMAT exceptions must be derived.

#### 8.8.2 Constructor & Destructor Documentation

##### 8.8.2.1 [BaseException\(\)](#) [1/2]

```
BaseException::BaseException (
 const std::string & message = "",
 const std::string & details = "",
 Gmat::MessageType mt = Gmat::GENERAL_) [protected]
```

##### 8.8.2.2 [BaseException\(\)](#) [2/2]

```
BaseException::BaseException (
 const BaseException & be) [protected]
```

##### 8.8.2.3 [~BaseException\(\)](#)

```
BaseException::~~BaseException () [protected], [virtual]
```

## 8.8.3 Member Function Documentation

### 8.8.3.1 GetDetails()

```
std::string BaseException::GetDetails () const [virtual]
```

### 8.8.3.2 GetFullMessage()

```
std::string BaseException::GetFullMessage () const [virtual]
```

Exception class used by the GmatBase base class.

### 8.8.3.3 GetMessageType()

```
Gmat::MessageType BaseException::GetMessageType () [virtual]
```

### 8.8.3.4 IsFatal()

```
bool BaseException::IsFatal () const [virtual]
```

### 8.8.3.5 operator=() [1/2]

```
const BaseException & BaseException::operator= (
 const std::string & newMessage)
```

### 8.8.3.6 operator=() [2/2]

```
const BaseException & BaseException::operator= (
 const BaseException & be) [protected]
```

### 8.8.3.7 SetDetails() [1/2]

```
void BaseException::SetDetails (
 const std::string & details) [virtual]
```

#### 8.8.3.8 SetDetails() [2/2]

```
void BaseException::SetDetails (
 const char * details,
 ...) [virtual]
```

constructor taking variable arguments

#### 8.8.3.9 SetFatal()

```
void BaseException::SetFatal (
 bool fatal) [virtual]
```

#### 8.8.3.10 SetMessage()

```
void BaseException::SetMessage (
 const std::string & message) [virtual]
```

#### 8.8.3.11 SetMessageType()

```
void BaseException::SetMessageType (
 Gmat::MessageType mt) [virtual]
```

### 8.8.4 Member Data Documentation

#### 8.8.4.1 MAX\_MESSAGE\_LENGTH

```
const int BaseException::MAX_MESSAGE_LENGTH = 3000 [static]
```

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

- GMATsrc/util/[BaseException.hpp](#)
- GMATsrc/util/[BaseException.cpp](#)

## 8.9 GmatTimeUtil::CalDate Class Reference

```
#include <TimeTypes.hpp>
```

## Public Member Functions

- [CalDate](#) ([YearNumber](#) y, [MonthOfYear](#) mo, [DayOfMonth](#) d, [HourOfDay](#) h, [MinuteOfHour](#) m, [Real](#) s)
- [CalDate](#) ()

## Public Attributes

- [YearNumber](#) year
- [MonthOfYear](#) month
- [DayOfMonth](#) day
- [HourOfDay](#) hour
- [MinuteOfHour](#) minute
- [Real](#) second

## 8.9.1 Constructor & Destructor Documentation

### 8.9.1.1 [CalDate\(\)](#) [1/2]

```
GmatTimeUtil::CalDate::CalDate (
 YearNumber y,
 MonthOfYear mo,
 DayOfMonth d,
 HourOfDay h,
 MinuteOfHour m,
 Real s) [inline]
```

### 8.9.1.2 [CalDate\(\)](#) [2/2]

```
GmatTimeUtil::CalDate::CalDate () [inline]
```

## 8.9.2 Member Data Documentation

### 8.9.2.1 [day](#)

[DayOfMonth](#) GmatTimeUtil::CalDate::day

### 8.9.2.2 hour

[HourOfDay](#) GmatTimeUtil::CalDate::hour

### 8.9.2.3 minute

[MinuteOfHour](#) GmatTimeUtil::CalDate::minute

### 8.9.2.4 month

[MonthOfYear](#) GmatTimeUtil::CalDate::month

### 8.9.2.5 second

[Real](#) GmatTimeUtil::CalDate::second

### 8.9.2.6 year

[YearNumber](#) GmatTimeUtil::CalDate::year

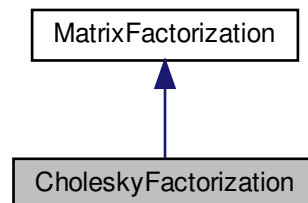
The documentation for this class was generated from the following file:

- GMATsrc/util/[TimeTypes.hpp](#)

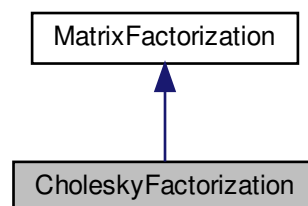
## 8.10 CholeskyFactorization Class Reference

```
#include <CholeskyFactorization.hpp>
```

Inheritance diagram for CholeskyFactorization:



Collaboration diagram for CholeskyFactorization:



### Public Member Functions

- [CholeskyFactorization](#) ()
- [CholeskyFactorization](#) (const [CholeskyFactorization](#) &choleskyfactorization)
- [~CholeskyFactorization](#) ()
- [CholeskyFactorization](#) & [operator=](#) (const [CholeskyFactorization](#) &choleskyfactorization)
- virtual void [Factor](#) (const [Rmatrix](#) inputMatrix, [Rmatrix](#) &R, [Rmatrix](#) &blankMatrix)
- virtual void [Invert](#) ([Rmatrix](#) &inputMatrix)
- virtual [Integer](#) [Invert](#) ([Real](#) \*sum1, [Integer](#) array\_size)

### Additional Inherited Members

#### 8.10.1 Detailed Description

Declares [CholeskyFactorization](#) class.



## 8.10.2 Constructor & Destructor Documentation

### 8.10.2.1 CholeskyFactorization() [1/2]

```
CholeskyFactorization::CholeskyFactorization ()
```

Declares [CholeskyFactorization](#) class. Constructor

### 8.10.2.2 CholeskyFactorization() [2/2]

```
CholeskyFactorization::CholeskyFactorization (
 const CholeskyFactorization & choleskyfactorization)
```

Copy constructor

### 8.10.2.3 ~CholeskyFactorization()

```
CholeskyFactorization::~~CholeskyFactorization ()
```

Destructor

## 8.10.3 Member Function Documentation

### 8.10.3.1 Factor()

```
void CholeskyFactorization::Factor (
 const Rmatrix inputMatrix,
 Rmatrix & R,
 Rmatrix & blankMatrix) [virtual]
```

Matrix factorization routine using Cholesky decomposition

#### Parameters

|                    |                                                                                                                                |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------|
| <i>inputMatrix</i> | The matrix to be factored, packed in upper triangular form                                                                     |
| <i>R</i>           | The matrix the factored result will be stored in                                                                               |
| <i>blankMatrix</i> | This is just to have the required third argument for the function due to class structure, this matrix is not used for anything |

Implements [MatrixFactorization](#).

### 8.10.3.2 Invert() [1/2]

```
void CholeskyFactorization::Invert (
 Rmatrix & inputMatrix) [virtual]
```

Matrix inversion routine using Cholesky decomposition

This method is a port of the inversion code from GEODYN, as ported by Angel Wang of Thinking Systems and then integrated into GMAT by D. Conway.

#### Parameters

|                    |                           |
|--------------------|---------------------------|
| <i>inputMatrix</i> | The matrix to be inverted |
|--------------------|---------------------------|

Implements [MatrixFactorization](#).

### 8.10.3.3 Invert() [2/2]

```
Integer CholeskyFactorization::Invert (
 Real * sum1,
 Integer array_size) [virtual]
```

Matrix inversion routine using Cholesky decomposition

This method is a port of the inversion code from GEODYN, as ported by Angel Wang of Thinking Systems and then integrated into GMAT by D. Conway.

#### Parameters

|                   |                                                            |
|-------------------|------------------------------------------------------------|
| <i>sum1</i>       | The matrix to be inverted, packed in upper triangular form |
| <i>array_size</i> | The size of the sum1 array                                 |

#### Returns

0 on success, anything else indicates a problem

### 8.10.3.4 operator=()

```
CholeskyFactorization & CholeskyFactorization::operator= (
 const CholeskyFactorization & choleskyfactorization)
```

Assignment operator

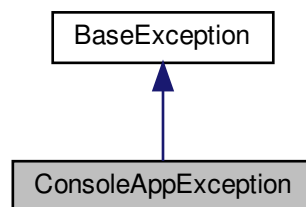
The documentation for this class was generated from the following files:

- GMATsrc/util/matrixoperations/[CholeskyFactorization.hpp](#)
- GMATsrc/util/matrixoperations/[CholeskyFactorization.cpp](#)

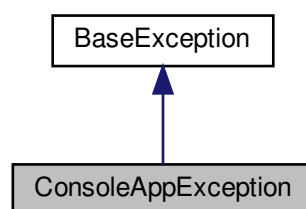
## 8.11 ConsoleAppException Class Reference

```
#include <ConsoleAppException.hpp>
```

Inheritance diagram for ConsoleAppException:



Collaboration diagram for ConsoleAppException:



### Public Member Functions

- [ConsoleAppException](#) (std::string details)
- [~ConsoleAppException](#) ()

### Additional Inherited Members

#### 8.11.1 Detailed Description

Insert descriptive text here.

#### Note

Any notes here. Class used to report exceptions to the console based driver for GMAT

## 8.11.2 Constructor & Destructor Documentation

### 8.11.2.1 ConsoleAppException()

```
ConsoleAppException::ConsoleAppException (
 std::string details)
```

Insert descriptive text here.

#### Note

Any notes here.

### 8.11.2.2 ~ConsoleAppException()

```
ConsoleAppException::~~ConsoleAppException ()
```

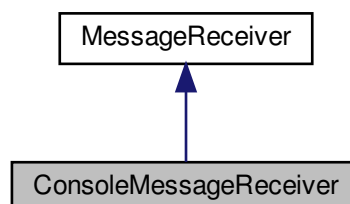
The documentation for this class was generated from the following files:

- GMATsrc/console/[ConsoleAppException.hpp](#)
- GMATsrc/console/[ConsoleAppException.cpp](#)

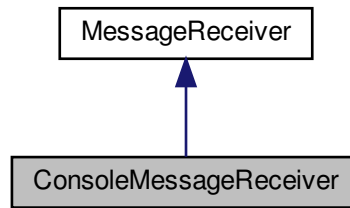
## 8.12 ConsoleMessageReceiver Class Reference

```
#include <ConsoleMessageReceiver.hpp>
```

Inheritance diagram for ConsoleMessageReceiver:



Collaboration diagram for ConsoleMessageReceiver:



## Public Member Functions

- virtual void [ShowMessage](#) (const std::string &msg)
- virtual void [ShowMessage](#) (const char \*msg,...)
- virtual void [PopupMessage](#) ([Gmat::MessageType](#) msgType, const std::string &msg)
- virtual void [PopupMessage](#) ([Gmat::MessageType](#) msgType, const char \*msg,...)
- virtual std::string [GetLogFileName](#) ()
- virtual bool [GetLogEnable](#) ()
- virtual void [SetLogEnable](#) (bool flag)
- virtual void [SetLogPath](#) (const std::string &pathname, bool append=false)
- virtual void [SetLogFile](#) (const std::string &filename)
- virtual void [LogMessage](#) (const std::string &msg)
- virtual void [LogMessage](#) (const char \*msg,...)
- virtual void [ClearMessage](#) ()
- virtual void [OpenLogFile](#) (const std::string &filename, bool append=false)
- virtual void [CloseLogFile](#) ()
- virtual std::string [GetMessage](#) ()
- virtual void [PutMessage](#) (const std::string &msg)
- virtual void [ClearMessageQueue](#) ()

## Static Public Member Functions

- static [ConsoleMessageReceiver](#) \* [Instance](#) ()

## Additional Inherited Members

### 8.12.1 Detailed Description

Declares operations on messages for the Console app. [ConsoleMessageReceiver](#) implements the methods to present messages to the user on the console.

This class is implemented as a singleton.

## 8.12.2 Member Function Documentation

### 8.12.2.1 ClearMessage()

```
void ConsoleMessageReceiver::ClearMessage () [virtual]
```

Clears the message window. This console version does nothing.

Implements [MessageReceiver](#).

### 8.12.2.2 ClearMessageQueue()

```
void ConsoleMessageReceiver::ClearMessageQueue () [virtual]
```

Tells the [MessageReceiver](#) to clear the message queue.

Implements [MessageReceiver](#).

### 8.12.2.3 CloseLogFile()

```
void ConsoleMessageReceiver::CloseLogFile () [virtual]
```

Closes the log file.

### 8.12.2.4 GetLogEnable()

```
bool ConsoleMessageReceiver::GetLogEnable () [virtual]
```

returns if logging is on or off.

Implements [MessageReceiver](#).

### 8.12.2.5 GetLogFileName()

```
std::string ConsoleMessageReceiver::GetLogFileName () [virtual]
```

Retrieves the fully qualified name of the log file.

#### Returns

The name of the log file, including path information.

Implements [MessageReceiver](#).

### 8.12.2.6 GetMessage()

```
std::string ConsoleMessageReceiver::GetMessage () [virtual]
```

Pops the messages off the message queue and concatenates them together.

#### Returns

The concatenated messages.

Implements [MessageReceiver](#).

### 8.12.2.7 Instance()

```
ConsoleMessageReceiver * ConsoleMessageReceiver::Instance () [static]
```

Singleton accessor method

This method creates the [ConsoleMessageReceiver](#) singleton if it has not been constructed, and returns the singleton instance.

#### Returns

The [ConsoleMessageReceiver](#) instance.

### 8.12.2.8 LogMessage() [1/2]

```
void ConsoleMessageReceiver::LogMessage (
 const std::string & msg) [virtual]
```

Logs the message to the log file.

This method displays the input message on the console and writes it to the log file.

#### Parameters

|            |              |
|------------|--------------|
| <i>msg</i> | The message. |
|------------|--------------|

Implements [MessageReceiver](#).

### 8.12.2.9 LogMessage() [2/2]

```
void ConsoleMessageReceiver::LogMessage (
 const char * msg,
 ...) [virtual]
```

Logs a variable argument formatted message to the log file.

This method displays the input message on the console and writes it to the log file.

#### Parameters

|            |                                                                             |
|------------|-----------------------------------------------------------------------------|
| <i>msg</i> | The message, possibly including markers for variable argument substitution. |
| ...        | The optional list of parameters that are inserted into the msg string.      |

Implements [MessageReceiver](#).

#### 8.12.2.10 OpenLogFile()

```
void ConsoleMessageReceiver::OpenLogFile (
 const std::string & filename,
 bool append = false) [virtual]
```

#### 8.12.2.11 PopupMessage() [1/2]

```
void ConsoleMessageReceiver::PopupMessage (
 Gmat::MessageType msgType,
 const std::string & msg) [virtual]
```

Pops up a message in a message box.

This method logs informational messages directed at pop-up message boxes. The Console application does not support pop-ups, so the message cannot be shown as a pop-up.

This method calls the variable argument version of the method to perform the actual logging.

#### Parameters

|                |                                                                                                                                          |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <i>msgType</i> | The type of message that is displayed, selected from the set {ERROR_, WARNING_, INFO_} enumerated in the <a href="#">Gmat</a> namespace. |
| <i>msg</i>     | The message.                                                                                                                             |

Implements [MessageReceiver](#).

#### 8.12.2.12 PopupMessage() [2/2]

```
void ConsoleMessageReceiver::PopupMessage (
 Gmat::MessageType msgType,
```



```
const char * msg,
...) [virtual]
```

Pops up a message in a message box.

This method logs informational messages directed at pop-up message boxes. The Console application does not support pop-ups, so the message cannot be shown as a pop-up.

#### Parameters

|                |                                                                                                                                          |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <i>msgType</i> | The type of message that is displayed, selected from the set {ERROR_, WARNING_, INFO_} enumerated in the <a href="#">Gmat</a> namespace. |
| <i>msg</i>     | The message, possibly including markers for variable argument substitution.                                                              |
| ...            | The optional list of parameters that are inserted into the msg string.                                                                   |

Implements [MessageReceiver](#).

#### 8.12.2.13 PutMessage()

```
void ConsoleMessageReceiver::PutMessage (
 const std::string & msg) [virtual]
```

Push the message into queue

Implements [MessageReceiver](#).

#### 8.12.2.14 SetLogEnable()

```
void ConsoleMessageReceiver::SetLogEnable (
 bool flag) [virtual]
```

Turns logging on or off.

#### Parameters

|             |                                                                                                       |
|-------------|-------------------------------------------------------------------------------------------------------|
| <i>flag</i> | The new logging state – true enables logging, and false disables it. The logging state is idempotent. |
|-------------|-------------------------------------------------------------------------------------------------------|

Implements [MessageReceiver](#).

#### 8.12.2.15 SetLogFile()

```
void ConsoleMessageReceiver::SetLogFile (
 const std::string & filename) [virtual]
```

Implements [MessageReceiver](#).

#### 8.12.2.16 SetLogPath()

```
void ConsoleMessageReceiver::SetLogPath (
 const std::string & pathname,
 bool append = false) [virtual]
```

Implements [MessageReceiver](#).

#### 8.12.2.17 ShowMessage() [1/2]

```
void ConsoleMessageReceiver::ShowMessage (
 const std::string & msg) [virtual]
```

Displays a message passed in as an std::string.

This method sends the message to the user's console and to the log file by calling the variable argument method, ShowMessage(const char \*msg, ...).

##### Parameters

|                  |                                |
|------------------|--------------------------------|
| <i>msgString</i> | The message that is displayed. |
|------------------|--------------------------------|

Implements [MessageReceiver](#).

#### 8.12.2.18 ShowMessage() [2/2]

```
void ConsoleMessageReceiver::ShowMessage (
 const char * msg,
 ...) [virtual]
```

Displays a message passed in as a char\* and a variable argument list.

##### Parameters

|            |                                                                             |
|------------|-----------------------------------------------------------------------------|
| <i>msg</i> | The message, possibly including markers for variable argument substitution. |
| <i>...</i> | The optional list of parameters that are inserted into the msg string.      |

Implements [MessageReceiver](#).

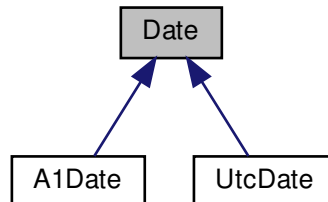
The documentation for this class was generated from the following files:

- GMATsrc/console/[ConsoleMessageReceiver.hpp](#)
- GMATsrc/console/[ConsoleMessageReceiver.cpp](#)

## 8.13 Date Class Reference

```
#include <Date.hpp>
```

Inheritance diagram for Date:



### Classes

- class [LeapYearError](#)
- class [TimeRangeError](#)

### Public Member Functions

- [Integer GetYear \(\)](#) const
- [Integer GetMonth \(\)](#) const
- [Integer GetDay \(\)](#) const
- [Real GetSecondsOfDay \(\)](#) const
- [Integer GetHour \(\)](#) const
- [Integer GetMinute \(\)](#) const
- [Real GetSecond \(\)](#) const
- [GmatTimeConstants::DayName GetDayName \(\)](#) const
- [Integer GetDaysPerMonth \(\)](#) const
- [GmatTimeConstants::MonthName GetMonthName \(\)](#) const
- [Real ToPackedCalendarReal \(\)](#) const
- [Real ToPackedYYMMDD \(\)](#) const
- [Real ToPackedHHMMSS \(\)](#) const
- [Real ToDayOfYear \(\)](#) const
- [std::string & ToPackedCalendarString \(\)](#)
- [void ToYearDOYHourMinSec \(Integer &year, Integer &dayOfYear, Integer &hour, Integer &minute, Real &second\)](#) const
- [void ToYearMonthDayHourMinSec \(Integer &year, Integer &month, Integer &day, Integer &hour, Integer &minute, Real &second\)](#) const
- [void ToYearMonthDayHourMinSec \(Real &year, Real &month, Real &day, Real &hour, Real &minute, Real &second\)](#) const
- [void ToYearMonthDayHourMinSec \(Real &ymd, Real &hms\)](#) const
- [bool IsValid \(\)](#) const
- [Integer GetNumData \(\)](#) const
- [const std::string \\* GetDataDescriptions \(\)](#) const
- [std::string \\* ToValueStrings \(\)](#)

## Protected Member Functions

- [Date](#) ()
- [Date](#) ([Integer](#) year, [Integer](#) month, [Integer](#) day, [Integer](#) hour, [Integer](#) minute, [Real](#) second)
- [Date](#) ([Integer](#) year, [Integer](#) dayOfYear, [Integer](#) hour, [Integer](#) minute, [Real](#) second)
- [Date](#) ([Integer](#) year, [Integer](#) month, [Integer](#) day, [Real](#) secondsOfDay)
- [Date](#) (const [GmatTimeUtil::CalDate](#) &date)
- [Date](#) (const std::string &time)
- [Date](#) (const [Date](#) &date)
- [~Date](#) ()
- bool [operator>](#) (const [Date](#) &date) const
- bool [operator<](#) (const [Date](#) &date) const

## Protected Attributes

- [Integer](#) yearD
- [Integer](#) monthD
- [Integer](#) dayD
- [Real](#) secondsOfDayD
- std::string mPackedString
- std::string stringValues [[NUM\\_DATA](#)]

## Static Protected Attributes

- static const [Integer](#) [NUM\\_DATA](#) = 6
- static const std::string [DATA\\_DESCRIPTIONS](#) [[NUM\\_DATA](#)]

### 8.13.1 Detailed Description

This class is abstract base class which provides conversions among various ways of representing calendar dates and times.

### 8.13.2 Constructor & Destructor Documentation

#### 8.13.2.1 [Date](#)() [1/7]

```
Date::Date () [protected]
```

### 8.13.2.2 Date() [2/7]

```
Date::Date (
 Integer year,
 Integer month,
 Integer day,
 Integer hour,
 Integer minute,
 Real second) [protected]
```

### 8.13.2.3 Date() [3/7]

```
Date::Date (
 Integer year,
 Integer dayOfYear,
 Integer hour,
 Integer minute,
 Real second) [protected]
```

### 8.13.2.4 Date() [4/7]

```
Date::Date (
 Integer year,
 Integer month,
 Integer day,
 Real secondsOfDay) [protected]
```

### 8.13.2.5 Date() [5/7]

```
Date::Date (
 const GmatTimeUtil::CalDate & date) [protected]
```

### 8.13.2.6 Date() [6/7]

```
Date::Date (
 const std::string & time) [protected]
```

#### Parameters

|        |                                             |
|--------|---------------------------------------------|
| <time> | time in string form of "YYYYMMDD.hhmmssnnn" |
|--------|---------------------------------------------|

#### 8.13.2.7 Date() [7/7]

```
Date::Date (
 const Date & date) [protected]
```

#### 8.13.2.8 ~Date()

```
Date::~Date () [protected]
```

### 8.13.3 Member Function Documentation

#### 8.13.3.1 GetDataDescriptions()

```
const std::string * Date::GetDataDescriptions () const
```

#### 8.13.3.2 GetDay()

```
Integer Date::GetDay () const
```

#### 8.13.3.3 GetDayName()

```
GmatTimeConstants::DayName Date::GetDayName () const
```

#### 8.13.3.4 GetDaysPerMonth()

```
Integer Date::GetDaysPerMonth () const
```

#### 8.13.3.5 GetHour()

```
Integer Date::GetHour () const
```

#### 8.13.3.6 GetMinute()

```
Integer Date::GetMinute () const
```

#### 8.13.3.7 GetMonth()

```
Integer Date::GetMonth () const
```

#### 8.13.3.8 GetMonthName()

```
GmatTimeConstants::MonthName Date::GetMonthName () const
```

#### 8.13.3.9 GetNumData()

```
Integer Date::GetNumData () const
```

#### 8.13.3.10 GetSecond()

```
Real Date::GetSecond () const
```

#### 8.13.3.11 GetSecondsOfDay()

```
Real Date::GetSecondsOfDay () const
```

#### 8.13.3.12 GetYear()

```
Integer Date::GetYear () const
```

#### 8.13.3.13 IsValid()

```
bool Date::IsValid () const
```

**8.13.3.14 operator<()**

```
bool Date::operator< (
 const Date & date) const [protected]
```

Comparison operator <

**8.13.3.15 operator>()**

```
bool Date::operator> (
 const Date & date) const [protected]
```

Comparison operator >

**8.13.3.16 ToDayOfYear()**

```
Real Date::ToDayOfYear () const
```

**8.13.3.17 ToPackedCalendarReal()**

```
Real Date::ToPackedCalendarReal () const
```

**8.13.3.18 ToPackedCalendarString()**

```
std::string & Date::ToPackedCalendarString ()
```

**8.13.3.19 ToPackedHHMMSS()**

```
Real Date::ToPackedHHMMSS () const
```

**8.13.3.20 ToPackedYYMMDD()**

```
Real Date::ToPackedYYMMDD () const
```



#### 8.13.3.21 ToValueStrings()

```
std::string * Date::ToValueStrings ()
```

#### 8.13.3.22 ToYearDOYHourMinSec()

```
void Date::ToYearDOYHourMinSec (
 Integer & year,
 Integer & dayOfYear,
 Integer & hour,
 Integer & minute,
 Real & second) const
```

#### 8.13.3.23 ToYearMonthDayHourMinSec() [1/3]

```
void Date::ToYearMonthDayHourMinSec (
 Integer & year,
 Integer & month,
 Integer & day,
 Integer & hour,
 Integer & minute,
 Real & second) const
```

#### 8.13.3.24 ToYearMonthDayHourMinSec() [2/3]

```
void Date::ToYearMonthDayHourMinSec (
 Real & year,
 Real & month,
 Real & day,
 Real & hour,
 Real & minute,
 Real & second) const
```

#### 8.13.3.25 ToYearMonthDayHourMinSec() [3/3]

```
void Date::ToYearMonthDayHourMinSec (
 Real & ymd,
 Real & hms) const
```

Returns time in YYYYMMDD.0 and HHMMSS.mmm format

### 8.13.4 Member Data Documentation

#### 8.13.4.1 DATA\_DESCRIPTIONS

```
const std::string Date::DATA_DESCRIPTIONS [static], [protected]
```

**Initial value:**

```
=
{
 "Year", "Month", "Day", "Hour", "Minute", "Second"
}
```

This class is abstract base class which provides conversions among various ways of representing calendar dates and times.

#### 8.13.4.2 dayD

```
Integer Date::dayD [protected]
```

#### 8.13.4.3 monthD

```
Integer Date::monthD [protected]
```

#### 8.13.4.4 mPackedString

```
std::string Date::mPackedString [protected]
```

#### 8.13.4.5 NUM\_DATA

```
const Integer Date::NUM_DATA = 6 [static], [protected]
```

#### 8.13.4.6 secondsOfDayD

```
Real Date::secondsOfDayD [protected]
```

## 8.13.4.7 stringValues

```
std::string Date::stringValues[NUM_DATA] [protected]
```

## 8.13.4.8 yearD

```
Integer Date::yearD [protected]
```

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

- [GMATsrc/util/Date.hpp](#)
- [GMATsrc/util/Date.cpp](#)

## 8.14 DateUtil Class Reference

```
#include <DateUtil.hpp>
```

## Static Public Member Functions

- static [Integer](#) [JulianDay](#) ([YearNumber](#) year, [MonthOfYear](#) month, [DayOfMonth](#) day)
- static std::string [FormatGregorian](#) ([YearNumber](#) year, [MonthOfYear](#) month, [DayOfMonth](#) day, [Integer](#) hour, [Integer](#) minute, [Real](#) second)
- static bool [IsValidGregorian](#) (const std::string &str, bool checkDate=false)

## Static Public Attributes

- static const std::string [EARLIEST\\_VALID\\_GREGORIAN](#) = "04 Oct 1957 12:00:00.000"
- static const std::string [LATEST\\_VALID\\_GREGORIAN](#) = "28 Feb 2100 00:00:00.000"
- static const std::string [EARLIEST\\_VALID\\_MJD](#) = "6116.00"
- static const std::string [LATEST\\_VALID\\_MJD](#) = "58127.5"
- static const [Real](#) [EARLIEST\\_VALID\\_MJD\\_VALUE](#) = 6116.00
- static const [Real](#) [LATEST\\_VALID\\_MJD\\_VALUE](#) = 58127.5

## Static Protected Attributes

- static const [Integer](#) [MIN\\_YEAR](#) = 1957
- static const [Integer](#) [MIN\\_MONTH](#) = 10
- static const [Integer](#) [MIN\\_DAY](#) = 4
- static const [Integer](#) [MIN\\_HOUR](#) = 12
- static const [Integer](#) [MIN\\_MINUTE](#) = 0
- static const [Real](#) [MIN\\_SEC](#) = 0.000
- static const [Integer](#) [MAX\\_YEAR](#) = 2100
- static const [Integer](#) [MAX\\_MONTH](#) = 2
- static const [Integer](#) [MAX\\_DAY](#) = 28
- static const [Integer](#) [MAX\\_HOUR](#) = 0
- static const [Integer](#) [MAX\\_MINUTE](#) = 0
- static const [Real](#) [MAX\\_SEC](#) = 0.000

## Friends

- [Real](#) [GMATUTIL\\_API](#) [JulianDate](#) ([YearNumber](#) year, [MonthOfYear](#) month, [DayOfMonth](#) day, [Integer](#) hour, [Integer](#) minute, [Real](#) second)
- [Real](#) [GMATUTIL\\_API](#) [ModifiedJulianDate](#) ([YearNumber](#) year, [MonthOfYear](#) month, [DayOfMonth](#) day, [Integer](#) hour, [Integer](#) minute, [Real](#) second, [Real](#) refEpochJD)
- [GmatTime](#) [GMATUTIL\\_API](#) [ModifiedJulianDateGT](#) ([YearNumber](#) year, [MonthOfYear](#) month, [DayOfMonth](#) day, [Integer](#) hour, [Integer](#) minute, [Real](#) second, [Real](#) refEpochJD)
- void [GMATUTIL\\_API](#) [UnpackDate](#) ([Real](#) packedDate, [Integer](#) &year, [Integer](#) &month, [Integer](#) &day)
- void [GMATUTIL\\_API](#) [UnpackDateWithDOY](#) ([Real](#) packedDate, [Integer](#) &year, [Integer](#) &day)
- void [GMATUTIL\\_API](#) [UnpackTime](#) ([Real](#) packedTime, [Integer](#) &hour, [Integer](#) &minute, [Real](#) &second)
- void [GMATUTIL\\_API](#) [ToMonthDayFromYearDOY](#) ([Integer](#) year, [Integer](#) dayOfYear, [Integer](#) &month, [Integer](#) &day)
- [Integer](#) [GMATUTIL\\_API](#) [ToDOYFromYearMonthDay](#) ([Integer](#) year, [Integer](#) month, [Integer](#) day)
- [Real](#) [GMATUTIL\\_API](#) [ToSecondsOfDayFromHMS](#) ([Integer](#) hour, [Integer](#) minute, [Real](#) second)
- void [GMATUTIL\\_API](#) [ToHMSFromSecondsOfDay](#) ([Real](#) secsOfDay, [Integer](#) &hour, [Integer](#) &minute, [Real](#) &second)
- bool [GMATUTIL\\_API](#) [IsValidTime](#) ([Integer](#) year, [Integer](#) month, [Integer](#) day, [Integer](#) hour, [Integer](#) minute, [Real](#) second)
- bool [GMATUTIL\\_API](#) [IsLeapYear](#) ([Integer](#) year)

### 8.14.1 Detailed Description

This class provides conversions among various ways of representing calendar dates and times.

### 8.14.2 Member Function Documentation

#### 8.14.2.1 FormatGregorian()

```
static std::string DateUtil::FormatGregorian (
 YearNumber year,
 MonthOfYear month,
 DayOfMonth day,
 Integer hour,
 Integer minute,
 Real second) [static]
```

#### 8.14.2.2 IsValidGregorian()

```
bool DateUtil::IsValidGregorian (
 const std::string & str,
 bool checkDate = false) [static]
```

Determines if input date string is valid Gregorian or not. Valid format is dd mmm yyyy hh:mm:ss.mmm. For example, 01 Jan 2000 12:00:00.000

## Parameters

|                  |                                                         |
|------------------|---------------------------------------------------------|
| <i>greg</i>      | input gregorian string                                  |
| <i>checkDate</i> | check for valid date (i.e. occurs after Sputnik launch) |

## Returns

true if time is in valid Gregorian format; otherwise, false

## 8.14.2.3 JulianDay()

```
Integer DateUtil::JulianDay (
 YearNumber year,
 MonthOfYear month,
 DayOfMonth day) [static]
```

## 8.14.3 Friends And Related Function Documentation

## 8.14.3.1 IsLeapYear

```
bool GMATUTIL_API IsLeapYear (
 Integer year) [friend]
```

Friend function.

## 8.14.3.2 IsValidTime

```
bool GMATUTIL_API IsValidTime (
 Integer year,
 Integer month,
 Integer day,
 Integer hour,
 Integer minute,
 Real second) [friend]
```

Friend function.

## 8.14.3.3 JulianDate

```
Real GMATUTIL_API JulianDate (
 YearNumber year,
 MonthOfYear month,
 DayOfMonth day,
 Integer hour,
 Integer minute,
 Real second) [friend]
```

Friend function. Converted from calendar date to Julian [Date](#).

**Parameters**

|          |                                   |
|----------|-----------------------------------|
| <year>   | - year of calendar                |
| <month>  | - month in calendar format        |
| <day>    | - day of month in calendar format |
| <hour>   | - hour of day                     |
| <minute> | - minute of hour                  |
| <second> | - seconds including millisecond   |

**Returns**

Julian date

**Note**

: The algorithm is used in the Vallado book.

**8.14.3.4 ModifiedJulianDate**

```
Real GMATUTIL_API ModifiedJulianDate (
 YearNumber year,
 MonthOfYear month,
 DayOfMonth day,
 Integer hour,
 Integer minute,
 Real second,
 Real refEpochJD) [friend]
```

Friend function. Converted from calendar date to Modified Julian [Date](#).

**Parameters**

|              |                                               |
|--------------|-----------------------------------------------|
| <year>       | - year of calendar                            |
| <month>      | - month in calendar format                    |
| <day>        | - day of month in calendar format             |
| <hour>       | - hour of day                                 |
| <minute>     | - minute of hour                              |
| <second>     | - seconds including millisecond               |
| <refEpochJD> | - reference epoch Julian <a href="#">Date</a> |

**Returns**

Modified Julian date

## 8.14.3.5 ModifiedJulianDateGT

```
GmatTime GMATUTIL_API ModifiedJulianDateGT (
 YearNumber year,
 MonthOfYear month,
 DayOfMonth day,
 Integer hour,
 Integer minute,
 Real second,
 Real refEpochJD) [friend]
```

Friend function. Converted from calendar date to Modified Julian [Date](#).

## Parameters

|              |                                               |
|--------------|-----------------------------------------------|
| <year>       | - year of calendar                            |
| <month>      | - month in calendar format                    |
| <day>        | - day of month in calendar format             |
| <hour>       | - hour of day                                 |
| <minute>     | - minute of hour                              |
| <second>     | - seconds including millisecond               |
| <refEpochJD> | - reference epoch Julian <a href="#">Date</a> |

## Returns

Modified Julian date

## 8.14.3.6 ToDOYFromYearMonthDay

```
Integer GMATUTIL_API ToDOYFromYearMonthDay (
 Integer year,
 Integer month,
 Integer day) [friend]
```

Friend function.

## Note

Year is needed to determine if it is a leap year

## 8.14.3.7 ToHMSFromSecondsOfDay

```
void GMATUTIL_API ToHMSFromSecondsOfDay (
 Real secsOfDay,
 Integer & hour,
 Integer & minute,
 Real & second) [friend]
```

Friend function.

Notes: Seconds are Real to permit fractions; seconds of day constrained to 0.0 .. 86401.0; the last second of a leap second day will be 23:59:60; assumes only 1 leap second per day maximum.

#### 8.14.3.8 ToMonthDayFromYearDOY

```
void GMATUTIL_API ToMonthDayFromYearDOY (
 Integer year,
 Integer dayOfYear,
 Integer & month,
 Integer & day) [friend]
```

Friend function.

##### Note

Year is needed to determine if it is a leap year.

#### 8.14.3.9 ToSecondsOfDayFromHMS

```
Real GMATUTIL_API ToSecondsOfDayFromHMS (
 Integer hour,
 Integer minute,
 Real second) [friend]
```

Friend function.

##### Note

Seconds of day constrained to 0.0..86401.0; assumes only 1 leap second per day maximum.

#### 8.14.3.10 UnpackDate

```
void GMATUTIL_API UnpackDate (
 Real packedDate,
 Integer & year,
 Integer & month,
 Integer & day) [friend]
```

Friend function.

##### Note

Input date in YYYYMMDD



#### 8.14.3.11 UnpackDateWithDOY

```
void GMATUTIL_API UnpackDateWithDOY (
 Real packedDate,
 Integer & year,
 Integer & day) [friend]
```

Friend function.

##### Note

Input date is in YYYYDDD; Day is rounded because of potential floating point representation problem. Do not pre-correct the input by adding 0.5.

#### 8.14.3.12 UnpackTime

```
void GMATUTIL_API UnpackTime (
 Real packedTime,
 Integer & hour,
 Integer & minute,
 Real & second) [friend]
```

Friend function.

##### Note

Input time is in hhmmssnnn; added 20 seconds to the input time, to be subtracted later, to avoid gross errors around minute boundaries.

### 8.14.4 Member Data Documentation

#### 8.14.4.1 EARLIEST\_VALID\_GREGORIAN

```
const std::string DateUtil::EARLIEST_VALID_GREGORIAN = "04 Oct 1957 12:00:00.000" [static]
```

#### 8.14.4.2 EARLIEST\_VALID\_MJD

```
const std::string DateUtil::EARLIEST_VALID_MJD = "6116.00" [static]
```

#### 8.14.4.3 EARLIEST\_VALID\_MJD\_VALUE

```
const Real DateUtil::EARLIEST_VALID_MJD_VALUE = 6116.00 [static]
```

#### 8.14.4.4 LATEST\_VALID\_GREGORIAN

```
const std::string DateUtil::LATEST_VALID_GREGORIAN = "28 Feb 2100 00:00:00.000" [static]
```

#### 8.14.4.5 LATEST\_VALID\_MJD

```
const std::string DateUtil::LATEST_VALID_MJD = "58127.5" [static]
```

#### 8.14.4.6 LATEST\_VALID\_MJD\_VALUE

```
const Real DateUtil::LATEST_VALID_MJD_VALUE = 58127.5 [static]
```

#### 8.14.4.7 MAX\_DAY

```
const Integer DateUtil::MAX_DAY = 28 [static], [protected]
```

#### 8.14.4.8 MAX\_HOUR

```
const Integer DateUtil::MAX_HOUR = 0 [static], [protected]
```

#### 8.14.4.9 MAX\_MINUTE

```
const Integer DateUtil::MAX_MINUTE = 0 [static], [protected]
```

#### 8.14.4.10 MAX\_MONTH

```
const Integer DateUtil::MAX_MONTH = 2 [static], [protected]
```

#### 8.14.4.11 MAX\_SEC

```
const Real DateUtil::MAX_SEC = 0.000 [static], [protected]
```

#### 8.14.4.12 MAX\_YEAR

```
const Integer DateUtil::MAX_YEAR = 2100 [static], [protected]
```

#### 8.14.4.13 MIN\_DAY

```
const Integer DateUtil::MIN_DAY = 4 [static], [protected]
```

#### 8.14.4.14 MIN\_HOUR

```
const Integer DateUtil::MIN_HOUR = 12 [static], [protected]
```

#### 8.14.4.15 MIN\_MINUTE

```
const Integer DateUtil::MIN_MINUTE = 0 [static], [protected]
```

#### 8.14.4.16 MIN\_MONTH

```
const Integer DateUtil::MIN_MONTH = 10 [static], [protected]
```

#### 8.14.4.17 MIN\_SEC

```
const Real DateUtil::MIN_SEC = 0.000 [static], [protected]
```

#### 8.14.4.18 MIN\_YEAR

```
const Integer DateUtil::MIN_YEAR = 1957 [static], [protected]
```

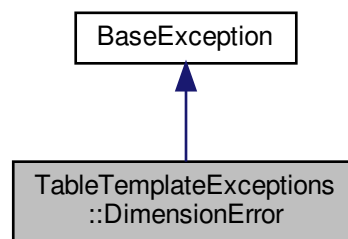
The documentation for this class was generated from the following files:

- GMATsrc/util/[DateUtil.hpp](#)
- GMATsrc/util/[DateUtil.cpp](#)

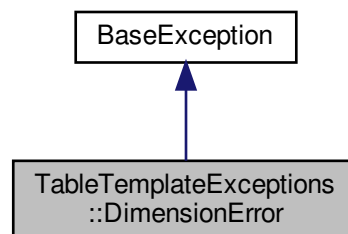
### 8.15 TableTemplateExceptions::DimensionError Class Reference

```
#include <TableTemplate.hpp>
```

Inheritance diagram for TableTemplateExceptions::DimensionError:



Collaboration diagram for TableTemplateExceptions::DimensionError:



#### Public Member Functions

- [DimensionError](#) (const std::string &message="TableTemplate error : dimension error or mismatch.\")

## Additional Inherited Members

### 8.15.1 Constructor & Destructor Documentation

#### 8.15.1.1 DimensionError()

```
TableTemplateExceptions::DimensionError::DimensionError (
 const std::string & message = "TableTemplate error : dimension error or mismatch.\n"
) [inline]
```

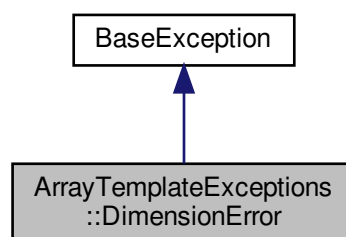
The documentation for this class was generated from the following file:

- GMATsrc/util/[TableTemplate.hpp](#)

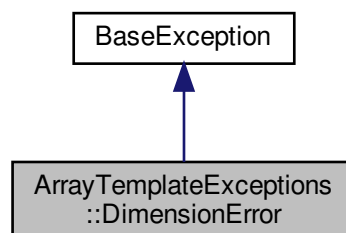
## 8.16 ArrayTemplateExceptions::DimensionError Class Reference

```
#include <ArrayTemplate.hpp>
```

Inheritance diagram for ArrayTemplateExceptions::DimensionError:



Collaboration diagram for ArrayTemplateExceptions::DimensionError:



## Public Member Functions

- [DimensionError](#) (const std::string &message="ArrayTemplate error : dimension error.")

## Additional Inherited Members

### 8.16.1 Constructor & Destructor Documentation

#### 8.16.1.1 DimensionError()

```
ArrayTemplateExceptions::DimensionError::DimensionError (
 const std::string & message = "ArrayTemplate error : dimension error.") [inline]
```

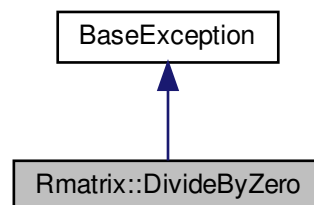
The documentation for this class was generated from the following file:

- GMATsrc/util/[ArrayTemplate.hpp](#)

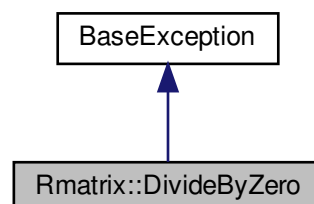
### 8.17 Rmatrix::DivideByZero Class Reference

```
#include <Rmatrix.hpp>
```

Inheritance diagram for Rmatrix::DivideByZero:



Collaboration diagram for Rmatrix::DivideByZero:



## Public Member Functions

- [DivideByZero](#) (const std::string &message="Rmatrix error: attempt to divide by zero\

## Additional Inherited Members

### 8.17.1 Constructor & Destructor Documentation

#### 8.17.1.1 DivideByZero()

```
Rmatrix::DivideByZero::DivideByZero (
 const std::string & message = "Rmatrix error: attempt to divide by zero\n")
[inline]
```

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

- GMATsrc/util/[Rmatrix.hpp](#)

## 8.18 GmatTimeUtil::ElapsedDate Class Reference

```
#include <TimeTypes.hpp>
```

## Public Member Functions

- [ElapsedDate](#) (Integer d, Integer h, Integer m, Real s)
- [ElapsedDate](#) ()

## Public Attributes

- [Integer](#) days
- [Integer](#) hours
- [Integer](#) minutes
- [Real](#) seconds

### 8.18.1 Constructor & Destructor Documentation

#### 8.18.1.1 ElapsedDate() [1/2]

```
GmatTimeUtil::ElapsedDate::ElapsedDate (
 Integer d,
 Integer h,
 Integer m,
 Real s) [inline]
```

#### 8.18.1.2 ElapsedDate() [2/2]

```
GmatTimeUtil::ElapsedDate::ElapsedDate () [inline]
```

### 8.18.2 Member Data Documentation

#### 8.18.2.1 days

```
Integer GmatTimeUtil::ElapsedDate::days
```

#### 8.18.2.2 hours

```
Integer GmatTimeUtil::ElapsedDate::hours
```

#### 8.18.2.3 minutes

```
Integer GmatTimeUtil::ElapsedDate::minutes
```

#### 8.18.2.4 seconds

```
Real GmatTimeUtil::ElapsedDate::seconds
```

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

- GMATsrc/util/[TimeTypes.hpp](#)



## 8.19 ElapsedTime Class Reference

```
#include <ElapsedTime.hpp>
```

### Public Member Functions

- [ElapsedTime](#) (const [Real](#) &secs=0.0, const [Real](#) tol=[GmatRealConstants::REAL\\_EPSILON](#))
- [ElapsedTime](#) (const [ElapsedTime](#) &elapsedTime, const [Real](#) tol=[GmatRealConstants::REAL\\_EPSILON](#))
- [ElapsedTime](#) & [operator=](#) (const [ElapsedTime](#) &right)
- virtual [~ElapsedTime](#) ()
- [ElapsedTime](#) [operator+](#) (const [Real](#) &right) const
- [ElapsedTime](#) [operator-](#) (const [Real](#) &right) const
- const [ElapsedTime](#) & [operator+=](#) (const [Real](#) &right)
- const [ElapsedTime](#) & [operator-=](#) (const [Real](#) &right)
- bool [operator<](#) (const [ElapsedTime](#) &right) const
- bool [operator>](#) (const [ElapsedTime](#) &right) const
- bool [operator==](#) (const [ElapsedTime](#) &right) const
- bool [operator!=](#) (const [ElapsedTime](#) &right) const
- bool [operator>=](#) (const [ElapsedTime](#) &right) const
- bool [operator<=](#) (const [ElapsedTime](#) &right) const
- [Real](#) [Get](#) () const
- void [Set](#) ([Real](#) secs)
- [GmatTimeUtil::ElapsedDate](#) [ToElapsedDate](#) () const
- [Integer](#) [GetNumData](#) () const
- const std::string \* [GetDataDescriptions](#) () const
- std::string \* [ToValueStrings](#) ()

### 8.19.1 Detailed Description

Declares elapsed time in operations. Internal elapsed time is in seconds.

### 8.19.2 Constructor & Destructor Documentation

#### 8.19.2.1 [ElapsedTime\(\)](#) [1/2]

```
ElapsedTime::ElapsedTime (
 const Real & secs = 0.0,
 const Real tol = GmatRealConstants::REAL_EPSILON)
```

#### 8.19.2.2 [ElapsedTime\(\)](#) [2/2]

```
ElapsedTime::ElapsedTime (
 const ElapsedTime & elapsedTime,
 const Real tol = GmatRealConstants::REAL_EPSILON)
```

### 8.19.2.3 ~ElapsedTime()

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

## 8.19.3 Member Function Documentation

### 8.19.3.1 Get()

```
Real ElapsedTime::Get () const
```

### 8.19.3.2 GetDataDescriptions()

```
const std::string * ElapsedTime::GetDataDescriptions () const
```

### 8.19.3.3 GetNumData()

```
Integer ElapsedTime::GetNumData () const
```

### 8.19.3.4 operator"!="()

```
bool ElapsedTime::operator!= (
 const ElapsedTime & right) const
```

### 8.19.3.5 operator+()

```
ElapsedTime ElapsedTime::operator+ (
 const Real & right) const
```

### 8.19.3.6 operator+=()

```
const ElapsedTime & ElapsedTime::operator+= (
 const Real & right)
```

#### 8.19.3.7 operator-()

```
ElapsedTime ElapsedTime::operator- (
 const Real & right) const
```

#### 8.19.3.8 operator-=( )

```
const ElapsedTime & ElapsedTime::operator-= (
 const Real & right)
```

#### 8.19.3.9 operator<()

```
bool ElapsedTime::operator< (
 const ElapsedTime & right) const
```

#### 8.19.3.10 operator<=()

```
bool ElapsedTime::operator<= (
 const ElapsedTime & right) const
```

#### 8.19.3.11 operator=( )

```
ElapsedTime & ElapsedTime::operator= (
 const ElapsedTime & right)
```

#### 8.19.3.12 operator==( )

```
bool ElapsedTime::operator==(
 const ElapsedTime & right) const
```

#### 8.19.3.13 operator>()

```
bool ElapsedTime::operator> (
 const ElapsedTime & right) const
```

#### 8.19.3.14 operator>=()

```
bool ElapsedTime::operator>= (
 const ElapsedTime & right) const
```

#### 8.19.3.15 Set()

```
void ElapsedTime::Set (
 Real secs)
```

#### 8.19.3.16 ToElapsedDate()

```
GmatTimeUtil::ElapsedDate ElapsedTime::ToElapsedDate () const
```

#### 8.19.3.17 ToValueStrings()

```
std::string * ElapsedTime::ToValueStrings ()
```

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

- [GMATsrc/util/ElapsedTime.hpp](#)
- [GMATsrc/util/ElapsedTime.cpp](#)

## 8.20 Element Struct Reference

```
#include <Rvector.hpp>
```

### Public Attributes

- [Real value](#)
- [Integer index](#)

#### 8.20.1 Member Data Documentation

## 8.20.1.1 index

`Integer` `Element::index`

## 8.20.1.2 value

`Real` `Element::value`

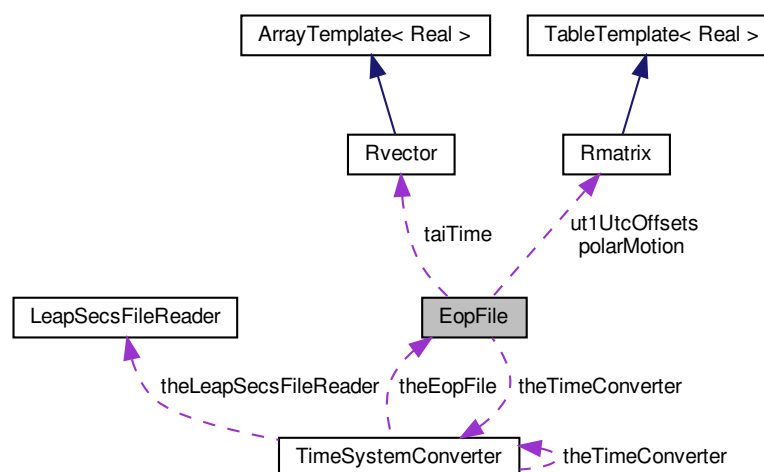
The documentation for this struct was generated from the following file:

- GMATsrc/util/[Rvector.hpp](#)

## 8.21 EopFile Class Reference

```
#include <EopFile.hpp>
```

Collaboration diagram for EopFile:



### Public Member Functions

- `EopFile` (const std::string &fileName="eopc04.62-now", GmatEop::EopFileType eop=GmatEop::EOP\_C04)
- `EopFile` (const `EopFile` &eopF)
- const `EopFile` & `operator=` (const `EopFile` &eopF)
- virtual `~EopFile` ()
- virtual void `Initialize` ()
- virtual void `ResetEopFile` (const std::string &toName, GmatEop::EopFileType toType=GmatEop::EOP\_C04)
- virtual std::string `GetFileName` () const
- virtual `Real` `GetUt1UtcOffset` (const `Real` taiMjd)
- virtual `Rmatrix` `GetPolarMotionData` ()
- virtual bool `GetPolarMotionAndLod` (const `GmatTime` &forUtcMjd, `Real` &xval, `Real` &yval, `Real` &lodval)
- void `GetTimeRange` (`Real` &timeMin, `Real` &timeMax)

## Protected Member Functions

- bool [IsBlank](#) (const char \*aLine)

## Protected Attributes

- [GmatEop::EopFileType](#) eopFType
- [std::string](#) eopFileName
- [Integer](#) tableSz
- [Rmatrix](#) \* polarMotion  
*table of polar motion data : MJD, X, Y, LOD*
- [Rmatrix](#) \* ut1UtcOffsets  
*vector of UT1-UTC offsets : MJD, offset*
- [Rvector](#) \* taiTime
- [Real](#) lastUtcJd
- [Real](#) lastTaiMjd
- [Real](#) lastOffset
- [Integer](#) lastIndex
- bool isInitialized
- [TimeSystemConverter](#) \* theTimeConverter  
*Time converter singleton.*
- [Integer](#) previousIndex

## Static Protected Attributes

- static const [Integer](#) MAX\_TABLE\_SIZE = 50405

## 8.21.1 Constructor & Destructor Documentation

### 8.21.1.1 EopFile() [1/2]

```
EopFile::EopFile (
 const std::string & fileName = "eopc04.62-now",
 GmatEop::EopFileType eop = GmatEop::EOP_C04)
```

Constructs base [EopFile](#) structures used in derived classes (default constructor).

#### Parameters

|                |                |
|----------------|----------------|
| <i>fileNme</i> | EOP file name. |
|----------------|----------------|

### 8.21.1.2 EopFile() [2/2]

```
EopFile::EopFile (
```

```
const EopFile & eopF)
```

Constructs base [EopFile](#) structures, by copying the input instance (copy constructor).

#### Parameters

|             |                                                                     |
|-------------|---------------------------------------------------------------------|
| <i>eopF</i> | <a href="#">EopFile</a> instance to copy to create "this" instance. |
|-------------|---------------------------------------------------------------------|

#### 8.21.1.3 ~EopFile()

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

Destructor.

### 8.21.2 Member Function Documentation

#### 8.21.2.1 GetFileName()

```
std::string EopFile::GetFileName () const [virtual]
```

Returns the name of the EOP file.

#### Returns

name of the EOP file.

#### 8.21.2.2 GetPolarMotionAndLod()

```
bool EopFile::GetPolarMotionAndLod (
 const GmatTime & forUtcMjd,
 Real & xval,
 Real & yval,
 Real & lodval) [virtual]
```

Returns the polar motion data X, Y, and LOD, for the input UTC MJD time.

#### Parameters

|                  |                                              |
|------------------|----------------------------------------------|
| <i>forUtcMjd</i> | time for which to return the data            |
| <i>xval</i>      | return X value of polar motion data (arcsec) |
| <i>yval</i>      | return Y value of polar motion data (arcsec) |
| <i>lodval</i>    | return LOD value (seconds)                   |

### 8.21.2.3 GetPolarMotionData()

```
Rmatrix EopFile::GetPolarMotionData () [virtual]
```

Returns the polar motion data. for each row: mjd, x, y

#### Returns

polar motion data.

### 8.21.2.4 GetTimeRange()

```
void EopFile::GetTimeRange (
 Real & timeMin,
 Real & timeMax)
```

### 8.21.2.5 GetUt1UtcOffset()

```
Real EopFile::GetUt1UtcOffset (
 const Real taiMjd) [virtual]
```

Returns the UT1-UTC offset for the given tai mjd.

#### Parameters

|               |                                             |
|---------------|---------------------------------------------|
| <i>taiMjd</i> | The tai mjd for which to return the offset. |
|---------------|---------------------------------------------|

#### Returns

UT1-UTC offset for the given time; values between table entries are interpolated linearly.

### 8.21.2.6 Initialize()

```
void EopFile::Initialize () [virtual]
```

This method initializes the [EopFile](#) class, by reading the file and storing the UT1-UTC offset and polar motion data.



## 8.21.2.7 IsBlank()

```
bool EopFile::IsBlank (
 const char * aLine) [protected]
```

This method returns true if the string is empty or is all white space.

## Returns

success flag.

## 8.21.2.8 operator=()

```
const EopFile & EopFile::operator= (
 const EopFile & eopF)
```

Assignment operator for [EopFile](#) structures.

## Parameters

|             |                                    |
|-------------|------------------------------------|
| <i>eopF</i> | The original that is being copied. |
|-------------|------------------------------------|

## Returns

Reference to this object

## 8.21.2.9 ResetEopFile()

```
void EopFile::ResetEopFile (
 const std::string & toName,
 GmatEop::EopFileType toType = GmatEop::EOP_C04) [virtual]
```

## 8.21.3 Member Data Documentation

## 8.21.3.1 eopFileName

```
std::string EopFile::eopFileName [protected]
```

### 8.21.3.2 eopFType

```
GmatEop::EopFileType EopFile::eopFType [protected]
```

### 8.21.3.3 isInitialized

```
bool EopFile::isInitialized [protected]
```

### 8.21.3.4 lastIndex

```
Integer EopFile::lastIndex [protected]
```

### 8.21.3.5 lastOffset

```
Real EopFile::lastOffset [protected]
```

### 8.21.3.6 lastTaiMjd

```
Real EopFile::lastTaiMjd [protected]
```

### 8.21.3.7 lastUtcJd

```
Real EopFile::lastUtcJd [protected]
```

### 8.21.3.8 MAX\_TABLE\_SIZE

```
const Integer EopFile::MAX_TABLE_SIZE = 50405 [static], [protected]
```

Implementation of the [EopFile](#) class. This is the code that reads the polar motion information from EOP file (type 14 C04 or type 08 C04). NOTE: reading of Finals file is no longer supported.

### 8.21.3.9 polarMotion

`Rmatrix*` EopFile::polarMotion [protected]

table of polar motion data : MJD, X, Y, LOD

### 8.21.3.10 previousIndex

`Integer` EopFile::previousIndex [protected]

### 8.21.3.11 tableSz

`Integer` EopFile::tableSz [protected]

### 8.21.3.12 taiTime

`Rvector*` EopFile::taiTime [protected]

### 8.21.3.13 theTimeConverter

`TimeSystemConverter*` EopFile::theTimeConverter [protected]

Time converter singleton.

### 8.21.3.14 ut1UtcOffsets

`Rmatrix*` EopFile::ut1UtcOffsets [protected]

vector of UT1-UTC offsets : MJD, offset

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

- GMATsrc/util/[EopFile.hpp](#)
- GMATsrc/util/[EopFile.cpp](#)

## 8.22 ExponentialAtmosphere Class Reference

```
#include <ExponentialAtmosphere.hpp>
```

### Public Member Functions

- [ExponentialAtmosphere](#) (const std::string &name="")
- virtual [~ExponentialAtmosphere](#) ()
- [ExponentialAtmosphere](#) (const [ExponentialAtmosphere](#) &atm)
- [ExponentialAtmosphere](#) & [operator=](#) (const [ExponentialAtmosphere](#) &atm)
- [Real Density](#) ([Real](#) height)
- [Real GetScaleHeight](#) ([Real](#) height)

### Protected Member Functions

- virtual void [SetConstants](#) ()
- [Integer FindBand](#) ([Real](#) height)
- [Real Smooth](#) ([Real](#) height, [Integer](#) index)
- virtual [ExponentialAtmosphere](#) \* [Clone](#) () const

### Protected Attributes

- [Real](#) \* [scaleHeight](#)  
*Table of scale heights,  $H$ .*
- [Real](#) \* [refHeight](#)  
*Table of Reference heights,  $h_0$ .*
- [Real](#) \* [refDensity](#)  
*Table of reference densities,  $\rho_0$ .*
- [Integer altitudeBands](#)  
*Number of altitude bands used in the model.*
- bool [smoothDensity](#)  
*Flag indicating if the altitude is "at" a boundary.*

### 8.22.1 Detailed Description

An exponentially modeled atmosphere based on Vallado, pp 532-534 and Wertz, p 820. The exponential atmosphere model in Vallado (2001) and Wertz (1978).

This code calculates the atmospheric density at a given position based on the altitude of the input state above the spherical Earth. This model does not include an atmospheric bulge due to solar heating. The code will need to be refined to use the oblate Earth once oblateness is added to the code.

The density is given by

$$\rho = \rho_0 e^{-\frac{h_{ellp} - h_0}{H}}$$

where  $\rho_0$  is a reference density, specified at a reference altitude  $h_0$ ,  $h_{ellp}$  is the height of the specified position above the body's ellipsoid, and  $H$  is a scale height, used to scale the other variables in the formula.

Developers and other users can build exponential models for bodies other than the Earth by deriving a class off of this one and overriding the SetConstants method.

**Todo** Replace the spherical Earth model with an oblate Earth model.

**Test** Check to see if the band discontinuities merit smoothing.

## 8.22.2 Constructor & Destructor Documentation

### 8.22.2.1 ExponentialAtmosphere() [1/2]

```
ExponentialAtmosphere::ExponentialAtmosphere (
 const std::string & name = "")
```

Vallado's exponentially modeled atmosphere, with one correction. Default constructor.

#### Parameters

|             |                                      |
|-------------|--------------------------------------|
| <i>name</i> | name of the model (default is blank) |
|-------------|--------------------------------------|

### 8.22.2.2 ~ExponentialAtmosphere()

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

Destructor.

### 8.22.2.3 ExponentialAtmosphere() [2/2]

```
ExponentialAtmosphere::ExponentialAtmosphere (
 const ExponentialAtmosphere & atm)
```

Copy constructor.

#### Parameters

|       |                                                                               |
|-------|-------------------------------------------------------------------------------|
| <atm> | <a href="#">ExponentialAtmosphere</a> object to copy in creating the new one. |
|-------|-------------------------------------------------------------------------------|

## 8.22.3 Member Function Documentation

### 8.22.3.1 Clone()

```
ExponentialAtmosphere * ExponentialAtmosphere::Clone () const [protected], [virtual]
```

Clone the object (inherited from GmatBase).

#### Returns

a clone of "this" object.

### 8.22.3.2 Density()

```
Real ExponentialAtmosphere::Density (
 Real height)
```

Calculates the density given height above the ellipsoid Vallado's method to interpolate the densities.

#### Parameters

|           |                        |
|-----------|------------------------|
| <height>  | Height above ellipsoid |
| <density> | density                |

#### Returns

true on success, throws on failure.

### 8.22.3.3 FindBand()

```
Integer ExponentialAtmosphere::FindBand (
 Real height) [protected]
```

Determines which altitude band the point of interest occupies.

#### Parameters

|          |                                                  |
|----------|--------------------------------------------------|
| <height> | The height above the body's reference ellipsoid. |
|----------|--------------------------------------------------|

#### Returns

The index of the corresponding band.

### 8.22.3.4 GetScaleHeight()

```
Real ExponentialAtmosphere::GetScaleHeight (
 Real height)
```

Calculates scale height for exponential model given height Vallado's method to interpolate the densities.

#### Parameters

|          |                        |
|----------|------------------------|
| <height> | Height above ellipsoid |
| <scale>  | scale height, km       |

**Returns**

true on success, throws on failure.

**8.22.3.5 operator=()**

```
ExponentialAtmosphere & ExponentialAtmosphere::operator= (
 const ExponentialAtmosphere & atm)
```

Assignment operator for the [ExponentialAtmosphere](#) class.

**Parameters**

|       |                                                                                                                         |
|-------|-------------------------------------------------------------------------------------------------------------------------|
| <atm> | the <a href="#">ExponentialAtmosphere</a> object whose data to assign to "this" <a href="#">ExponentialAtmosphere</a> . |
|-------|-------------------------------------------------------------------------------------------------------------------------|

**Returns**

"this" [ExponentialAtmosphere](#) with data of input [ExponentialAtmosphere](#) atm.

**8.22.3.6 SetConstants()**

```
void ExponentialAtmosphere::SetConstants () [protected], [virtual]
```

Builds 3 arrays corresponding to the columns in Vallado's Table 8-4.

Users that want to build other atmosphere models that have the same form as Vallado's (and Wertz's) can derive a class from this one and override this method with their choice of constants.

**Note**

This coefficient was corrected from Vallado's value of 9.158e-12

**8.22.3.7 Smooth()**

```
Real ExponentialAtmosphere::Smooth (
 Real height,
 Integer index) [protected]
```

Smooths discontinuities between the altitude bands.

**Parameters**

|          |                                                  |
|----------|--------------------------------------------------|
| <height> | The height above the body's reference ellipsoid. |
| <index>  | The index corresponding to this height.          |

**Returns**

The smoothed density.

**Note**

Smoothing has not been implemented in this build because integration seems stable across the small discontinuities in Vallado's model.

## 8.22.4 Member Data Documentation

### 8.22.4.1 altitudeBands

`Integer` ExponentialAtmosphere::altitudeBands [protected]

Number of altitude bands used in the model.

### 8.22.4.2 refDensity

`Real*` ExponentialAtmosphere::refDensity [protected]

Table of reference densities,  $\rho_0$ .

### 8.22.4.3 refHeight

`Real*` ExponentialAtmosphere::refHeight [protected]

Table of Reference heights,  $h_0$ .

### 8.22.4.4 scaleHeight

`Real*` ExponentialAtmosphere::scaleHeight [protected]

Table of scale heights,  $H$ .



## 8.22.4.5 smoothDensity

```
bool ExponentialAtmosphere::smoothDensity [protected]
```

Flag indicating if the altitude is "at" a boundary.

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

- GMATsrc/base/[ExponentialAtmosphere.hpp](#)
- GMATsrc/base/[ExponentialAtmosphere.cpp](#)

## 8.23 FileManager Class Reference

```
#include <FileManager.hpp>
```

## Public Types

- enum [FileType](#) {  
[BEGIN\\_OF\\_PATH](#) = 0, [ROOT\\_PATH](#), [TIME\\_PATH](#), [PLANETARY\\_COEFF\\_PATH](#),  
[PLANETARY\\_EPHEM\\_DE\\_PATH](#), [PLANETARY\\_EPHEM\\_SPK\\_PATH](#), [VEHICLE\\_EPHEM\\_PATH](#), [VEHICLE\\_EPHEM\\_SPK\\_PATH](#),  
[VEHICLE\\_EPHEM\\_CCSDS\\_PATH](#), [EARTH\\_POT\\_PATH](#), [LUNA\\_POT\\_PATH](#), [VENUS\\_POT\\_PATH](#),  
[MARS\\_POT\\_PATH](#), [OTHER\\_POT\\_PATH](#), [TEXTURE\\_PATH](#), [BODY\\_3D\\_MODEL\\_PATH](#),  
[MEASUREMENT\\_PATH](#), [GUI\\_CONFIG\\_PATH](#), [SPLASH\\_PATH](#), [ICON\\_PATH](#),  
[STAR\\_PATH](#), [VEHICLE\\_MODEL\\_PATH](#), [SPAD\\_PATH](#), [ATMOSPHERE\\_PATH](#),  
[FILE\\_UPDATE\\_PATH](#), [OUTPUT\\_PATH](#), [END\\_OF\\_PATH](#), [LOG\\_FILE](#),  
[REPORT\\_FILE](#), [EPHEM\\_OUTPUT\\_FILE](#), [SPLASH\\_FILE](#), [TIME\\_COEFF\\_FILE](#),  
[DE405\\_FILE](#), [DE421\\_FILE](#), [DE424\\_FILE](#), [DE430\\_FILE](#),  
[IAUSOFA\\_FILE](#), [ICRF\\_FILE](#), [PLANETARY\\_SPK\\_FILE](#), [JGM2\\_FILE](#),  
[JGM3\\_FILE](#), [EGM96\\_FILE](#), [LP165P\\_FILE](#), [MGNP180U\\_FILE](#),  
[MARS50C\\_FILE](#), [EOP\\_FILE](#), [PLANETARY\\_COEFF\\_FILE](#), [NUTATION\\_COEFF\\_FILE](#),  
[PLANETARY\\_PCK\\_FILE](#), [EARTH\\_LATEST\\_PCK\\_FILE](#), [EARTH\\_PCK\\_PREDICTED\\_FILE](#), [EARTH\\_PCK\\_CURRENT\\_FILE](#),  
[LUNA\\_PCK\\_CURRENT\\_FILE](#), [LUNA\\_FRAME\\_KERNEL\\_FILE](#), [LEAP\\_SECS\\_FILE](#), [LSK\\_FILE](#),  
[PERSONALIZATION\\_FILE](#), [MAIN\\_ICON\\_FILE](#), [STAR\\_FILE](#), [CONSTELLATION\\_FILE](#),  
[SPACECRAFT\\_MODEL\\_FILE](#), [SPAD\\_SRP\\_FILE](#), [CSSI\\_FLUX\\_FILE](#), [SCHATTEN\\_FILE](#),  
[MARINI\\_TROPO\\_FILE](#), [HELP\\_FILE](#), [FileTypeCount](#) }

## Public Member Functions

- [~FileManager](#) ()
- [std::string GetBinDirectory](#) (const [std::string](#) &appName="GMAT.exe")  
*GMAT application directory.*
- [bool SetBinDirectory](#) (const [std::string](#) &appName="GMAT.exe", const [std::string](#) &newBin="")
- [std::string GetGmatWorkingDirectory](#) ()  
*GMAT working directory.*
- [bool SetGmatWorkingDirectory](#) (const [std::string](#) &newDir="")
- [std::string GetCurrentWorkingDirectory](#) ()  
*System's current working directory of the process.*
- [bool SetCurrentWorkingDirectory](#) (const [std::string](#) &newDir="")

- `std::string FindPath` (const `std::string` &fileName, const `FileType` type, bool forInput, bool writeWarning=false, bool writeInfo=false, const `std::string` &objName="")  
*Finds file path using search order.*
- `std::string FindPath` (const `std::string` &fileName, const `std::string` &fileType, bool forInput, bool writeWarning=false, bool writeInfo=false, const `std::string` &objName="")
- `std::string FindMainIconFile` (bool writeInfo=false)
- `std::string GetPathSeparator` ()
- bool `DoesDirectoryExist` (const `std::string` &dirPath, bool isBlankOk=true)
- bool `DoesFileExist` (const `std::string` &filename)
- bool `RenameFile` (const `std::string` &oldName, const `std::string` &newName, `Integer` &retCode, bool overwriteIfExists=false)
- bool `CopyFile` (const `std::string` &oldName, const `std::string` &newName, `Integer` &retCode, bool overwriteIfExists=false)
- bool `ValidatePaths` ()
- `std::string GetStartupFileDir` ()
- `std::string GetStartupFileName` ()
- `std::string GetFullStartupFilePath` ()
- void `ReadStartupFile` (const `std::string` &fileName="")
- void `WriteStartupFile` (const `std::string` &fileName="")
- `std::string GetRootPath` ()
- bool `GetTextureMapFile` (const `std::string` &inFileName, const `std::string` &bodyName, const `std::string` &objName, `std::string` &outFileName, `std::string` &outFullPathName, bool writeWarning)
- bool `GetBody3dModelFile` (const `std::string` &inFileName, const `std::string` &bodyName, const `std::string` &objName, `std::string` &outFileName, `std::string` &outFullPathName, bool writeWarning)
- `std::string GetPathname` (const `FileType` type)
- `std::string GetPathname` (const `std::string` &typeName)
- `std::string GetFilename` (const `FileType` type)
- `std::string GetFilename` (const `std::string` &typeName)
- `std::string GetFullPathname` (const `FileType` type)
- `std::string GetFullPathname` (const `std::string` &typeName)
- `std::string GetAbsPathname` (const `FileType` type)
- `std::string GetAbsPathname` (const `std::string` &typeName)
- `std::string ConvertToAbsPath` (const `std::string` &relPath, bool appendPathSep=true)
- void `SetAbsPathname` (const `FileType` type, const char \*newpath)
- void `SetAbsPathname` (const `FileType` type, const `std::string` &newpath)
- void `SetAbsPathname` (const `std::string` &type, const char \*newpath)
- void `SetAbsPathname` (const `std::string` &type, const `std::string` &newpath)
- void `ClearGmatIncludePath` ()
- void `AddGmatIncludePath` (const char \*path, bool addFront=true)
- void `AddGmatIncludePath` (const `std::string` &path, bool addFront=true)
- `std::string GetGmatIncludePath` (const char \*incName)
- `std::string GetGmatIncludePath` (const `std::string` &incName)
- const `StringArray` & `GetAllGmatIncludePaths` ()
- void `ClearGmatFunctionPath` ()
- void `AddGmatFunctionPath` (const char \*path, bool addFront=true)
- void `AddGmatFunctionPath` (const `std::string` &path, bool addFront=true)
- `std::string GetGmatFunctionPath` (const char \*funcName)
- `std::string GetGmatFunctionPath` (const `std::string` &funcName)
- const `StringArray` & `GetAllGmatFunctionPaths` ()
- void `ClearMatlabFunctionPath` ()
- void `AddMatlabFunctionPath` (const char \*path, bool addFront=true)
- void `AddMatlabFunctionPath` (const `std::string` &path, bool addFront=true)
- `std::string GetMatlabFunctionPath` (const char \*funcName)
- `std::string GetMatlabFunctionPath` (const `std::string` &funcName)
- const `StringArray` & `GetAllMatlabFunctionPaths` ()

- void [AddPythonModulePath](#) (const std::string &path)
- const [StringArray](#) & [GetAllPythonModulePaths](#) ()
- std::string [GetLastFilePathMessage](#) ()
- const [StringArray](#) & [GetPluginList](#) ()
- void [AdjustSettings](#) (const std::string &suffix, const [StringArray](#) &forEntries)

### Static Public Member Functions

- static [FileManager](#) \* [Instance](#) (const std::string &appName="GMAT.exe")

#### 8.23.1 Detailed Description

Declares [FileManager](#) class. This is singleton class which manages list of file paths and names.

The textures files and non-Earth gravity potential files not appear in the predefined enum `FileType` list can be retrieved by using file naming convention. The texture files should have `PLANETNAME_TEXTURE_FILE`. e.g. "EARTH\_TEXTURE\_FILE", "LUNA\_TEXTURE\_FILE", etc. The potential files should have `PLANETNAME_POT_FILE`.

#### 8.23.2 Member Enumeration Documentation

##### 8.23.2.1 FileType

enum [FileManager::FileType](#)

##### Enumerator

|                          |  |
|--------------------------|--|
| BEGIN_OF_PATH            |  |
| ROOT_PATH                |  |
| TIME_PATH                |  |
| PLANETARY_COEFF_PATH     |  |
| PLANETARY_EPHEM_DE_PATH  |  |
| PLANETARY_EPHEM_SPK_PATH |  |
| VEHICLE_EPHEM_PATH       |  |
| VEHICLE_EPHEM_SPK_PATH   |  |
| VEHICLE_EPHEM_CCSDS_PATH |  |
| EARTH_POT_PATH           |  |
| LUNA_POT_PATH            |  |
| VENUS_POT_PATH           |  |
| MARS_POT_PATH            |  |
| OTHER_POT_PATH           |  |
| TEXTURE_PATH             |  |
| BODY_3D_MODEL_PATH       |  |
| MEASUREMENT_PATH         |  |
| GUI_CONFIG_PATH          |  |
| SPLASH_PATH              |  |
| ICON_PATH                |  |

## Enumerator

|                          |  |
|--------------------------|--|
| STAR_PATH                |  |
| VEHICLE_MODEL_PATH       |  |
| SPAD_PATH                |  |
| ATMOSPHERE_PATH          |  |
| FILE_UPDATE_PATH         |  |
| OUTPUT_PATH              |  |
| END_OF_PATH              |  |
| LOG_FILE                 |  |
| REPORT_FILE              |  |
| EPHEM_OUTPUT_FILE        |  |
| SPLASH_FILE              |  |
| TIME_COEFF_FILE          |  |
| DE405_FILE               |  |
| DE421_FILE               |  |
| DE424_FILE               |  |
| DE430_FILE               |  |
| IAUSOFA_FILE             |  |
| ICRF_FILE                |  |
| PLANETARY_SPK_FILE       |  |
| JGM2_FILE                |  |
| JGM3_FILE                |  |
| EGM96_FILE               |  |
| LP165P_FILE              |  |
| MGNP180U_FILE            |  |
| MARS50C_FILE             |  |
| EOP_FILE                 |  |
| PLANETARY_COEFF_FILE     |  |
| NUTATION_COEFF_FILE      |  |
| PLANETARY_PCK_FILE       |  |
| EARTH_LATEST_PCK_FILE    |  |
| EARTH_PCK_PREDICTED_FILE |  |
| EARTH_PCK_CURRENT_FILE   |  |
| LUNA_PCK_CURRENT_FILE    |  |
| LUNA_FRAME_KERNEL_FILE   |  |
| LEAP_SECS_FILE           |  |
| LSK_FILE                 |  |
| PERSONALIZATION_FILE     |  |
| MAIN_ICON_FILE           |  |
| STAR_FILE                |  |
| CONSTELLATION_FILE       |  |
| SPACECRAFT_MODEL_FILE    |  |
| SPAD_SRP_FILE            |  |
| CSSI_FLUX_FILE           |  |
| SCHATTEN_FILE            |  |
| MARINI_TROPO_FILE        |  |
| HELP_FILE                |  |
| FileTypeCount            |  |

### 8.23.3 Constructor & Destructor Documentation

#### 8.23.3.1 ~FileManager()

```
FileManager::~FileManager ()
```

### 8.23.4 Member Function Documentation

#### 8.23.4.1 AddGmatFunctionPath() [1/2]

```
void FileManager::AddGmatFunctionPath (
 const char * path,
 bool addFront = true)
```

#### 8.23.4.2 AddGmatFunctionPath() [2/2]

```
void FileManager::AddGmatFunctionPath (
 const std::string & path,
 bool addFront = true)
```

#### 8.23.4.3 AddGmatIncludePath() [1/2]

```
void FileManager::AddGmatIncludePath (
 const char * path,
 bool addFront = true)
```

#### 8.23.4.4 AddGmatIncludePath() [2/2]

```
void FileManager::AddGmatIncludePath (
 const std::string & path,
 bool addFront = true)
```

#### 8.23.4.5 AddMatlabFunctionPath() [1/2]

```
void FileManager::AddMatlabFunctionPath (
 const char * path,
 bool addFront = true)
```

#### 8.23.4.6 AddMatlabFunctionPath() [2/2]

```
void FileManager::AddMatlabFunctionPath (
 const std::string & path,
 bool addFront = true)
```

#### 8.23.4.7 AddPythonModulePath()

```
void FileManager::AddPythonModulePath (
 const std::string & path)
```

Adds a folder to the buffer for the Python search path

##### Parameters

|             |                                                |
|-------------|------------------------------------------------|
| <i>path</i> | The new folder that may contain Python modules |
|-------------|------------------------------------------------|

#### 8.23.4.8 AdjustSettings()

```
void FileManager::AdjustSettings (
 const std::string & suffix,
 const StringArray & forEntries)
```

Appends a suffix to a list of settings stored in the file manager

##### Parameters

|                   |                                           |
|-------------------|-------------------------------------------|
| <i>suffix</i>     | The suffix to be appended to the setting  |
| <i>forEntries</i> | A list of entries that receive the suffix |

#### 8.23.4.9 ClearGmatFunctionPath()

```
void FileManager::ClearGmatFunctionPath ()
```

#### 8.23.4.10 ClearGmatIncludePath()

```
void FileManager::ClearGmatIncludePath ()
```

#### 8.23.4.11 ClearMatlabFunctionPath()

```
void FileManager::ClearMatlabFunctionPath ()
```

#### 8.23.4.12 ConvertToAbsPath()

```
std::string FileManager::ConvertToAbsPath (
 const std::string & relPath,
 bool appendPathSep = true)
```

Converts relative path to absolute path

#### 8.23.4.13 CopyFile()

```
bool FileManager::CopyFile (
 const std::string & oldName,
 const std::string & newName,
 Integer & retCode,
 bool overwriteIfExists = false)
```

#### 8.23.4.14 DoesDirectoryExist()

```
bool FileManager::DoesDirectoryExist (
 const std::string & dirPath,
 bool isBlankOk = true)
```

#### 8.23.4.15 DoesFileExist()

```
bool FileManager::DoesFileExist (
 const std::string & filename)
```

#### 8.23.4.16 FindMainIconFile()

```
std::string FileManager::FindMainIconFile (
 bool writeInfo = false)
```

#### 8.23.4.17 FindPath() [1/2]

```
std::string FileManager::FindPath (
 const std::string & fileName,
 const FileType type,
 bool forInput,
 bool writeWarning = false,
 bool writeInfo = false,
 const std::string & objName = "")
```

Finds file path using search order.

Finds path for requested fileName using the file path search order. This method calls [FindPath\(\)](#) taking type name.

#### Returns

path found using search order

#### 8.23.4.18 FindPath() [2/2]

```
std::string FileManager::FindPath (
 const std::string & fileName,
 const std::string & fileType,
 bool forInput,
 bool writeWarning = false,
 bool writeInfo = false,
 const std::string & objName = "")
```

Finds path for requested fileName. If fileName has a absolute path, it will return fileName or blank if path not found. If fileName has a relative path or no path, it will find path using the following file path search order. For Input: 1) Current GMAT working directory 2) Directory from the startup file in the application directory For Output: 1) Current GMAT working directory if it has relative path 2) Directory from the startup file in the application directory if no path found 3) Application directory

It returns blank if filename is blank It returns blank if path not found for input file. If input fileName is blank, it uses default filename using the type

#### Parameters

|                     |                                                                                                |
|---------------------|------------------------------------------------------------------------------------------------|
| <i>fileName</i>     | The requested filename to be searched Enter blank name if default name to be used for the type |
| <i>file Type</i>    | The file type name of the input file                                                           |
| <i>forInput</i>     | Set to true if filename is for input                                                           |
| <i>writeWarning</i> | Set to true if warning should be written when no path found                                    |
| <i>writeInfo</i>    | Set to true if information should be written for output path (currently not used)              |
| <i>objName</i>      | The name of the calling object to be written to informational message                          |



**Returns**

full path name using search order

**8.23.4.19 GetAbsPathname()** [1/2]

```
std::string FileManager::GetAbsPathname (
 const FileType type)
```

Retrieves full pathname for the type.

**Parameters**

|        |                                             |
|--------|---------------------------------------------|
| <type> | file type of which filename to be returned. |
|--------|---------------------------------------------|

**Returns**

file pathname if file type found

**Exceptions**

|               |                               |
|---------------|-------------------------------|
| <i>thrown</i> | if enum type is out of bounds |
|---------------|-------------------------------|

**8.23.4.20 GetAbsPathname()** [2/2]

```
std::string FileManager::GetAbsPathname (
 const std::string & typeName)
```

Retrieves full pathname for the type name.

**Parameters**

|        |                                                  |
|--------|--------------------------------------------------|
| <type> | file type name of which filename to be returned. |
|--------|--------------------------------------------------|

**Returns**

file pathname if file type name found

**Exceptions**

|               |                          |
|---------------|--------------------------|
| <i>thrown</i> | if type cannot be found. |
|---------------|--------------------------|

#### 8.23.4.21 GetAllGmatFunctionPaths()

```
const StringArray & FileManager::GetAllGmatFunctionPaths ()
```

#### 8.23.4.22 GetAllGmatIncludePaths()

```
const StringArray & FileManager::GetAllGmatIncludePaths ()
```

#### 8.23.4.23 GetAllMatlabFunctionPaths()

```
const StringArray & FileManager::GetAllMatlabFunctionPaths ()
```

#### 8.23.4.24 GetAllPythonModulePaths()

```
const StringArray & FileManager::GetAllPythonModulePaths ()
```

#### 8.23.4.25 GetBinDirectory()

```
std::string FileManager::GetBinDirectory (
 const std::string & appName = "GMAT.exe")
```

GMAT application directory.

#### 8.23.4.26 GetBody3dModelFile()

```
bool FileManager::GetBody3dModelFile (
 const std::string & inFileName,
 const std::string & bodyName,
 const std::string & objName,
 std::string & outFileName,
 std::string & outFullPathName,
 bool writeWarning)
```

#### 8.23.4.27 GetCurrentWorkingDirectory()

```
std::string FileManager::GetCurrentWorkingDirectory ()
```

System's current working directory of the process.

##### Returns

System's current working directory of the process

#### 8.23.4.28 GetFilename() [1/2]

```
std::string FileManager::GetFilename (
 const FileType type)
```

Retrives filename for the type without path.

##### Parameters

|        |                                                  |
|--------|--------------------------------------------------|
| <type> | enum file type of which filename to be returned. |
|--------|--------------------------------------------------|

##### Returns

file filename if file type found

##### Exceptions

|        |                               |
|--------|-------------------------------|
| thrown | if enum type is out of bounds |
|--------|-------------------------------|

#### 8.23.4.29 GetFilename() [2/2]

```
std::string FileManager::GetFilename (
 const std::string & typeName)
```

Retrives filename for the type name without path.

##### Parameters

|        |                                                  |
|--------|--------------------------------------------------|
| <type> | file type name of which filename to be returned. |
|--------|--------------------------------------------------|

##### Returns

file filename if file type found

**Exceptions**

|               |                          |
|---------------|--------------------------|
| <i>thrown</i> | if type cannot be found. |
|---------------|--------------------------|

**8.23.4.30 GetFullPathname()** [1/2]

```
std::string FileManager::GetFullPathname (
 const FileType type)
```

Retrieves full pathname for the type.

**Parameters**

|        |                                             |
|--------|---------------------------------------------|
| <type> | file type of which filename to be returned. |
|--------|---------------------------------------------|

**Returns**

file pathname if file type found

**Exceptions**

|               |                               |
|---------------|-------------------------------|
| <i>thrown</i> | if enum type is out of bounds |
|---------------|-------------------------------|

**8.23.4.31 GetFullPathname()** [2/2]

```
std::string FileManager::GetFullPathname (
 const std::string & typeName)
```

Retrives full pathname for the type name.

**Parameters**

|        |                                                  |
|--------|--------------------------------------------------|
| <type> | file type name of which filename to be returned. |
|--------|--------------------------------------------------|

**Returns**

file pathname if file type name found

**Exceptions**

|               |                          |
|---------------|--------------------------|
| <i>thrown</i> | if type cannot be found. |
|---------------|--------------------------|

**8.23.4.32 GetFullStartupFilePath()**

```
std::string FileManager::GetFullStartupFilePath ()
```

**8.23.4.33 GetGmatFunctionPath()** [1/2]

```
std::string FileManager::GetGmatFunctionPath (
 const char * funcName)
```

**8.23.4.34 GetGmatFunctionPath()** [2/2]

```
std::string FileManager::GetGmatFunctionPath (
 const std::string & funcName)
```

**8.23.4.35 GetGmatIncludePath()** [1/2]

```
std::string FileManager::GetGmatIncludePath (
 const char * incName)
```

**8.23.4.36 GetGmatIncludePath()** [2/2]

```
std::string FileManager::GetGmatIncludePath (
 const std::string & incName)
```

**8.23.4.37 GetGmatWorkingDirectory()**

```
std::string FileManager::GetGmatWorkingDirectory ()
```

GMAT working directory.

Returns GMAT working directory. This is the directory where script is passed to GMAT from the command line.

**8.23.4.38 GetLastFilePathMessage()**

```
std::string FileManager::GetLastFilePathMessage ()
```

Returns the last file path message set from [FindPath\(\)](#).

**8.23.4.39 GetMatlabFunctionPath()** [1/2]

```
std::string FileManager::GetMatlabFunctionPath (
 const char * funcName)
```

**8.23.4.40 GetMatlabFunctionPath()** [2/2]

```
std::string FileManager::GetMatlabFunctionPath (
 const std::string & funcName)
```

**8.23.4.41 GetPathname()** [1/2]

```
std::string FileManager::GetPathname (
 const FileType type)
```

Retrives absolute path for the type without filename.

**Parameters**

|        |                                              |
|--------|----------------------------------------------|
| <type> | enum file type of which path to be returned. |
|--------|----------------------------------------------|

**Returns**

file pathname if path type found.

**Exceptions**

|        |                                |
|--------|--------------------------------|
| thrown | if enum type is out of bounds. |
|--------|--------------------------------|

**8.23.4.42 GetPathname()** [2/2]

```
std::string FileManager::GetPathname (
 const std::string & typeName)
```

Retrives absolute pathname for the type name without filename.

## Parameters

|                               |                                                  |
|-------------------------------|--------------------------------------------------|
| <code>&lt;typeName&gt;</code> | file type name of which pathname to be returned. |
|-------------------------------|--------------------------------------------------|

## Returns

pathname if type found.

## Exceptions

|               |                          |
|---------------|--------------------------|
| <i>thrown</i> | if type cannot be found. |
|---------------|--------------------------|

## 8.23.4.43 GetPathSeparator()

```
std::string FileManager::GetPathSeparator ()
```

## Returns

path separator; "/" or "\\" depends on the platform

## 8.23.4.44 GetPluginList()

```
const StringArray & FileManager::GetPluginList ()
```

Accesses the list of plug-in libraries parsed from the startup file.

## Returns

The list of plug-in libraries

## 8.23.4.45 GetRootPath()

```
std::string FileManager::GetRootPath ()
```

Retrives root pathname.

## Returns

file pathname if path type found.

#### 8.23.4.46 GetStartupFileDir()

```
std::string FileManager::GetStartupFileDir ()
```

#### 8.23.4.47 GetStartupFileName()

```
std::string FileManager::GetStartupFileName ()
```

#### 8.23.4.48 GetTextureMapFile()

```
bool FileManager::GetTextureMapFile (
 const std::string & inFileName,
 const std::string & bodyName,
 const std::string & objName,
 std::string & outFileName,
 std::string & outFullPathName,
 bool writeWarning)
```

#### 8.23.4.49 Instance()

```
FileManager * FileManager::Instance (
 const std::string & appName = "GMAT.exe") [static]
```

#### 8.23.4.50 ReadStartupFile()

```
void FileManager::ReadStartupFile (
 const std::string & fileName = "")
```

Reads GMAT startup file.

##### Parameters

|            |                    |
|------------|--------------------|
| <fileName> | startup file name. |
|------------|--------------------|

**Todo** This code replaces relative paths with absolute. It was implemented to address an issue in R2014a, but the side effects were too severe for the release. It is commented out so that post release, we can assess how to proceed addressing path issues in GMAT.



#### 8.23.4.51 RenameFile()

```
bool FileManager::RenameFile (
 const std::string & oldName,
 const std::string & newName,
 Integer & retCode,
 bool overwriteIfExists = false)
```

#### 8.23.4.52 SetAbsPathname() [1/4]

```
void FileManager::SetAbsPathname (
 const FileType type,
 const char * newPath)
```

Sets absolute pathname for the type.

##### Parameters

|           |                                    |
|-----------|------------------------------------|
| <type>    | file type of which path to be set. |
| <newpath> | new pathname.                      |

##### Exceptions

|               |                               |
|---------------|-------------------------------|
| <i>thrown</i> | if enum type is out of bounds |
|---------------|-------------------------------|

#### 8.23.4.53 SetAbsPathname() [2/4]

```
void FileManager::SetAbsPathname (
 const FileType type,
 const std::string & newPath)
```

Sets absolute pathname for the type.

##### Parameters

|           |                                    |
|-----------|------------------------------------|
| <type>    | file type of which path to be set. |
| <newpath> | new pathname.                      |

##### Exceptions

|               |                               |
|---------------|-------------------------------|
| <i>thrown</i> | if enum type is out of bounds |
|---------------|-------------------------------|

**8.23.4.54 SetAbsPathname()** [3/4]

```
void FileManager::SetAbsPathname (
 const std::string & type,
 const char * newpath)
```

Sets absolute pathname for the type.

**Parameters**

|                    |                                    |
|--------------------|------------------------------------|
| < <i>type</i> >    | type name of which path to be set. |
| < <i>newpath</i> > | new pathname.                      |

**Exceptions**

|               |                               |
|---------------|-------------------------------|
| <i>thrown</i> | if enum type is out of bounds |
|---------------|-------------------------------|

**8.23.4.55 SetAbsPathname()** [4/4]

```
void FileManager::SetAbsPathname (
 const std::string & type,
 const std::string & newpath)
```

Sets absolute pathname for the type.

**Parameters**

|                    |                                    |
|--------------------|------------------------------------|
| < <i>type</i> >    | type name of which path to be set. |
| < <i>newpath</i> > | new pathname.                      |

**Exceptions**

|               |                               |
|---------------|-------------------------------|
| <i>thrown</i> | if enum type is out of bounds |
|---------------|-------------------------------|

**8.23.4.56 SetBinDirectory()**

```
bool FileManager::SetBinDirectory (
 const std::string & appName = "GMAT.exe",
 const std::string & binDir = "")
```

Sets bin directory where GMAT.exe reside. It sets only once when GMAT.exe found in the directory. If input binDir is blank, it will try with [GmatFileUtil::GetApplicationPath\(\)](#).

**8.23.4.57 SetCurrentWorkingDirectory()**

```
bool FileManager::SetCurrentWorkingDirectory (
 const std::string & newDir = "")
```

Sets system's current working directory of the process.

**8.23.4.58 SetGmatWorkingDirectory()**

```
bool FileManager::SetGmatWorkingDirectory (
 const std::string & newDir = "")
```

Sets GMAT working directory. This is the directory where script resides.

**8.23.4.59 ValidatePaths()**

```
bool FileManager::ValidatePaths ()
```

**8.23.4.60 WriteStartupFile()**

```
void FileManager::WriteStartupFile (
 const std::string & fileName = "")
```

Writes GMAT startup file.

**Parameters**

|            |                    |
|------------|--------------------|
| <fileName> | startup file name. |
|------------|--------------------|

**Exceptions**

|                                         |                                 |
|-----------------------------------------|---------------------------------|
| <a href="#"><i>UtilityException</i></a> | thrown if file cannot be opened |
|-----------------------------------------|---------------------------------|

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

- GMATsrc/util/[FileManager.hpp](#)
- GMATsrc/util/[FileManager.cpp](#)

**8.24 geoparms Struct Reference**

```
#include <gmatdefs.hpp>
```

## Public Attributes

- [Real xtemp](#)
- [Real tkp](#)

*minimum global exospheric temperature (degrees K)*

## 8.24.1 Member Data Documentation

### 8.24.1.1 tkp

[Real](#) geoparms::tkp

minimum global exospheric temperature (degrees K)

### 8.24.1.2 xtemp

[Real](#) geoparms::xtemp

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

- [GMATsrc/include/gmatdefs.hpp](#)

## 8.25 GmatGlobal Class Reference

```
#include <GmatGlobal.hpp>
```

## Public Types

- enum [RunMode](#) { [NORMAL](#) = 10, [EXIT\\_AFTER\\_RUN](#), [TESTING](#), [TESTING\\_NO\\_PLOTS](#) }
- enum [GuiMode](#) { [NORMAL\\_GUI](#) = 20, [MINIMIZED\\_GUI](#) }
- enum [PlotMode](#) { [NORMAL\\_PLOT](#) = 25, [TILED\\_PLOT](#), [CASCADED\\_PLOT](#) }
- enum [MatlabMode](#) { [SINGLE\\_USE](#) = 30, [SHARED](#), [NO\\_MATLAB](#) }
- enum [LogfileSource](#) { [CMD\\_LINE](#) = 35, [SCRIPT](#), [STARTUP](#) }

## Public Member Functions

- [std::string GetGmatVersion \(\)](#)
- [bool IsGmatCompiledIn64Bit \(\)](#)
- [Integer GetDataPrecision \(\)](#)
- [Integer GetTimePrecision \(\)](#)
- [Integer GetDataWidth \(\)](#)
- [Integer GetTimeWidth \(\)](#)
- [Integer GetIntegerWidth \(\)](#)
- [std::string GetOutputPath \(\)](#)
- [void SetDataPrecision \(Integer p\)](#)
- [void SetTimePrecision \(Integer p\)](#)
- [void SetDataWidth \(Integer w\)](#)
- [void SetTimeWidth \(Integer w\)](#)
- [void SetIntegerWidth \(Integer w\)](#)
- [void SetOutputPath \(const std::string &path\)](#)
- [void SetMatlabFuncNameExt \(const std::string &ext\)](#)
- [std::string GetMatlabFuncNameExt \(\)](#)
- [bool IsBatchMode \(\)](#)
- [void SetBatchMode \(bool flag\)](#)
- [bool IsNitsClient \(\)](#)
- [void SetNitsClient \(bool flag\)](#)
- [bool GetRunInterrupted \(\)](#)
- [void SetRunInterrupted \(bool flag\)](#)
- [Gmat::RunState GetRunState \(\)](#)
- [void SetRunState \(Gmat::RunState rs\)](#)
- [Gmat::RunState GetDetailedRunState \(\)](#)
- [void SetDetailedRunState \(Gmat::RunState drs\)](#)
- [Integer GetRunMode \(\)](#)
- [Integer GetRunModeStartUp \(\)](#)
- [void SetRunMode \(Integer mode\)](#)
- [Integer GetGuiMode \(\)](#)
- [void SetGuiMode \(Integer mode\)](#)
- [Integer GetPlotMode \(\)](#)
- [void SetPlotMode \(Integer mode\)](#)
- [void SetCommandEchoMode \(bool tf\)](#)
- [bool EchoCommands \(\)](#)
- [void SetSkipSplashMode \(bool tfSplash\)](#)
- [bool SkipSplashMode \(\)](#)
- [Integer GetMatlabMode \(\)](#)
- [void SetMatlabMode \(Integer mode\)](#)
- [bool IsMatlabAvailable \(\)](#)
- [void SetMatlabAvailable \(bool flag\)](#)
- [bool IsMatlabDebugOn \(\)](#)
- [void SetMatlabDebug \(bool flag\)](#)
- [bool IsMissionTreeDebugOn \(\)](#)
- [void SetMissionTreeDebug \(bool flag\)](#)
- [bool IsWritingParameterInfo \(\)](#)
- [void SetWriteParameterInfo \(bool flag\)](#)
- [bool IsWritingFilePathInfo \(\)](#)
- [void SetWriteFilePathInfo \(bool flag\)](#)
- [bool IsWritingGmatKeyword \(\)](#)
- [void SetWriteGmatKeyword \(bool flag\)](#)
- [void SetEventLocationAvailable \(bool flag\)](#)
- [bool IsEventLocationAvailable \(\)](#)

- void [SetIncludeFoundInScriptResource](#) (bool flag)
- bool [GetIncludeFoundInScriptResource](#) ()
- void [SetWritePersonalizationFile](#) (bool flag)
- bool [GetWritePersonalizationFile](#) ()
- bool [IsUISavable](#) ()
- bool [IsScientific](#) ()
- bool [ShowPoint](#) ()
- bool [IsHorizontal](#) ()
- bool [IsBinaryIn](#) ()
- bool [IsBinaryOut](#) ()
- [Integer](#) [GetSpacing](#) ()
- void [SetScientific](#) (bool flag)
- void [SetShowPoint](#) (bool flag)
- void [SetHorizontal](#) (bool flag)
- void [SetBinaryIn](#) (bool flag)
- void [SetBinaryOut](#) (bool flag)
- void [SetSpacing](#) ([Integer](#) sp)
- void [SetPrefix](#) (const std::string &prefix)
- void [SetAppendEol](#) (bool flag)
- void [SetDefaultFormat](#) ()
- void [SetCurrentFormat](#) (bool scientific=false, bool showPoint=false, [Integer](#) width=[GmatGlobal::DATA\\_WIDTH](#), [Integer](#) precision=[GmatGlobal::DATA\\_PRECISION](#), bool horizontal=true, [Integer](#) spacing=1, const std::string &prefix="", bool appendEol=true, bool binaryIn=false, bool binaryOut=false)
- void [GetActualFormat](#) (bool &scientific, bool &showPoint, [Integer](#) &precision, [Integer](#) &width, bool &horizontal, [Integer](#) &spacing, std::string &prefix, bool &appendEol)
- void [SetActualFormat](#) (bool scientific, bool showPoint, [Integer](#) precision, [Integer](#) width, bool horizontal=true, [Integer](#) spacing=1, const std::string &prefix="", bool appendEol=true)
- void [SetToDefaultFormat](#) ()
- void [SetToCurrentFormat](#) ()
- void [ClearHiddenCommands](#) ()
- void [AddHiddenCommand](#) (const std::string &cmd)
- bool [IsHiddenCommand](#) (const char \*cmd)
- bool [IsHiddenCommand](#) (const std::string &cmd)
- void [RemoveHiddenCommand](#) (const std::string &cmd)
- const [StringArray](#) & [GetHiddenCommands](#) ()
- [EopFile](#) \* [GetEopFile](#) ()
- [ItrfCoefficientsFile](#) \* [GetItrfCoefficientsFile](#) ()
- void [SetEopFile](#) ([EopFile](#) \*eop)
- void [SetItrfCoefficientsFile](#) ([ItrfCoefficientsFile](#) \*itr)
- void [SetLogfileSource](#) ([Integer](#) src, const std::string logfileName="")
- void [SetLogfileName](#) ([Integer](#) forSrc, const std::string logfileName)
- [Integer](#) [GetLogfileSource](#) ()
- std::string [GetLogfileName](#) ([Integer](#) forSrc=0)

### Static Public Member Functions

- static [GmatGlobal](#) \* [Instance](#) ()

### Static Public Attributes

- static const [Integer](#) [DATA\\_PRECISION](#) = 16
- static const [Integer](#) [TIME\\_PRECISION](#) = 16
- static const [Integer](#) [DATA\\_WIDTH](#) = 16
- static const [Integer](#) [TIME\\_WIDTH](#) = 16
- static const [Integer](#) [INTEGER\\_WIDTH](#) = 4

## 8.25.1 Member Enumeration Documentation

### 8.25.1.1 GuiMode

enum `GmatGlobal::GuiMode`

#### Enumerator

|               |  |
|---------------|--|
| NORMAL_GUI    |  |
| MINIMIZED_GUI |  |

### 8.25.1.2 LogfileSource

enum `GmatGlobal::LogfileSource`

#### Enumerator

|          |  |
|----------|--|
| CMD_LINE |  |
| SCRIPT   |  |
| STARTUP  |  |

### 8.25.1.3 MatlabMode

enum `GmatGlobal::MatlabMode`

#### Note

MatlabInterface uses the same enum

#### Enumerator

|            |  |
|------------|--|
| SINGLE_USE |  |
| SHARED     |  |
| NO_MATLAB  |  |

### 8.25.1.4 PlotMode

enum `GmatGlobal::PlotMode`

**Enumerator**

|               |  |
|---------------|--|
| NORMAL_PLOT   |  |
| TILED_PLOT    |  |
| CASCADED_PLOT |  |

**8.25.1.5 RunMode**

```
enum GmatGlobal::RunMode
```

**Enumerator**

|                  |  |
|------------------|--|
| NORMAL           |  |
| EXIT_AFTER_RUN   |  |
| TESTING          |  |
| TESTING_NO_PLOTS |  |

**8.25.2 Member Function Documentation****8.25.2.1 AddHiddenCommand()**

```
void GmatGlobal::AddHiddenCommand (
 const std::string & cmd)
```

**8.25.2.2 ClearHiddenCommands()**

```
void GmatGlobal::ClearHiddenCommands ()
```

**8.25.2.3 EchoCommands()**

```
bool GmatGlobal::EchoCommands ()
```

Returns the current command echo mode

**Returns**

true if the mode is set to echo commands to the message window



#### 8.25.2.4 GetActualFormat()

```
void GmatGlobal::GetActualFormat (
 bool & scientific,
 bool & showPoint,
 Integer & precision,
 Integer & width,
 bool & horizontal,
 Integer & spacing,
 std::string & prefix,
 bool & appendEol)
```

#### 8.25.2.5 GetDataPrecision()

```
Integer GmatGlobal::GetDataPrecision ()
```

#### 8.25.2.6 GetDataWidth()

```
Integer GmatGlobal::GetDataWidth ()
```

#### 8.25.2.7 GetDetailedRunState()

```
Gmat::RunState GmatGlobal::GetDetailedRunState ()
```

#### 8.25.2.8 GetEopFile()

```
EopFile * GmatGlobal::GetEopFile ()
```

#### 8.25.2.9 GetGmatVersion()

```
std::string GmatGlobal::GetGmatVersion ()
```

#### 8.25.2.10 GetGuiMode()

```
Integer GmatGlobal::GetGuiMode ()
```

#### 8.25.2.11 GetHiddenCommands()

```
const StringArray & GmatGlobal::GetHiddenCommands ()
```

#### 8.25.2.12 GetIncludeFoundInScriptResource()

```
bool GmatGlobal::GetIncludeFoundInScriptResource ()
```

Returns the #Include statement found in the script resource flag. Normally this flag is set from the ScriptInterpreter and the ResourceTree retrieves it.

##### Returns

The flag

#### 8.25.2.13 GetIntegerWidth()

```
Integer GmatGlobal::GetIntegerWidth ()
```

#### 8.25.2.14 GetItrfCoefficientsFile()

```
ItrfCoefficientsFile * GmatGlobal::GetItrfCoefficientsFile ()
```

#### 8.25.2.15 GetLogfileName()

```
std::string GmatGlobal::GetLogfileName (
 Integer forSrc = 0)
```

#### 8.25.2.16 GetLogfileSource()

```
Integer GmatGlobal::GetLogfileSource ()
```

#### 8.25.2.17 GetMatlabFuncNameExt()

```
std::string GmatGlobal::GetMatlabFuncNameExt ()
```

#### 8.25.2.18 GetMatlabMode()

```
Integer GmatGlobal::GetMatlabMode ()
```

#### 8.25.2.19 GetOutputPath()

```
std::string GmatGlobal::GetOutputPath ()
```

#### 8.25.2.20 GetPlotMode()

```
Integer GmatGlobal::GetPlotMode ()
```

#### 8.25.2.21 GetRunInterrupted()

```
bool GmatGlobal::GetRunInterrupted ()
```

#### 8.25.2.22 GetRunMode()

```
Integer GmatGlobal::GetRunMode ()
```

Returns current run mode: NORMAL - Normal run EXIT\_AFTER\_RUN - GMAT closes after run complete TESTING - GMAT shows extra menu options for testing with plots TESTING\_NO\_PLOTS - GMAT shows extra menu options for testing without plots

#### 8.25.2.23 GetRunModeStartUp()

```
Integer GmatGlobal::GetRunModeStartUp () [inline]
```

#### 8.25.2.24 GetRunState()

```
Gmat::RunState GmatGlobal::GetRunState ()
```

#### 8.25.2.25 GetSpacing()

```
Integer GmatGlobal::GetSpacing ()
```

#### 8.25.2.26 GetTimePrecision()

```
Integer GmatGlobal::GetTimePrecision ()
```

#### 8.25.2.27 GetTimeWidth()

```
Integer GmatGlobal::GetTimeWidth ()
```

#### 8.25.2.28 GetWritePersonalizationFile()

```
bool GmatGlobal::GetWritePersonalizationFile ()
```

#### 8.25.2.29 Instance()

```
GmatGlobal * GmatGlobal::Instance () [static]
```

Accessor method used to obtain the singleton.

##### Returns

the singleton instance of the [GmatGlobal](#) class.

#### 8.25.2.30 IsBatchMode()

```
bool GmatGlobal::IsBatchMode ()
```

**8.25.2.31 IsBinaryIn()**

```
bool GmatGlobal::IsBinaryIn ()
```

**8.25.2.32 IsBinaryOut()**

```
bool GmatGlobal::IsBinaryOut ()
```

**8.25.2.33 IsEventLocationAvailable()**

```
bool GmatGlobal::IsEventLocationAvailable ()
```

Returns the event locator available flag

**Returns**

The flag

**8.25.2.34 IsGmatCompiledIn64Bit()**

```
bool GmatGlobal::IsGmatCompiledIn64Bit ()
```

**8.25.2.35 IsGUISavable()**

```
bool GmatGlobal::IsGUISavable ()
```

Returns flag indicating whether GUI can be saved or not.

**8.25.2.36 IsHiddenCommand()** [1/2]

```
bool GmatGlobal::IsHiddenCommand (
 const char * cmd)
```

**8.25.2.37 IsHiddenCommand()** [2/2]

```
bool GmatGlobal::IsHiddenCommand (
 const std::string & cmd)
```

**8.25.2.38 IsHorizontal()**

```
bool GmatGlobal::IsHorizontal ()
```

**8.25.2.39 IsMatlabAvailable()**

```
bool GmatGlobal::IsMatlabAvailable ()
```

**8.25.2.40 IsMatlabDebugOn()**

```
bool GmatGlobal::IsMatlabDebugOn ()
```

**8.25.2.41 IsMissionTreeDebugOn()**

```
bool GmatGlobal::IsMissionTreeDebugOn ()
```

**8.25.2.42 IsNitsClient()**

```
bool GmatGlobal::IsNitsClient ()
```

**8.25.2.43 IsScientific()**

```
bool GmatGlobal::IsScientific ()
```

**8.25.2.44 IsWritingFilePathInfo()**

```
bool GmatGlobal::IsWritingFilePathInfo ()
```

**8.25.2.45 IsWritingGmatKeyword()**

```
bool GmatGlobal::IsWritingGmatKeyword ()
```

**8.25.2.46 IsWritingParameterInfo()**

```
bool GmatGlobal::IsWritingParameterInfo ()
```

**8.25.2.47 RemoveHiddenCommand()**

```
void GmatGlobal::RemoveHiddenCommand (
 const std::string & cmd)
```

**8.25.2.48 SetActualFormat()**

```
void GmatGlobal::SetActualFormat (
 bool scientific,
 bool showPoint,
 Integer precision,
 Integer width,
 bool horizontal = true,
 Integer spacing = 1,
 const std::string & prefix = "",
 bool appendEol = true)
```

**8.25.2.49 SetAppendEol()**

```
void GmatGlobal::SetAppendEol (
 bool flag)
```

**8.25.2.50 SetBatchMode()**

```
void GmatGlobal::SetBatchMode (
 bool flag)
```

**8.25.2.51 SetBinaryIn()**

```
void GmatGlobal::SetBinaryIn (
 bool flag)
```

#### 8.25.2.52 SetBinaryOut()

```
void GmatGlobal::SetBinaryOut (
 bool flag)
```

#### 8.25.2.53 SetCommandEchoMode()

```
void GmatGlobal::SetCommandEchoMode (
 bool tf)
```

Sets the command echo mode

##### Parameters

|           |                                                        |
|-----------|--------------------------------------------------------|
| <i>tf</i> | true to turn on command echo mde, false to turn it off |
|-----------|--------------------------------------------------------|

#### 8.25.2.54 SetCurrentFormat()

```
void GmatGlobal::SetCurrentFormat (
 bool scientific = false,
 bool showPoint = false,
 Integer width = GmatGlobal::DATA_WIDTH,
 Integer precision = GmatGlobal::DATA_PRECISION,
 bool horizontal = true,
 Integer spacing = 1,
 const std::string & prefix = "",
 bool appendEol = true,
 bool binaryIn = false,
 bool binaryOut = false)
```

#### 8.25.2.55 SetDataPrecision()

```
void GmatGlobal::SetDataPrecision (
 Integer p)
```

#### 8.25.2.56 SetDataWidth()

```
void GmatGlobal::SetDataWidth (
 Integer w)
```



#### 8.25.2.57 SetDefaultFormat()

```
void GmatGlobal::SetDefaultFormat ()
```

#### 8.25.2.58 SetDetailedRunState()

```
void GmatGlobal::SetDetailedRunState (
 Gmat::RunState drs)
```

#### 8.25.2.59 SetEopFile()

```
void GmatGlobal::SetEopFile (
 EopFile * eop)
```

#### 8.25.2.60 SetEventLocationAvailable()

```
void GmatGlobal::SetEventLocationAvailable (
 bool flag)
```

Sets the event locator flag

##### Parameters

|             |                                                             |
|-------------|-------------------------------------------------------------|
| <i>flag</i> | Flag that is true if there are event locators, false if not |
|-------------|-------------------------------------------------------------|

#### 8.25.2.61 SetGuiMode()

```
void GmatGlobal::SetGuiMode (
 Integer mode)
```

#### 8.25.2.62 SetHorizontal()

```
void GmatGlobal::SetHorizontal (
 bool flag)
```

#### 8.25.2.63 SetIncludeFoundInScriptResource()

```
void GmatGlobal::SetIncludeFoundInScriptResource (
 bool flag)
```

Sets the #Include statement found in the script resource flag. Normally this flag is set from the ScriptInterpreter and the ResourceTree retrieves it.

##### Parameters

|             |                                                                  |
|-------------|------------------------------------------------------------------|
| <i>flag</i> | Flag that is true if there are #Include statements, false if not |
|-------------|------------------------------------------------------------------|

#### 8.25.2.64 SetIntegerWidth()

```
void GmatGlobal::SetIntegerWidth (
 Integer w)
```

#### 8.25.2.65 SetItrfCoefficientsFile()

```
void GmatGlobal::SetItrfCoefficientsFile (
 ItrfCoefficientsFile * itr)
```

#### 8.25.2.66 SetLogfileName()

```
void GmatGlobal::SetLogfileName (
 Integer forSrc,
 const std::string logfileName)
```

#### 8.25.2.67 SetLogfileSource()

```
void GmatGlobal::SetLogfileSource (
 Integer src,
 const std::string logfileName = "")
```

#### 8.25.2.68 SetMatlabAvailable()

```
void GmatGlobal::SetMatlabAvailable (
 bool flag)
```

**8.25.2.69 SetMatlabDebug()**

```
void GmatGlobal::SetMatlabDebug (
 bool flag)
```

**8.25.2.70 SetMatlabFuncNameExt()**

```
void GmatGlobal::SetMatlabFuncNameExt (
 const std::string & ext)
```

**8.25.2.71 SetMatlabMode()**

```
void GmatGlobal::SetMatlabMode (
 Integer mode)
```

**8.25.2.72 SetMissionTreeDebug()**

```
void GmatGlobal::SetMissionTreeDebug (
 bool flag)
```

**8.25.2.73 SetNitsClient()**

```
void GmatGlobal::SetNitsClient (
 bool flag)
```

**8.25.2.74 SetOutputPath()**

```
void GmatGlobal::SetOutputPath (
 const std::string & path)
```

**8.25.2.75 SetPlotMode()**

```
void GmatGlobal::SetPlotMode (
 Integer mode)
```

**8.25.2.76 SetPrefix()**

```
void GmatGlobal::SetPrefix (
 const std::string & prefix)
```

**8.25.2.77 SetRunInterrupted()**

```
void GmatGlobal::SetRunInterrupted (
 bool flag)
```

**8.25.2.78 SetRunMode()**

```
void GmatGlobal::SetRunMode (
 Integer mode)
```

**8.25.2.79 SetRunState()**

```
void GmatGlobal::SetRunState (
 Gmat::RunState rs)
```

**8.25.2.80 SetScientific()**

```
void GmatGlobal::SetScientific (
 bool flag)
```

**8.25.2.81 SetShowPoint()**

```
void GmatGlobal::SetShowPoint (
 bool flag)
```

**8.25.2.82 SetSkipSplashMode()**

```
void GmatGlobal::SetSkipSplashMode (
 bool tfSplash)
```

Sets the skip splash screen mode on startup

## Parameters

|           |                                                          |
|-----------|----------------------------------------------------------|
| <i>tf</i> | true to turn on skip splash screen, false to turn it off |
|-----------|----------------------------------------------------------|

**8.25.2.83 SetSpacing()**

```
void GmatGlobal::SetSpacing (
 Integer sp)
```

**8.25.2.84 SetTimePrecision()**

```
void GmatGlobal::SetTimePrecision (
 Integer p)
```

**8.25.2.85 SetTimeWidth()**

```
void GmatGlobal::SetTimeWidth (
 Integer w)
```

**8.25.2.86 SetToCurrentFormat()**

```
void GmatGlobal::SetToCurrentFormat ()
```

**8.25.2.87 SetToDefaultFormat()**

```
void GmatGlobal::SetToDefaultFormat ()
```

**8.25.2.88 SetWriteFilePathInfo()**

```
void GmatGlobal::SetWriteFilePathInfo (
 bool flag)
```

#### 8.25.2.89 SetWriteGmatKeyword()

```
void GmatGlobal::SetWriteGmatKeyword (
 bool flag)
```

#### 8.25.2.90 SetWriteParameterInfo()

```
void GmatGlobal::SetWriteParameterInfo (
 bool flag)
```

#### 8.25.2.91 SetWritePersonalizationFile()

```
void GmatGlobal::SetWritePersonalizationFile (
 bool flag)
```

#### 8.25.2.92 ShowPoint()

```
bool GmatGlobal::ShowPoint ()
```

#### 8.25.2.93 SkipSplashMode()

```
bool GmatGlobal::SkipSplashMode ()
```

Returns the current skip splash screen mode

##### Returns

true if the mode is set to not display the splash screen on startup

### 8.25.3 Member Data Documentation

#### 8.25.3.1 DATA\_PRECISION

```
const Integer GmatGlobal::DATA_PRECISION = 16 [static]
```

## 8.25.3.2 DATA\_WIDTH

```
const Integer GmatGlobal::DATA_WIDTH = 16 [static]
```

## 8.25.3.3 INTEGER\_WIDTH

```
const Integer GmatGlobal::INTEGER_WIDTH = 4 [static]
```

## 8.25.3.4 TIME\_PRECISION

```
const Integer GmatGlobal::TIME_PRECISION = 16 [static]
```

## 8.25.3.5 TIME\_WIDTH

```
const Integer GmatGlobal::TIME_WIDTH = 16 [static]
```

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

- GMATsrc/util/GmatGlobal.hpp
- GMATsrc/util/GmatGlobal.cpp

## 8.26 GmatTime Class Reference

```
#include <GmatTime.hpp>
```

## Public Member Functions

- [GmatTime](#) ()
- virtual [~GmatTime](#) ()
- [GmatTime](#) (const [GmatTime](#) &gt)
- [GmatTime](#) (const [Real](#) mjd)
- [GmatTime operator+](#) (const [GmatTime](#) &gt) const
- [GmatTime operator-](#) (const [GmatTime](#) &gt) const
- [GmatTime operator+](#) (const [Real](#) mjd) const
- [GmatTime operator-](#) (const [Real](#) mjd) const
- [GmatTime operator\\*](#) (const [Real](#) num) const
- [GmatTime operator/](#) (const [Real](#) num) const
- const [GmatTime](#) & [operator=](#) (const [GmatTime](#) &gt)
- const [GmatTime](#) & [operator+=](#) (const [GmatTime](#) &gt)
- const [GmatTime](#) & [operator-=](#) (const [GmatTime](#) &gt)
- const [GmatTime](#) & [operator=](#) (const [Real](#) mjd)

- const [GmatTime](#) & [operator+=](#) (const [Real](#) mjd)
- const [GmatTime](#) & [operator-=](#) (const [Real](#) mjd)
- bool [operator==](#) (const [GmatTime](#) &gt) const
- bool [operator!=](#) (const [GmatTime](#) &gt) const
- bool [operator<](#) (const [GmatTime](#) &gt) const
- bool [operator>](#) (const [GmatTime](#) &gt) const
- bool [operator<=](#) (const [GmatTime](#) &gt) const
- bool [operator>=](#) (const [GmatTime](#) &gt) const
- bool [operator==](#) (const [Real](#) mjd) const
- bool [operator!=](#) (const [Real](#) mjd) const
- bool [operator<](#) (const [Real](#) mjd) const
- bool [operator>](#) (const [Real](#) mjd) const
- bool [operator<=](#) (const [Real](#) mjd) const
- bool [operator>=](#) (const [Real](#) mjd) const
- virtual [GmatTime](#) \* [Clone](#) ()
- virtual void [SetTimeInSec](#) (const [Real](#) sec)
- [GmatEpoch](#) [GetMjd](#) () const
- [Real](#) [GetTimeInSec](#) () const
- long [GetDays](#) () const
- long [GetSec](#) () const
- [Real](#) [GetFracSec](#) () const
- void [SetDays](#) (long days)
- void [SetSec](#) (long sec)
- void [SetFracSec](#) ([Real](#) fsec)
- bool [SetMjdString](#) (std::string sMjd)
- std::string [ToString](#) () const
- bool [IsNearlyEqual](#) ([GmatTime](#) gt, [Real](#) tolerance)
- const [GmatTime](#) & [AddSeconds](#) (const [Real](#) sec)
- const [GmatTime](#) & [SubtractSeconds](#) (const [Real](#) sec)

## Protected Attributes

- long [Days](#)
- long [Sec](#)
- [Real](#) [FracSec](#)

## 8.26.1 Detailed Description

This class is used to define GMAT time with a high precision. It has 2 parts: The first part stores number of seconds. The second part stores fraction of seconds.

## 8.26.2 Constructor & Destructor Documentation

### 8.26.2.1 [GmatTime](#)() [1/3]

```
GmatTime::GmatTime ()
```

This class is used to define GMAT time with a high precision. It has 2 parts: The first part stores number of seconds. The second part stores fraction of seconds.



### 8.26.2.2 ~GmatTime()

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

### 8.26.2.3 GmatTime() [2/3]

```
GmatTime::GmatTime (
 const GmatTime & gt)
```

### 8.26.2.4 GmatTime() [3/3]

```
GmatTime::GmatTime (
 const Real mjd)
```

## 8.26.3 Member Function Documentation

### 8.26.3.1 AddSeconds()

```
const GmatTime & GmatTime::AddSeconds (
 const Real sec)
```

### 8.26.3.2 Clone()

```
GmatTime * GmatTime::Clone () [virtual]
```

### 8.26.3.3 GetDays()

```
long GmatTime::GetDays () const [inline]
```

### 8.26.3.4 GetFracSec()

```
Real GmatTime::GetFracSec () const [inline]
```

#### 8.26.3.5 GetMjd()

```
Real GmatTime::GetMjd () const
```

#### 8.26.3.6 GetSec()

```
long GmatTime::GetSec () const [inline]
```

#### 8.26.3.7 GetTimeInSec()

```
Real GmatTime::GetTimeInSec () const
```

#### 8.26.3.8 IsNearlyEqual()

```
bool GmatTime::IsNearlyEqual (
 GmatTime gt,
 Real tolerance)
```

#### 8.26.3.9 operator!=( ) [1/2]

```
bool GmatTime::operator!= (
 const GmatTime & gt) const
```

#### 8.26.3.10 operator!=( ) [2/2]

```
bool GmatTime::operator!= (
 const Real mjd) const
```

#### 8.26.3.11 operator\*( )

```
GmatTime GmatTime::operator* (
 const Real num) const
```

**8.26.3.12 operator+()** [1/2]

```
GmatTime GmatTime::operator+ (
 const GmatTime & gt) const
```

**8.26.3.13 operator+()** [2/2]

```
GmatTime GmatTime::operator+ (
 const Real mjd) const
```

**8.26.3.14 operator+=()** [1/2]

```
const GmatTime & GmatTime::operator+= (
 const GmatTime & gt)
```

**8.26.3.15 operator+=()** [2/2]

```
const GmatTime & GmatTime::operator+= (
 const Real mjd)
```

**8.26.3.16 operator-()** [1/2]

```
GmatTime GmatTime::operator- (
 const GmatTime & gt) const
```

**8.26.3.17 operator-()** [2/2]

```
GmatTime GmatTime::operator- (
 const Real mjd) const
```

**8.26.3.18 operator-=()** [1/2]

```
const GmatTime & GmatTime::operator-= (
 const GmatTime & gt)
```

**8.26.3.19 operator-=( )** [2/2]

```
const GmatTime & GmatTime::operator-= (
 const Real mjd)
```

**8.26.3.20 operator/( )**

```
GmatTime GmatTime::operator/ (
 const Real num) const
```

**8.26.3.21 operator<( )** [1/2]

```
bool GmatTime::operator< (
 const GmatTime & gt) const
```

**8.26.3.22 operator<=( )** [2/2]

```
bool GmatTime::operator<= (
 const Real mdj) const
```

**8.26.3.23 operator<=( )** [1/2]

```
bool GmatTime::operator<= (
 const GmatTime & gt) const
```

**8.26.3.24 operator<=( )** [2/2]

```
bool GmatTime::operator<= (
 const Real mjd) const
```

**8.26.3.25 operator=( )** [1/2]

```
const GmatTime & GmatTime::operator= (
 const GmatTime & gt)
```

**8.26.3.26 operator=()** [2/2]

```
const GmatTime & GmatTime::operator= (
 const Real mjd)
```

**8.26.3.27 operator==(** [1/2]

```
bool GmatTime::operator== (
 const GmatTime & gt) const
```

**8.26.3.28 operator==(** [2/2]

```
bool GmatTime::operator== (
 const Real mjd) const
```

**8.26.3.29 operator>()** [1/2]

```
bool GmatTime::operator> (
 const GmatTime & gt) const
```

**8.26.3.30 operator>()** [2/2]

```
bool GmatTime::operator> (
 const Real mjd) const
```

**8.26.3.31 operator>=()** [1/2]

```
bool GmatTime::operator>= (
 const GmatTime & gt) const
```

**8.26.3.32 operator>=()** [2/2]

```
bool GmatTime::operator>= (
 const Real mjd) const
```

#### 8.26.3.33 SetDays()

```
void GmatTime::SetDays (
 long days) [inline]
```

#### 8.26.3.34 SetFracSec()

```
void GmatTime::SetFracSec (
 Real fsec) [inline]
```

#### 8.26.3.35 SetMjdString()

```
bool GmatTime::SetMjdString (
 std::string sMjd)
```

#### 8.26.3.36 SetSec()

```
void GmatTime::SetSec (
 long sec) [inline]
```

#### 8.26.3.37 SetTimeInSec()

```
void GmatTime::SetTimeInSec (
 const Real sec) [virtual]
```

#### 8.26.3.38 SubtractSeconds()

```
const GmatTime & GmatTime::SubtractSeconds (
 const Real sec)
```

#### 8.26.3.39 ToString()

```
std::string GmatTime::ToString () const
```

## 8.26.4 Member Data Documentation

### 8.26.4.1 Days

```
long GmatTime::Days [protected]
```

### 8.26.4.2 FracSec

```
Real GmatTime::FracSec [protected]
```

### 8.26.4.3 Sec

```
long GmatTime::Sec [protected]
```

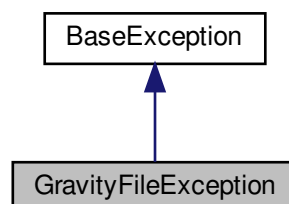
The documentation for this class was generated from the following files:

- GMATsrc/util/GmatTime.hpp
- GMATsrc/util/GmatTime.cpp

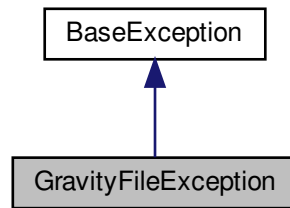
## 8.27 GravityFileException Class Reference

```
#include <UtilityException.hpp>
```

Inheritance diagram for GravityFileException:



Collaboration diagram for GravityFileException:



### Public Member Functions

- [GravityFileException](#) (const std::string &details="")

### Additional Inherited Members

## 8.27.1 Constructor & Destructor Documentation

### 8.27.1.1 GravityFileException()

```
GravityFileException::GravityFileException (
 const std::string & details = "") [inline]
```

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

- GMATsrc/util/[UtilityException.hpp](#)

## 8.28 GregorianCalendar Class Reference

```
#include <GregorianCalendar.hpp>
```

### Classes

- class [GregorianCalendarException](#)



## Public Member Functions

- `GregorianCalendar ()`
- `GregorianCalendar (const std::string &str)`
- `GregorianCalendar (Date *newDate, Integer format=1)`
- `~GregorianCalendar ()`
- `std::string GetDate () const`
- `bool SetDate (const std::string &str)`
- `bool SetDate (Date *newDate, Integer format=1)`
- `std::string GetType () const`
- `bool SetType (const std::string &str)`
- `std::string GetYMDHMS () const`
- `bool IsValid () const`

## Static Public Member Functions

- static `bool IsValid (const std::string &greg)`

### 8.28.1 Detailed Description

Definition of the `GregorianCalendar` class base

### 8.28.2 Constructor & Destructor Documentation

#### 8.28.2.1 `GregorianCalendar()` [1/3]

```
GregorianCalendar::GregorianCalendar ()
```

Definition of the `GregorianCalendar` class base Creates default constructor.

#### 8.28.2.2 `GregorianCalendar()` [2/3]

```
GregorianCalendar::GregorianCalendar (
 const std::string & str)
```

Creates constructor with parameters.

#### Parameters

|       |                                   |
|-------|-----------------------------------|
| <str> | Given String of <code>Date</code> |
|-------|-----------------------------------|

### 8.28.2.3 `GregorianDate()` [3/3]

```
GregorianDate::GregorianDate (
 Date * newDate,
 Integer format = 1)
```

Creates default constructor with new `Date`.

### 8.28.2.4 `~GregorianDate()`

```
GregorianDate::~~GregorianDate ()
```

Destructor.

## 8.28.3 Member Function Documentation

### 8.28.3.1 `GetDate()`

```
std::string GregorianDate::GetDate () const
```

Get the date in string.

#### Returns

the date in string.

### 8.28.3.2 `GetType()`

```
std::string GregorianDate::GetType () const
```

### 8.28.3.3 `GetYMDHMS()`

```
std::string GregorianDate::GetYMDHMS () const
```

Get YYYYMMDD.HHMMSSmmm from Gregorian format in string

#### Returns

string in YYYYMMDD.HHMMSSmmm

#### 8.28.3.4 `IsValid()` [1/2]

```
bool GregorianCalendar::IsValid () const
```

Determines if the date is valid or not.

##### Returns

return flag indicator (true = valid; otherwise, false)

#### 8.28.3.5 `IsValid()` [2/2]

```
bool GregorianCalendar::IsValid (
 const std::string & greg) [static]
```

Determines if input date is valid or not. Valid format is dd mmm yyyy hh:mm:ss.mmm. For example, 01 Jan 2000 12:00:00.000

##### Parameters

|             |                        |
|-------------|------------------------|
| <i>greg</i> | input gregorian string |
|-------------|------------------------|

##### Returns

true if time is in valid Gregorian format; otherwise, false

#### 8.28.3.6 `SetDate()` [1/2]

```
bool GregorianCalendar::SetDate (
 const std::string & str)
```

Set the date in string.

##### Returns

return flag indicator (true = successful; otherwise, false)

#### 8.28.3.7 `SetDate()` [2/2]

```
bool GregorianCalendar::SetDate (
 Date * newDate,
 Integer format = 1)
```

Set the new date in `Date`

## Parameters

|               |                                                              |
|---------------|--------------------------------------------------------------|
| <i>format</i> | 1 = "01 Jan 2000 11:59:28.000" 2 = "2000-01-01T11:59:28.000" |
|---------------|--------------------------------------------------------------|

## Returns

return flag indicator (true = successful; otherwise, false)

## 8.28.3.8 SetType()

```
bool GregorianCalendar::SetType (
 const std::string & str)
```

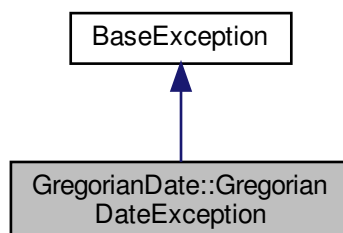
The documentation for this class was generated from the following files:

- [GMATsrc/util/GregorianCalendar.hpp](#)
- [GMATsrc/util/GregorianCalendar.cpp](#)

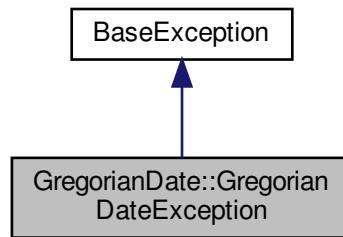
## 8.29 GregorianCalendar::GregorianCalendarException Class Reference

```
#include <GregorianCalendar.hpp>
```

Inheritance diagram for GregorianCalendar::GregorianCalendarException:



Collaboration diagram for `GregorianCalendar::GregorianCalendarException`:



### Public Member Functions

- [GregorianCalendarException](#) (const std::string &message="GregorianCalendarException: Invalid date format")

### Additional Inherited Members

#### 8.29.1 Constructor & Destructor Documentation

##### 8.29.1.1 GregorianCalendarException()

```

GregorianCalendar::GregorianCalendarException::GregorianCalendarException (
 const std::string & message = "GregorianCalendarException: Invalid date format")
[inline]

```

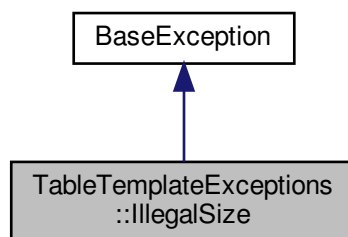
The documentation for this class was generated from the following file:

- `GMATsrc/util/GregorianCalendar.hpp`

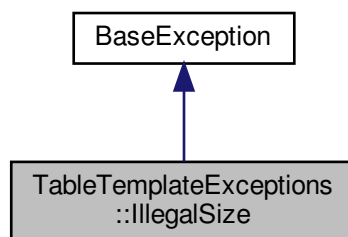
## 8.30 TableTemplateExceptions::IllegalSize Class Reference

```
#include <TableTemplate.hpp>
```

Inheritance diagram for TableTemplateExceptions::IllegalSize:



Collaboration diagram for TableTemplateExceptions::IllegalSize:



## Public Member Functions

- [IllegalSize](#) (const std::string &message="TableTemplate error : illegal (negative) size.\n")

## Additional Inherited Members

### 8.30.1 Constructor & Destructor Documentation

#### 8.30.1.1 IllegalSize()

```
TableTemplateExceptions::IllegalSize::IllegalSize (
 const std::string & message = "TableTemplate error : illegal (negative) size.\n"
) [inline]
```

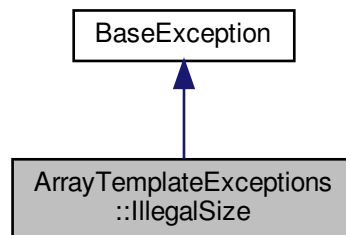
The documentation for this class was generated from the following file:

- GMATsrc/util/[TableTemplate.hpp](#)

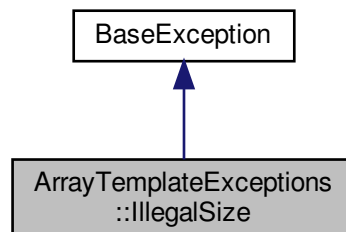
## 8.31 ArrayTemplateExceptions::IllegalSize Class Reference

```
#include <ArrayTemplate.hpp>
```

Inheritance diagram for ArrayTemplateExceptions::IllegalSize:



Collaboration diagram for ArrayTemplateExceptions::IllegalSize:



### Public Member Functions

- [`IllegalSize`](#) (const std::string &message="ArrayTemplate error : illegal size.")

### Additional Inherited Members

#### 8.31.1 Constructor & Destructor Documentation

### 8.31.1.1 IllegalSize()

```
ArrayTemplateExceptions::IllegalSize::IllegalSize (
 const std::string & message = "ArrayTemplate error : illegal size.") [inline]
```

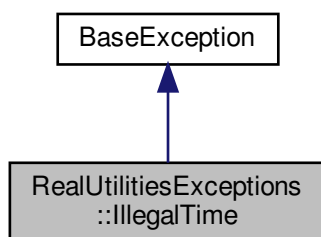
The documentation for this class was generated from the following file:

- GMATsrc/util/[ArrayTemplate.hpp](#)

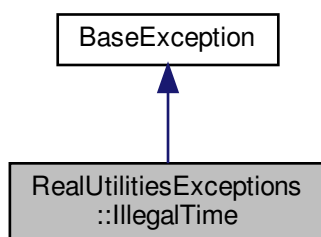
## 8.32 RealUtilitiesExceptions::IllegalTime Class Reference

```
#include <RealUtilities.hpp>
```

Inheritance diagram for RealUtilitiesExceptions::IllegalTime:



Collaboration diagram for RealUtilitiesExceptions::IllegalTime:



### Public Member Functions

- [IllegalTime](#) (const std::string &message="")



## Additional Inherited Members

### 8.32.1 Constructor & Destructor Documentation

#### 8.32.1.1 IllegalTime()

```
RealUtilitiesExceptions::IllegalTime::IllegalTime (
 const std::string & message = "") [inline]
```

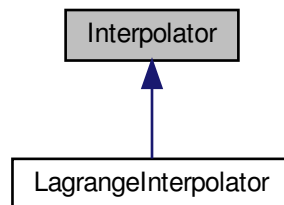
The documentation for this class was generated from the following file:

- GMATsrc/util/[RealUtilities.hpp](#)

## 8.33 Interpolator Class Reference

```
#include <Interpolator.hpp>
```

Inheritance diagram for Interpolator:



### Public Member Functions

- [Interpolator](#) (const std::string &name, const std::string &typestr, [Integer](#) dim=1)
- virtual [~Interpolator](#) ()
- [Interpolator](#) (const [Interpolator](#) &i)
- [Interpolator](#) & operator= (const [Interpolator](#) &i)
- virtual [Integer](#) IsInterpolationFeasible ([Real](#) ind)
- virtual void [SetForceInterpolation](#) (bool flag)
- virtual bool [GetForceInterpolation](#) ()
- virtual bool [AddPoint](#) (const [Real](#) ind, const [Real](#) \*data)
- virtual void [Clear](#) ()
- virtual [Integer](#) GetBufferSize ()
- virtual [Integer](#) GetPointCount ()
- virtual void [SetExtrapolation](#) (bool flag)
- std::string [GetName](#) ()
- bool [GetRange](#) ([Real](#) &lower, [Real](#) &upper)
- virtual bool [Interpolate](#) (const [Real](#) ind, [Real](#) \*results)=0
- virtual [Interpolator](#) \* [Clone](#) () const =0

## Protected Member Functions

- virtual void [AllocateArrays](#) ()
- virtual void [CleanupArrays](#) ()
- virtual void [CopyArrays](#) (const [Interpolator](#) &i)
- void [SetRange](#) ()

## Protected Attributes

- [Real](#) \* [independent](#)  
*Data array used for the independent variable.*
- [Real](#) \*\* [dependent](#)  
*The data that gets interpolated.*
- [Real](#) [previousX](#)  
*Previous independent value, used to determine direction data is going.*
- [Integer](#) [dimension](#)  
*Number of dependent points to be interpolated.*
- [Integer](#) [requiredPoints](#)  
*Number of points required to interpolate.*
- [Integer](#) [bufferSize](#)  
*Number of points managed by the interpolator.*
- [Integer](#) [pointCount](#)  
*Number of points fed to the interpolator.*
- [Integer](#) [latestPoint](#)  
*Pointer to most recent point, for the ring buffer implementation.*
- [Real](#) [range](#) [2]  
*Valid range for the data points.*
- bool [rangeCalculated](#)  
*Flag used to detect if range has already been calculated.*
- bool [dataIncreases](#)  
*Flag used to determine if independent variable increases or decreases.*
- bool [forceInterpolation](#)  
*Flag used for additional feasibility checking.*
- bool [allowExtrapolation](#)  
*Flag to allow extrapolation.*
- std::string [instanceName](#)  
*The name of this interpolator.*

### 8.33.1 Detailed Description

Definition for the [Interpolator](#) base class Base class for the GMAT Interpolators

### 8.33.2 Constructor & Destructor Documentation

#### 8.33.2.1 [Interpolator](#)() [1/2]

```
Interpolator::Interpolator (
 const std::string & name,
 const std::string & typestr,
 Integer dim = 1)
```

Implementation for the [Interpolator](#) base class Constructs the core elements of an [Interpolator](#).

## Parameters

|                              |                                                           |
|------------------------------|-----------------------------------------------------------|
| <code>&lt;name&gt;</code>    | Name for this interpolator.                               |
| <code>&lt;typestr&gt;</code> | Text string identifying the type of interpolator.         |
| <code>&lt;dim&gt;</code>     | Dimension of data that gets interpolated (defaults to 1). |

8.33.2.2 `~Interpolator()`

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

Destroys the core elements of an [Interpolator](#).

8.33.2.3 `Interpolator()` [2/2]

```
Interpolator::Interpolator (
 const Interpolator & i)
```

Constructs the core elements of an [Interpolator](#) (Copy Constructor).

## Parameters

|                        |                                  |
|------------------------|----------------------------------|
| <code>&lt;i&gt;</code> | The interpolator that is copied. |
|------------------------|----------------------------------|

## 8.33.3 Member Function Documentation

8.33.3.1 `AddPoint()`

```
bool Interpolator::AddPoint (
 const Real ind,
 const Real * data) [virtual]
```

Add a data point to the ring buffer used in the [Interpolator](#).

This method is the core method used to fill the buffer prior to interpolation. It manages a ring buffer of data points used by the interpolation routine ([Interpolate\(const Real, Real\\*\)](#)) to generate the estimated parameter values.

## Parameters

|                           |                                                                                                                                                                                                               |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>&lt;ind&gt;</code>  | Value of the independent (domain) variable.                                                                                                                                                                   |
| <code>&lt;data&gt;</code> | Array of dependent data associated with the independent value in the first variable. This array must not be smaller than the dimension of the interpolator instance, or a memory access violation will occur. |

### Returns

True if the data was added to the buffer, false if a problem was encountered.

**Todo** Handle memory access violations when the input array is too small.

Reimplemented in [LagrangeInterpolator](#).

#### 8.33.3.2 AllocateArrays()

```
void Interpolator::AllocateArrays () [protected], [virtual]
```

Allocates the data structures used by the ring buffer.

Reimplemented in [LagrangeInterpolator](#).

#### 8.33.3.3 CleanupArrays()

```
void Interpolator::CleanupArrays () [protected], [virtual]
```

Frees the memory used by the data arrays in the ring buffer, and resets the counters used to manage the buffer.

Reimplemented in [LagrangeInterpolator](#).

#### 8.33.3.4 Clear()

```
void Interpolator::Clear () [virtual]
```

Resets the buffer for the interpolator.

This method does not reallocate the buffer; it just resets the pointers and data counts so that the interpolation can be restarted.

Reimplemented in [LagrangeInterpolator](#).

#### 8.33.3.5 Clone()

```
virtual Interpolator* Interpolator::Clone () const [pure virtual]
```

Implemented in [LagrangeInterpolator](#).

#### 8.33.3.6 CopyArrays()

```
void Interpolator::CopyArrays (
 const Interpolator & i) [protected], [virtual]
```

Copies the ring buffer from one [Interpolator](#) to this one.

## Parameters

|          |                                                                             |
|----------|-----------------------------------------------------------------------------|
| <i>i</i> | The <a href="#">Interpolator</a> that supplies the data copied to this one. |
|----------|-----------------------------------------------------------------------------|

## 8.33.3.7 GetBufferSize()

```
Integer Interpolator::GetBufferSize () [virtual]
```

Access method for the bufferSize parameter.

## Returns

The buffer size.

## 8.33.3.8 GetForceInterpolation()

```
bool Interpolator::GetForceInterpolation () [virtual]
```

## 8.33.3.9 GetName()

```
std::string Interpolator::GetName ()
```

Retrieves the name assigned to this interpolator

## Returns

The instance name

## 8.33.3.10 GetPointCount()

```
Integer Interpolator::GetPointCount () [virtual]
```

Access method for the pointCount parameter.

## Returns

The buffer size.

## 8.33.3.11 GetRange()

```
bool Interpolator::GetRange (
 Real & lower,
 Real & upper)
```

Retrieves the lower and upper range of the independent variable

## Parameters

|              |                                                  |
|--------------|--------------------------------------------------|
| <i>lower</i> | [output] lower end of independent variable range |
| <i>upper</i> | [output] upper end of independent variable range |

## Returns

true if range obtained; false otherwise

## 8.33.3.12 Interpolate()

```
virtual bool Interpolator::Interpolate (
 const Real ind,
 Real * results) [pure virtual]
```

Interpolate the data.

Derived classes implement this method to provide the mathematics that perform the data interpolation, resulting in an array of interpolated data valid at the desired value of the independent variable.

## Parameters

|                    |                                                                      |
|--------------------|----------------------------------------------------------------------|
| < <i>ind</i> >     | Value of the independent variable at which the data is interpolated. |
| < <i>results</i> > | Array of interpolated data.                                          |

## Returns

true on success, false (or throw) on failure.

Implemented in [LagrangeInterpolator](#).

## 8.33.3.13 IsInterpolationFeasible()

```
Integer Interpolator::IsInterpolationFeasible (
 Real ind) [virtual]
```

Checks if interpolation is feasible. Derived class should implement this method if any checking is done.

## Parameters

|            |                                         |
|------------|-----------------------------------------|
| <i>ind</i> | The value of the independent parameter. |
|------------|-----------------------------------------|

**Returns**

1 if feasible -1 if there is not enough data to interpolate -2 if requested data is not within the interpolation range

Reimplemented in [LagrangeInterpolator](#).

**8.33.3.14 operator=()**

```
Interpolator & Interpolator::operator= (
 const Interpolator & i)
```

Assignment operator

**Parameters**

|     |                                                                           |
|-----|---------------------------------------------------------------------------|
| <i> | <a href="#">Interpolator</a> that is used to set the values for this one. |
|-----|---------------------------------------------------------------------------|

**Returns**

this instance, configured like the input instance.

**8.33.3.15 SetExtrapolation()**

```
void Interpolator::SetExtrapolation (
 bool flag) [virtual]
```

**8.33.3.16 SetForceInterpolation()**

```
void Interpolator::SetForceInterpolation (
 bool flag) [virtual]
```

**8.33.3.17 SetRange()**

```
void Interpolator::SetRange () [protected]
```

Finds the minimum and maximum values of the independent variable.

**8.33.4 Member Data Documentation**

#### 8.33.4.1 allowExtrapolation

```
bool Interpolator::allowExtrapolation [protected]
```

Flag to allow extrapolation.

#### 8.33.4.2 bufferSize

```
Integer Interpolator::bufferSize [protected]
```

Number of points managed by the interpolator.

#### 8.33.4.3 dataIncreases

```
bool Interpolator::dataIncreases [protected]
```

Flag used to determine if independent variable increases or decreases.

#### 8.33.4.4 dependent

```
Real** Interpolator::dependent [protected]
```

The data that gets interpolated.

#### 8.33.4.5 dimension

```
Integer Interpolator::dimension [protected]
```

Number of dependent points to be interpolated.

#### 8.33.4.6 forceInterpolation

```
bool Interpolator::forceInterpolation [protected]
```

Flag used for additional feasibility checking.



#### 8.33.4.7 independent

`Real* Interpolator::independent` [protected]

Data array used for the independent variable.

#### 8.33.4.8 instanceName

`std::string Interpolator::instanceName` [protected]

The name of this interpolator.

#### 8.33.4.9 latestPoint

`Integer Interpolator::latestPoint` [protected]

Pointer to most recent point, for the ring buffer implementation.

#### 8.33.4.10 pointCount

`Integer Interpolator::pointCount` [protected]

Number of points fed to the interpolator.

#### 8.33.4.11 previousX

`Real Interpolator::previousX` [protected]

Previous independent value, used to determine direction data is going.

#### 8.33.4.12 range

`Real Interpolator::range[2]` [protected]

Valid range for the data points.

#### 8.33.4.13 rangeCalculated

```
bool Interpolator::rangeCalculated [protected]
```

Flag used to detect if range has already been calculated.

#### 8.33.4.14 requiredPoints

```
Integer Interpolator::requiredPoints [protected]
```

Number of points required to interpolate.

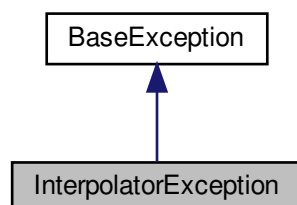
The documentation for this class was generated from the following files:

- GMATsrc/util/interpolator/[Interpolator.hpp](#)
- GMATsrc/util/interpolator/[Interpolator.cpp](#)

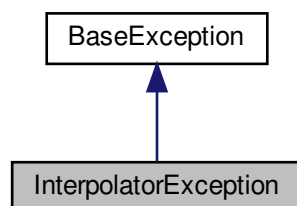
## 8.34 InterpolatorException Class Reference

```
#include <InterpolatorException.hpp>
```

Inheritance diagram for InterpolatorException:



Collaboration diagram for InterpolatorException:



## Public Member Functions

- [InterpolatorException](#) (std::string details="")
- [~InterpolatorException](#) ()

## Additional Inherited Members

### 8.34.1 Detailed Description

Exceptions thrown by the Interpolators

### 8.34.2 Constructor & Destructor Documentation

#### 8.34.2.1 InterpolatorException()

```
InterpolatorException::InterpolatorException (
 std::string details = "")
```

#### 8.34.2.2 ~InterpolatorException()

```
InterpolatorException::~~InterpolatorException ()
```

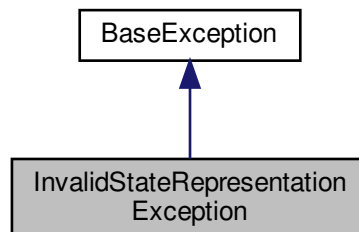
The documentation for this class was generated from the following files:

- GMATsrc/util/interpolator/[InterpolatorException.hpp](#)
- GMATsrc/util/interpolator/[InterpolatorException.cpp](#)

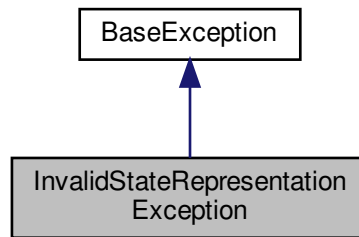
## 8.35 InvalidStateRepresentationException Class Reference

```
#include <BodyFixedStateConverter.hpp>
```

Inheritance diagram for InvalidStateRepresentationException:



Collaboration diagram for InvalidStateRepresentationException:



## Public Member Functions

- [InvalidStateRepresentationException](#) (const std::string &message="BodyFixedStateConverter: Conversion to invalid state representation requested: ")

## Additional Inherited Members

### 8.35.1 Detailed Description

Definition of the namespace containing methods to convert between celestial-body-fixed (CBF) state representations.

Cartesian states are (x,y,z) Spherical and Spherical-Ellipsoid states are (latitude, longitude, height)

### 8.35.2 Constructor & Destructor Documentation

#### 8.35.2.1 InvalidStateRepresentationException()

```
InvalidStateRepresentationException::InvalidStateRepresentationException (
 const std::string & message = "BodyFixedStateConverter: Conversion to invalid state representatio
: ") [inline]
```

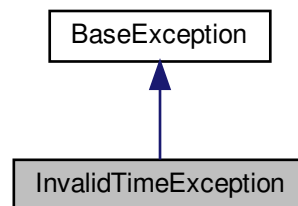
The documentation for this class was generated from the following file:

- GMATsrc/util/[BodyFixedStateConverter.hpp](#)

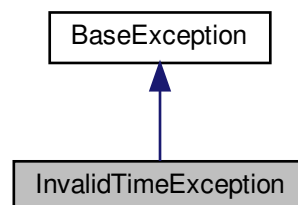
## 8.36 InvalidTimeException Class Reference

```
#include <TimeSystemConverter.hpp>
```

Inheritance diagram for InvalidTimeException:



Collaboration diagram for InvalidTimeException:



### Public Member Functions

- [InvalidTimeException](#) (const std::string &message="TimeSystemConverter: Requested time is invalid: ")

### Additional Inherited Members

#### 8.36.1 Constructor & Destructor Documentation

### 8.36.1.1 InvalidTimeException()

```
InvalidTimeException::InvalidTimeException (
 const std::string & message = "TimeSystemConverter: Requested time is invalid: "
) [inline]
```

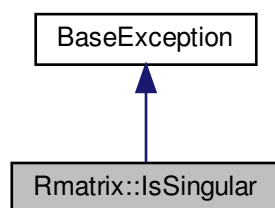
The documentation for this class was generated from the following file:

- GMATsrc/util/[TimeSystemConverter.hpp](#)

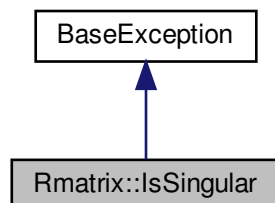
## 8.37 Rmatrix::IsSingular Class Reference

```
#include <Rmatrix.hpp>
```

Inheritance diagram for Rmatrix::IsSingular:



Collaboration diagram for Rmatrix::IsSingular:



### Public Member Functions

- [IsSingular](#) (const std::string &message="Rmatrix error: matrix is singular\')

## Additional Inherited Members

### 8.37.1 Constructor & Destructor Documentation

#### 8.37.1.1 IsSingular()

```
Rmatrix::IsSingular::IsSingular (
 const std::string & message = "Rmatrix error: matrix is singular\n") [inline]
```

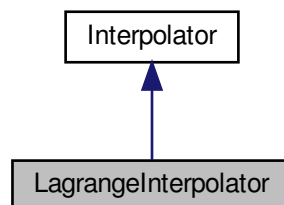
The documentation for this class was generated from the following file:

- GMATsrc/util/[Rmatrix.hpp](#)

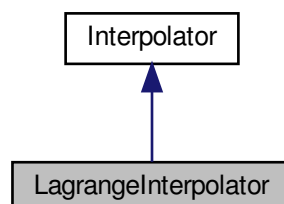
## 8.38 LagrangeInterpolator Class Reference

```
#include <LagrangeInterpolator.hpp>
```

Inheritance diagram for LagrangeInterpolator:



Collaboration diagram for LagrangeInterpolator:



## Public Member Functions

- [LagrangeInterpolator](#) (const std::string &name="", [Integer](#) dim=1, [Integer](#) ord=7)
- virtual [~LagrangeInterpolator](#) ()
- [LagrangeInterpolator](#) (const [LagrangeInterpolator](#) &li)
- [LagrangeInterpolator](#) & operator= (const [LagrangeInterpolator](#) &li)
- virtual [Integer](#) [IsInterpolationFeasible](#) ([Real](#) ind)
- virtual void [Clear](#) ()
- virtual bool [AddPoint](#) (const [Real](#) ind, const [Real](#) \*data)
- virtual bool [Interpolate](#) (const [Real](#) ind, [Real](#) \*results)
- virtual [Interpolator](#) \* [Clone](#) () const

## Protected Member Functions

- virtual void [AllocateArrays](#) ()
- virtual void [CleanupArrays](#) ()
- virtual void [CopyArrays](#) (const [LagrangeInterpolator](#) &i)
- void [BuildDataPoints](#) ([Real](#) ind)
- bool [UpdateBeginAndEndIndex](#) ([Real](#) ind)
- bool [IsDataNearCenter](#) ([Real](#) ind)
- [Integer](#) [FindStartingPoint](#) ([Real](#) ind)

## Protected Attributes

- [Integer](#) [order](#)  
*Order of interpolation.*
- [Integer](#) [actualSize](#)  
*Actual size to be used.*
- [Integer](#) [beginIndex](#)  
*Starting index used in finding center point.*
- [Integer](#) [endIndex](#)  
*Ending index used in finding center point.*
- [Integer](#) [dataIndex](#)  
*Index of nearest data point of requested data.*
- [Integer](#) [startPoint](#)  
*Starting index of interpolation range.*
- [Real](#) [lastX](#)  
*Value of the last point, to determine if the data buffer need updating.*
- [Real](#) \* [x](#)  
*Array of ordered independent variables used.*
- [Real](#) \*\* [y](#)  
*Array of ordered dependent variables used.*

## Static Protected Attributes

- static const [Integer](#) [MAX\\_BUFFER\\_SIZE](#) = 80

### 8.38.1 Detailed Description

Declares [LagrangeInterpolator](#) class as specified in the GMAT Math Spec.



## 8.38.2 Constructor & Destructor Documentation

### 8.38.2.1 LagrangeInterpolator() [1/2]

```
LagrangeInterpolator::LagrangeInterpolator (
 const std::string & name = "",
 Integer dim = 1,
 Integer ord = 7)
```

Implements [LagrangeInterpolator](#) class as specified in the GMAT Math Spec. Constructs lagrange interpolator (default constructor).

#### Parameters

|         |                                               |
|---------|-----------------------------------------------|
| <name>  | Name for this interpolator ("")               |
| <dim>   | Dimension of data that gets interpolated (1). |
| <order> | The order of interpolation (7)                |

### 8.38.2.2 ~LagrangeInterpolator()

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

Destroys lagrange interpolator (destructor).

### 8.38.2.3 LagrangeInterpolator() [2/2]

```
LagrangeInterpolator::LagrangeInterpolator (
 const LagrangeInterpolator & li)
```

Constructs lagrange interpolator, based on another (copy constructor).

#### Parameters

|           |                                    |
|-----------|------------------------------------|
| <i>li</i> | The original that is being copied. |
|-----------|------------------------------------|

## 8.38.3 Member Function Documentation

#### 8.38.3.1 AddPoint()

```
bool LagrangeInterpolator::AddPoint (
 const Real ind,
 const Real * data) [virtual]
```

See [Interpolator](#)

##### Exceptions

|               |                                         |
|---------------|-----------------------------------------|
| <i>thrown</i> | when independent data direction changes |
|---------------|-----------------------------------------|

Reimplemented from [Interpolator](#).

#### 8.38.3.2 AllocateArrays()

```
void LagrangeInterpolator::AllocateArrays () [protected], [virtual]
```

Allocates lagrange buffers and calls the base method to build the ring buffer.

Reimplemented from [Interpolator](#).

#### 8.38.3.3 BuildDataPoints()

```
void LagrangeInterpolator::BuildDataPoints (
 Real ind) [protected]
```

Use the ring buffer to load the arrays used to build the lagrange buffer. Lagrange buffer should be constructed so that requested data sits in the middle of the buffer as possible.

#### 8.38.3.4 CleanupArrays()

```
void LagrangeInterpolator::CleanupArrays () [protected], [virtual]
```

Frees the memory used by the lagrange buffer and calls the base method to manage the ring buffer.

Reimplemented from [Interpolator](#).

#### 8.38.3.5 Clear()

```
void LagrangeInterpolator::Clear () [virtual]
```

##### See also

[Interpolator](#)

Reimplemented from [Interpolator](#).

## 8.38.3.6 Clone()

```
Interpolator * LagrangeInterpolator::Clone () const [virtual]
```

This method returns a clone of the [LagrangeInterpolator](#).

## Returns

clone of the [LagrangeInterpolator](#).

Implements [Interpolator](#).

## 8.38.3.7 CopyArrays()

```
void LagrangeInterpolator::CopyArrays (
 const LagrangeInterpolator & i) [protected], [virtual]
```

Copies the ring buffer from one [Interpolator](#) to this one.

## Parameters

|          |                                                                             |
|----------|-----------------------------------------------------------------------------|
| <i>i</i> | The <a href="#">Interpolator</a> that supplies the data copied to this one. |
|----------|-----------------------------------------------------------------------------|

## 8.38.3.8 FindStartingPoint()

```
Integer LagrangeInterpolator::FindStartingPoint (
 Real ind) [protected]
```

## 8.38.3.9 Interpolate()

```
bool LagrangeInterpolator::Interpolate (
 const Real ind,
 Real * results) [virtual]
```

Perform the interpolation.

This method is the core interface for the lagrange interpolation. See the GMAT math spec for the algorithm.

## Parameters

|                |                                         |
|----------------|-----------------------------------------|
| <i>ind</i>     | The value of the independent parameter. |
| <i>results</i> | Data structure for the estimates.       |

**Returns**

true on success, false on failure.

Implements [Interpolator](#).

**8.38.3.10 IsDataNearCenter()**

```
bool LagrangeInterpolator::IsDataNearCenter (
 Real ind) [protected]
```

Checks if requested data is centered in the buffer.

**8.38.3.11 IsInterpolationFeasible()**

```
Integer LagrangeInterpolator::IsInterpolationFeasible (
 Real ind) [virtual]
```

Checks if interpolation is feasible. (Should I also check if ind is in the center of the interpolation range?)

**Parameters**

|            |                                         |
|------------|-----------------------------------------|
| <i>ind</i> | The value of the independent parameter. |
|------------|-----------------------------------------|

**Returns**

1 if feasible -1 if there is not enough data to interpolate -2 if requested data is before the first data -3 if requested data is after the last data

Reimplemented from [Interpolator](#).

**8.38.3.12 operator=()**

```
LagrangeInterpolator & LagrangeInterpolator::operator= (
 const LagrangeInterpolator & li)
```

Sets this lagrange interpolator to match another (assignment operator).

**Parameters**

|           |                                    |
|-----------|------------------------------------|
| <i>li</i> | The original that is being copied. |
|-----------|------------------------------------|

### Returns

A reference to the copy (aka \*this).

#### 8.38.3.13 UpdateBeginAndEndIndex()

```
bool LagrangeInterpolator::UpdateBeginAndEndIndex (
 Real ind) [protected]
```

Finds starting and ending index to use for interpolation.

### Returns

true if indices are valid, false otherwise

## 8.38.4 Member Data Documentation

### 8.38.4.1 actualSize

```
Integer LagrangeInterpolator::actualSize [protected]
```

Actual size to be used.

### 8.38.4.2 beginIndex

```
Integer LagrangeInterpolator::beginIndex [protected]
```

Starting index used in finding center point.

### 8.38.4.3 dataIndex

```
Integer LagrangeInterpolator::dataIndex [protected]
```

Index of nearest data point of requested data.

#### 8.38.4.4 endIndex

`Integer LagrangeInterpolator::endIndex [protected]`

Ending index used in finding center point.

#### 8.38.4.5 lastX

`Real LagrangeInterpolator::lastX [protected]`

Value of the last point, to determine if the data buffer need updating.

#### 8.38.4.6 MAX\_BUFFER\_SIZE

`const Integer LagrangeInterpolator::MAX_BUFFER_SIZE = 80 [static], [protected]`

#### 8.38.4.7 order

`Integer LagrangeInterpolator::order [protected]`

Order of interpolation.

#### 8.38.4.8 startPoint

`Integer LagrangeInterpolator::startPoint [protected]`

Starting index of interpolation range.

#### 8.38.4.9 x

`Real* LagrangeInterpolator::x [protected]`

Array of ordered independent variables used.

## 8.38.4.10 y

`Real** LagrangeInterpolator::y` [protected]

Array of ordered dependent variables used.

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

- GMATsrc/util/interpolator/[LagrangeInterpolator.hpp](#)
- GMATsrc/util/interpolator/[LagrangeInterpolator.cpp](#)

## 8.39 LeapSecondInformation Struct Reference

```
#include <LeapSecsFileReader.hpp>
```

### Public Attributes

- [Real julianDate](#)
- [Real taiMJD](#)
- [Real offset1](#)
- [Real offset2](#)
- [Real offset3](#)

### 8.39.1 Detailed Description

Reads time coefficient file, creates a table of coefficients and converts to the utc.

File found at : <ftp://maia.usno.navy.mil/ser7/tai-utc.dat> Structure defining internal leap second information.

Moved here from inside of the [LeapSecsFileReader](#) class to clean up import/ export issues with Visual Studio

### 8.39.2 Member Data Documentation

#### 8.39.2.1 julianDate

`Real LeapSecondInformation::julianDate`

#### 8.39.2.2 offset1

`Real LeapSecondInformation::offset1`

### 8.39.2.3 offset2

`Real LeapSecondInformation::offset2`

### 8.39.2.4 offset3

`Real LeapSecondInformation::offset3`

### 8.39.2.5 taiMJD

`Real LeapSecondInformation::taiMJD`

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

- [GMATsrc/util/LeapSecsFileReader.hpp](#)

## 8.40 LeapSecsFileReader Class Reference

```
#include <LeapSecsFileReader.hpp>
```

### Public Member Functions

- [LeapSecsFileReader](#) (const std::string &fileName="tai-utc.dat")
- virtual [~LeapSecsFileReader](#) ()
- [LeapSecsFileReader](#) (const [LeapSecsFileReader](#) &lsfr)
- [LeapSecsFileReader](#) & [operator=](#) (const [LeapSecsFileReader](#) &lsfr)
- bool [Initialize](#) ()
- [Real](#) [NumberOfLeapSecondsFrom](#) ([UtcMjd](#) utcMjd)
- [Real](#) [GetFirstLeapSecondMJD](#) ([Real](#) fromUtcMjd, [Real](#) toUtcMjd)
- bool [IsInLeapSecond](#) ([Real](#) theTaiMjd)

### 8.40.1 Constructor & Destructor Documentation



**8.40.1.1 LeapSecsFileReader()** [1/2]

```
LeapSecsFileReader::LeapSecsFileReader (
 const std::string & fileName = "tai-utc.dat")
```

Reads time coefficient file, creates a table of coefficients and converts to the utc.

File found at : <ftp://maia.usno.navy.mil/ser7/tai-utc.dat>

**Note**

The MJD-JD offset used is [GmatTimeConstants::JD\\_NOV\\_17\\_1858](#) Constructor.

**8.40.1.2 ~LeapSecsFileReader()**

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

Destructor.

**8.40.1.3 LeapSecsFileReader()** [2/2]

```
LeapSecsFileReader::LeapSecsFileReader (
 const LeapSecsFileReader & lsfr)
```

**8.40.2 Member Function Documentation****8.40.2.1 GetFirstLeapSecondMJD()**

```
Real LeapSecsFileReader::GetFirstLeapSecondMJD (
 Real fromUtcMjd,
 Real toUtcMjd)
```

Returns UTCMJD of first leap seconds occurred between fromUtcMjd and toUtcMjd. If file is not read or fromUtcMjd is greater than toUtcMjd, or no leap seconds occurred between input dates, -1 is returned.

**Returns**

First date of leap seconds occurred between two input dates

**Note**

Assumes that JD from table is utcjd.

#### 8.40.2.2 Initialize()

```
bool LeapSecsFileReader::Initialize ()
```

#### 8.40.2.3 IsInLeapSecond()

```
bool LeapSecsFileReader::IsInLeapSecond (
 Real theTaiMjd)
```

Determines whether or not the input time (in TAI MJD, referenced to [GmatTimeConstants::JD\\_MJD\\_OFFSET](#)) is in a leap second

#### 8.40.2.4 NumberOfLeapSecondsFrom()

```
Real LeapSecsFileReader::NumberOfLeapSecondsFrom (
 UtcMjd utcMjd)
```

Converts utcMjd to utcjd and then looks it up from the table. If file is not read, 0 is returned.

#### Returns

number of leap seconds

#### Note

Assumes that JD from table is utcjd.

#### 8.40.2.5 operator=()

```
LeapSecsFileReader& LeapSecsFileReader::operator= (
 const LeapSecsFileReader & lsfr)
```

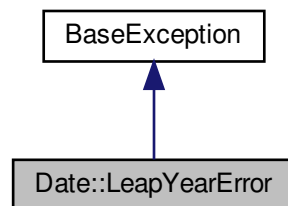
The documentation for this class was generated from the following files:

- [GMATsrc/util/LeapSecsFileReader.hpp](#)
- [GMATsrc/util/LeapSecsFileReader.cpp](#)

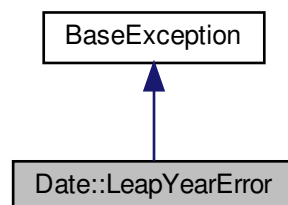
## 8.41 Date::LeapYearError Class Reference

```
#include <Date.hpp>
```

Inheritance diagram for Date::LeapYearError:



Collaboration diagram for Date::LeapYearError:



### Public Member Functions

- [LeapYearError](#) (const std::string &message="Date error: day number is invalid for specified year")

### Additional Inherited Members

#### 8.41.1 Constructor & Destructor Documentation

#### 8.41.1.1 LeapYearError()

```
Date::LeapYearError::LeapYearError (
 const std::string & message = "Date error: day number is invalid for specified year"
) [inline]
```

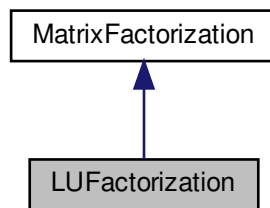
The documentation for this class was generated from the following file:

- GMATsrc/util/[Date.hpp](#)

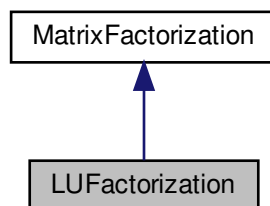
## 8.42 LUFactorization Class Reference

```
#include <LUFactorization.hpp>
```

Inheritance diagram for LUFactorization:



Collaboration diagram for LUFactorization:



### Public Member Functions

- [LUFactorization](#) (bool pivotOption=true)
- [LUFactorization](#) (const [LUFactorization](#) &lufactorization)
- [~LUFactorization](#) ()
- [LUFactorization & operator=](#) (const [LUFactorization](#) &lufactorization)
- void [Factor](#) (const [Rmatrix](#) inputMatrix, [Rmatrix](#) &L, [Rmatrix](#) &U)
- void [Invert](#) ([Rmatrix](#) &inputMatrix)
- void [SolveSystem](#) (const [Rmatrix](#) inputMatrix, [Rvector](#) b, [Rvector](#) &x)
- [Real Determinant](#) ([Rmatrix](#) A)

## Additional Inherited Members

### 8.42.1 Detailed Description

Declares LU factorization methods. This class strictly uses [Rvector](#) and [Rmatrix](#) from [Gmat](#) code for vector and matrix operations.

### 8.42.2 Constructor & Destructor Documentation

#### 8.42.2.1 LUFactorization() [1/2]

```
LUFactorization::LUFactorization (
 bool pivotOption = true)
```

Defines LU factorization methods. This class strictly uses [Rvector](#) and [Rmatrix](#) from [Gmat](#) code for vector and matrix operations. Constructor

##### Parameters

|                    |                                                                                                                                                                                                                                                                   |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>pivotOption</i> | True or false setting to determine whether to use or not use partial pivoting in calculations, defaulted to False. Note that pivoting should only be used when a diagonal value close or equal to zero is expected. Otherwise, it is a waste of computation time. |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

#### 8.42.2.2 LUFactorization() [2/2]

```
LUFactorization::LUFactorization (
 const LUFactorization & lufactorization)
```

Copy Constructor

#### 8.42.2.3 ~LUFactorization()

```
LUFactorization::~~LUFactorization ()
```

Class destructor

### 8.42.3 Member Function Documentation

#### 8.42.3.1 Determinant()

```
Real LUFactorization::Determinant (
 Rmatrix A)
```

Method that determines the determinant of a square matrix

## Parameters

|          |                                                        |
|----------|--------------------------------------------------------|
| <i>A</i> | The square matrix a determinant will be calculated for |
|----------|--------------------------------------------------------|

## Returns

det The determinant of the matrix

## 8.42.3.2 Factor()

```
void LUFactorization::Factor (
 const Rmatrix inputMatrix,
 Rmatrix & L,
 Rmatrix & U) [virtual]
```

Method used to factor the matrix with LU factorization ( $A = LU$ ). This method is based off algorithm 3.4.1 from Gene H. Golub and Charles F. Van Loan.

## Parameters

|                    |                                  |
|--------------------|----------------------------------|
| <i>inputMatrix</i> | The matrix that will be factored |
| <i>&amp;L</i>      | The lower triangular matrix      |
| <i>&amp;U</i>      | The upper triangular matrix      |

Implements [MatrixFactorization](#).

## 8.42.3.3 Invert()

```
void LUFactorization::Invert (
 Rmatrix & inputMatrix) [virtual]
```

Method used to invert the matrix using LU factorization

## Parameters

|                         |                                  |
|-------------------------|----------------------------------|
| <i>&amp;inputMatrix</i> | The matrix that will be inverted |
|-------------------------|----------------------------------|

Implements [MatrixFactorization](#).

## 8.42.3.4 operator=()

```
LUFactorization & LUFactorization::operator= (
 const LUFactorization & lufactorization)
```

Assignment operator

#### 8.42.3.5 SolveSystem()

```
void LUFactorization::SolveSystem (
 const Rmatrix inputMatrix,
 Rvector b,
 Rvector & x)
```

Method used to solve system of equations using LU factorization ( $Ax = b$ ). For square systems, a combination of algorithms 3.1.1 and 3.1.2 from Gene H. Golub and Charles F. Van Loan are used with matrices L and U. When an underdetermined system is input, algorithm 5.7.2 from Gene H. Golub and Charles F. Van Loan is used to find the minimum 2-norm solution.

#### Parameters

|                    |                                                                       |
|--------------------|-----------------------------------------------------------------------|
| <i>inputMatrix</i> | The matrix resembling coefficients of equations                       |
| <i>b</i>           | Column vector representing RHS values of equations from inputMatrix   |
| <i>x</i>           | Column vector containing solution values to solve system of equations |

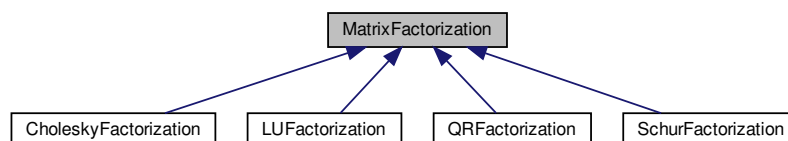
The documentation for this class was generated from the following files:

- GMATsrc/util/matrixoperations/LUFactorization.hpp
- GMATsrc/util/matrixoperations/LUFactorization.cpp

## 8.43 MatrixFactorization Class Reference

```
#include <MatrixFactorization.hpp>
```

Inheritance diagram for MatrixFactorization:



#### Public Member Functions

- [MatrixFactorization \(\)](#)
- [MatrixFactorization \(const MatrixFactorization &matrixfactorization\)](#)
- [~MatrixFactorization \(\)](#)
- [MatrixFactorization & operator= \(const MatrixFactorization &matrixfactorization\)](#)
- virtual void [Invert \(Rmatrix &inputMatrix\)=0](#)
- virtual void [Factor \(const Rmatrix inputMatrix, Rmatrix &output1, Rmatrix &output2\)=0](#)

## Static Public Member Functions

- static [Rmatrix CompressNormalMatrix](#) (const [Rmatrix](#) &infMatrix, [IntegerArray](#) &removedIndexes, [IntegerArray](#) &auxVector, [Integer](#) &numRemoved)
- static [Rmatrix ExpandNormalMatrixInverse](#) (const [Rmatrix](#) &covMatrix, const [IntegerArray](#) &auxVector, const [Integer](#) &numRemoved)
- static [Integer PackedArrayIndex](#) ([Integer](#) n, [Integer](#) i, [Integer](#) j)

### 8.43.1 Detailed Description

Declares [MatrixFactorization](#) class.

### 8.43.2 Constructor & Destructor Documentation

#### 8.43.2.1 [MatrixFactorization\(\)](#) [1/2]

```
MatrixFactorization::MatrixFactorization ()
```

Declares [MatrixFactorization](#) class. Constructor

#### 8.43.2.2 [MatrixFactorization\(\)](#) [2/2]

```
MatrixFactorization::MatrixFactorization (
 const MatrixFactorization & matrixfactorization)
```

Copy constructor

#### 8.43.2.3 [~MatrixFactorization\(\)](#)

```
MatrixFactorization::~~MatrixFactorization ()
```

Destructor

### 8.43.3 Member Function Documentation

#### 8.43.3.1 [CompressNormalMatrix\(\)](#)

```
Rmatrix MatrixFactorization::CompressNormalMatrix (
 const Rmatrix & infMatrix,
 IntegerArray & removedIndexes,
 IntegerArray & auxVector,
 Integer & numRemoved) [static]
```

Reduce the dimensionality of an N x N symmetric matrix by 1 for each row / column of zeros in the matrix. If no reduction is done, numRemoved is set to 0, and a copy of infMatrix is returned. If reduction is done, numRemoved is set to the number of dimensions removed from the matrix, and an N-numRemoved x N-numRemoved matrix is returned, with the zero rows and columns of infMatrix removed. auxVector is an array of additional data which is used by [ExpandNormalMatrixInverse\(\)](#) to restore the infMatrix, after inversion, back to N x N dimensions, by filling in rows and columns of 0s, where they were removed during the reduction. removedIndexes is set to contain the indexes of the rows/columns that were removed.

auxVector - an N element integer array. Each element i of auxVector which corresponds to a row/column which was removed is set to -1. Otherwise auxVector[i] contains the number of row/columns removed with index less than i.



## Parameters

|                       |                                                                                                                                                                           |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>infMatrix</i>      | The square symmetric matrix for which rows/columns of zeros are to be removed                                                                                             |
| <i>removedIndexes</i> | The indexes of the rows/columns that were removed                                                                                                                         |
| <i>auxVector</i>      | A vector of additional data, used to restore the inverse of the reduced matrix back to its original dimensions, by adding rows/columns of 0s back where they were removed |
| <i>numRemoved</i>     | the number of rows/columns removed from infMatrix                                                                                                                         |

## Returns

The infMatrix with rows/columns of zeros removed.

## 8.43.3.2 ExpandNormalMatrixInverse()

```
Rmatrix MatrixFactorization::ExpandNormalMatrixInverse (
 const Rmatrix & covMatrix,
 const IntegerArray & auxVector,
 const Integer & numRemoved) [static]
```

Given the inverse of an N x N symmetric normal matrix, which had numRemoved rows/columns of zeros removed before inversion, return a copy of the matrix which has been expanded back to size N x N by inserting rows/columns of 0s at the indexes where they were removed from the normal matrix.

## Parameters

|                   |                                                                                                                      |
|-------------------|----------------------------------------------------------------------------------------------------------------------|
| <i>covMatrix</i>  | The N-numRemoved x N-numRemoved inverse of the normal matrix                                                         |
| <i>auxVector</i>  | The auxiliary vector which was returned by <a href="#">CompressNormalMatrix()</a> when the normal matrix was reduced |
| <i>numRemoved</i> | The number of rows/columns which were removed when the normal matrix was reduced, and will be restored to covMatrix  |

## Returns

A copy of covMatrix, which has expanded back to size N x N by inserting rows/columns of 0s at the indexes where they were removed from the normal matrix.

## 8.43.3.3 Factor()

```
virtual void MatrixFactorization::Factor (
 const Rmatrix inputMatrix,
 Rmatrix & output1,
 Rmatrix & output2) [pure virtual]
```

Implemented in [LUFactorization](#), [CholeskyFactorization](#), [QRFactorization](#), and [SchurFactorization](#).

#### 8.43.3.4 Invert()

```
virtual void MatrixFactorization::Invert (
 Rmatrix & inputMatrix) [pure virtual]
```

Implemented in [QRFactorization](#), [CholeskyFactorization](#), [LUFactorization](#), and [SchurFactorization](#).

#### 8.43.3.5 operator=()

```
MatrixFactorization & MatrixFactorization::operator= (
 const MatrixFactorization & matrixfactorization)
```

Assignment operator

#### 8.43.3.6 PackedArrayIndex()

```
Integer MatrixFactorization::PackedArrayIndex (
 Integer N,
 Integer row,
 Integer col) [static]
```

Given the upper triangle of an N x N symmetric matrix stored in a linear array, and given a (row, column) index in the upper triangle,  $0 \leq \text{row} < N$ ,  $\text{row} \leq \text{col} < N$ , return the index in the linear array where that element is stored.

Note that for the array, matrix elements (0, 0) to (0, N-1) are stored in elements [0] to [N-1] of the array, matrix elements (1, 1) to (1, N-1) are store in elements [N] to [2N-2], etc.

##### Parameters

|            |                                                                     |
|------------|---------------------------------------------------------------------|
| <i>N</i>   | The number of rows and columns in the square symmetric matrix       |
| <i>row</i> | The row index of the element. $0 \leq \text{row} < N$ .             |
| <i>col</i> | The column index of the element. $\text{row} \leq \text{col} < N$ . |

##### Returns

The array index for matrix element (row, col)

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

- GMATsrc/util/matrixoperations/[MatrixFactorization.hpp](#)
- GMATsrc/util/matrixoperations/[MatrixFactorization.cpp](#)

## 8.44 MessageInterface Class Reference

```
#include <MessageInterface.hpp>
```

## Static Public Member Functions

- static bool [SetMessageReceiver](#) ([MessageReceiver](#) \*mr)
- static [MessageReceiver](#) \* [GetMessageReceiver](#) ()
- static void [ShowMessage](#) (const std::string &msg)
- static void [ShowMessage](#) (const char \*format,...)
- static void [PopupMessage](#) ([Gmat::MessageType](#) msgType, const std::string &msg)
- static void [PopupMessage](#) ([Gmat::MessageType](#) msgType, const char \*format,...)
- static std::string [GetLogFileName](#) ()
- static bool [GetLogEnable](#) ()
- static void [SetLogEnable](#) (bool flag)
- static void [SetLogPath](#) (const char \*pathname, bool append=false)
- static void [SetLogPath](#) (const std::string &pathname, bool append=false)
- static void [SetLogFile](#) (const std::string &filename)
- static void [LogMessage](#) (const std::string &msg)
- static void [LogMessage](#) (const char \*format,...)
- static void [ClearMessage](#) ()
- static std::string [GetQueuedMessage](#) ()
- static void [PutMessage](#) (const std::string &msg)
- static void [PutMessage](#) (const char \*format,...)
- static void [ClearMessageQueue](#) ()

## Static Public Attributes

- static const int [MAX\\_MESSAGE\\_LENGTH](#) = 30000

### 8.44.1 Detailed Description

Defines operations on messages. The [MessageInterface](#) class defines static methods that are called to send messages to the user. [MessageInterface](#) passes these messages to an implementation specific class derived from the abstract [MessageReceiver](#) class. Display to the user is handled in the derived [MessageReceiver](#).

### 8.44.2 Member Function Documentation

#### 8.44.2.1 ClearMessage()

```
void MessageInterface::ClearMessage () [static]
```

Tells the [MessageReceiver](#) to clear the message window.

#### 8.44.2.2 ClearMessageQueue()

```
void MessageInterface::ClearMessageQueue () [static]
```

Tells the [MessageReceiver](#) to clear the message queue.

#### 8.44.2.3 GetLogEnable()

```
bool MessageInterface::GetLogEnable () [static]
```

Is logging on or off.

#### 8.44.2.4 GetLogFileName()

```
std::string MessageInterface::GetLogFileName () [static]
```

Retrieves the fully qualified name of the log file from the [MessageReceiver](#).

##### Returns

The name of the log file, including path information.

#### 8.44.2.5 GetMessageReceiver()

```
MessageReceiver * MessageInterface::GetMessageReceiver () [static]
```

#### 8.44.2.6 GetQueuedMessage()

```
std::string MessageInterface::GetQueuedMessage () [static]
```

Tells the [MessageReceiver](#) to retrieve all message from the queue.

#### 8.44.2.7 LogMessage() [1/2]

```
void MessageInterface::LogMessage (
 const std::string & msg) [static]
```

Sends a message to the [MessageReceiver](#) for logging.

##### Parameters

|            |              |
|------------|--------------|
| <i>msg</i> | The message. |
|------------|--------------|

#### 8.44.2.8 LogMessage() [2/2]

```
void MessageInterface::LogMessage (
```

```
const char * format,
...) [static]
```

Sends a variable argument message to the [MessageReceiver](#) for logging.

#### Parameters

|               |                                                                            |
|---------------|----------------------------------------------------------------------------|
| <i>format</i> | The format, possibly including markers for variable argument substitution. |
| ...           | The optional list of parameters that are inserted into the format string.  |

#### 8.44.2.9 PopupMessage() [1/2]

```
void MessageInterface::PopupMessage (
 Gmat::MessageType msgType,
 const std::string & msg) [static]
```

Passes a popup message to the [MessageReceiver](#).

#### Parameters

|                |                                                                                                                                          |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <i>msgType</i> | The type of message that is displayed, selected from the set {ERROR_, WARNING_, INFO_} enumerated in the <a href="#">Gmat</a> namespace. |
| <i>msg</i>     | The message.                                                                                                                             |

#### 8.44.2.10 PopupMessage() [2/2]

```
void MessageInterface::PopupMessage (
 Gmat::MessageType msgType,
 const char * format,
 ...) [static]
```

Passes a variable argument delimited popup message to the [MessageReceiver](#).

#### Parameters

|                |                                                                                                                                          |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <i>msgType</i> | The type of message that is displayed, selected from the set {ERROR_, WARNING_, INFO_} enumerated in the <a href="#">Gmat</a> namespace. |
| <i>format</i>  | The format, possibly including markers for variable argument substitution.                                                               |
| ...            | The optional list of parameters that are inserted into the format string.                                                                |

#### 8.44.2.11 PutMessage() [1/2]

```
void MessageInterface::PutMessage (
```

```
const std::string & msg) [static]
```

Tells the [MessageReceiver](#) to push the message into queue

#### 8.44.2.12 PutMessage() [2/2]

```
void MessageInterface::PutMessage (
 const char * format,
 ...) [static]
```

Tells the [MessageReceiver](#) to push the message into queue

#### 8.44.2.13 SetLogEnable()

```
void MessageInterface::SetLogEnable (
 bool flag) [static]
```

Tells the [MessageReceiver](#) to turn logging on or off.

##### Parameters

|             |                                                                                                       |
|-------------|-------------------------------------------------------------------------------------------------------|
| <i>flag</i> | The new logging state – true enables logging, and false disables it. The logging state is idempotent. |
|-------------|-------------------------------------------------------------------------------------------------------|

#### 8.44.2.14 SetLogFile()

```
void MessageInterface::SetLogFile (
 const std::string & filename) [static]
```

#### 8.44.2.15 SetLogPath() [1/2]

```
void MessageInterface::SetLogPath (
 const char * pathname,
 bool append = false) [static]
```

#### 8.44.2.16 SetLogPath() [2/2]

```
void MessageInterface::SetLogPath (
 const std::string & pathname,
 bool append = false) [static]
```

#### 8.44.2.17 SetMessageReceiver()

```
bool MessageInterface::SetMessageReceiver (
 MessageReceiver * mr) [static]
```

#### 8.44.2.18 ShowMessage() [1/2]

```
void MessageInterface::ShowMessage (
 const std::string & msgString) [static]
```

Passes an std::string message to the [MessageReceiver](#).

##### Parameters

|                  |                                |
|------------------|--------------------------------|
| <i>msgString</i> | The message that is displayed. |
|------------------|--------------------------------|

#### 8.44.2.19 ShowMessage() [2/2]

```
void MessageInterface::ShowMessage (
 const char * format,
 ...) [static]
```

Passes a variable argument delimited message to the [MessageReceiver](#).

##### Parameters

|               |                                                                            |
|---------------|----------------------------------------------------------------------------|
| <i>format</i> | The format, possibly including markers for variable argument substitution. |
| ...           | The optional list of parameters that are inserted into the format string.  |

### 8.44.3 Member Data Documentation

#### 8.44.3.1 MAX\_MESSAGE\_LENGTH

```
const int MessageInterface::MAX_MESSAGE_LENGTH = 30000 [static]
```

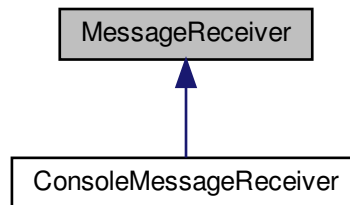
The documentation for this class was generated from the following files:

- GMATsrc/util/[MessageInterface.hpp](#)
- GMATsrc/util/[MessageInterface.cpp](#)

## 8.45 MessageReceiver Class Reference

```
#include <MessageReceiver.hpp>
```

Inheritance diagram for MessageReceiver:



### Public Member Functions

- virtual void [ShowMessage](#) (const std::string &msg)=0
- virtual void [ShowMessage](#) (const char \*msg,...)=0
- virtual void [PopupMessage](#) (Gmat::MessageType msgType, const std::string &msg)=0
- virtual void [PopupMessage](#) (Gmat::MessageType msgType, const char \*msg,...)=0
- virtual std::string [GetLogFileName](#) ()=0
- virtual bool [GetLogEnable](#) ()=0
- virtual void [SetLogEnable](#) (bool flag)=0
- virtual void [SetLogPath](#) (const std::string &pathname, bool append=false)=0
- virtual void [SetLogFile](#) (const std::string &filename)=0
- virtual void [LogMessage](#) (const std::string &msg)=0
- virtual void [LogMessage](#) (const char \*msg,...)=0
- virtual void [ClearMessage](#) ()=0
- virtual std::string [GetMessage](#) ()=0
- virtual void [PutMessage](#) (const std::string &msg)=0
- virtual void [ClearMessageQueue](#) ()=0

### Protected Member Functions

- [MessageReceiver](#) ()
- virtual [~MessageReceiver](#) ()
- bool [IsValidLogFile](#) (const std::string fullLogFilePath)
- std::string [GetLogFileText](#) ()

#### 8.45.1 Detailed Description

Defines output operations for messages. This is the abstract base class for these operations. [MessageReceiver](#) is an abstract class designed to receive messages from GMAT's components and display them to the user. Specific implementations of GMAT derive a class from the [MessageReceiver](#) class that implements the abstract methods to present messages to the user using an appropriate venue – the console for console applications, the GUI for GUI based applications, and other methods as deemed appropriate for other implementations.

The derived classes are best implemented as singletons.



## 8.45.2 Constructor & Destructor Documentation

### 8.45.2.1 MessageReceiver()

```
MessageReceiver::MessageReceiver () [protected]
```

Implements output operations for messages. This is the abstract base class for these operations. Constructor

### 8.45.2.2 ~MessageReceiver()

```
MessageReceiver::~MessageReceiver () [protected], [virtual]
```

Destructor

## 8.45.3 Member Function Documentation

### 8.45.3.1 ClearMessage()

```
virtual void MessageReceiver::ClearMessage () [pure virtual]
```

Implemented in [ConsoleMessageReceiver](#).

### 8.45.3.2 ClearMessageQueue()

```
virtual void MessageReceiver::ClearMessageQueue () [pure virtual]
```

Implemented in [ConsoleMessageReceiver](#).

### 8.45.3.3 GetLogEnable()

```
virtual bool MessageReceiver::GetLogEnable () [pure virtual]
```

Implemented in [ConsoleMessageReceiver](#).

#### 8.45.3.4 GetLogFileName()

```
virtual std::string MessageReceiver::GetLogFileName () [pure virtual]
```

Implemented in [ConsoleMessageReceiver](#).

#### 8.45.3.5 GetLogFileText()

```
std::string MessageReceiver::GetLogFileText () [protected]
```

#### 8.45.3.6 GetMessage()

```
virtual std::string MessageReceiver::GetMessage () [pure virtual]
```

Implemented in [ConsoleMessageReceiver](#).

#### 8.45.3.7 IsValidLogFile()

```
bool MessageReceiver::IsValidLogFile (
 const std::string fullLogFilePath) [protected]
```

#### 8.45.3.8 LogMessage() [1/2]

```
virtual void MessageReceiver::LogMessage (
 const std::string & msg) [pure virtual]
```

Implemented in [ConsoleMessageReceiver](#).

#### 8.45.3.9 LogMessage() [2/2]

```
virtual void MessageReceiver::LogMessage (
 const char * msg,
 ...) [pure virtual]
```

Implemented in [ConsoleMessageReceiver](#).

**8.45.3.10 PopupMessage()** [1/2]

```
virtual void MessageReceiver::PopupMessage (
 Gmat::MessageType msgType,
 const std::string & msg) [pure virtual]
```

Implemented in [ConsoleMessageReceiver](#).

**8.45.3.11 PopupMessage()** [2/2]

```
virtual void MessageReceiver::PopupMessage (
 Gmat::MessageType msgType,
 const char * msg,
 ...) [pure virtual]
```

Implemented in [ConsoleMessageReceiver](#).

**8.45.3.12 PutMessage()**

```
virtual void MessageReceiver::PutMessage (
 const std::string & msg) [pure virtual]
```

Implemented in [ConsoleMessageReceiver](#).

**8.45.3.13 SetLogEnable()**

```
virtual void MessageReceiver::SetLogEnable (
 bool flag) [pure virtual]
```

Implemented in [ConsoleMessageReceiver](#).

**8.45.3.14 SetLogFile()**

```
virtual void MessageReceiver::SetLogFile (
 const std::string & filename) [pure virtual]
```

Implemented in [ConsoleMessageReceiver](#).

#### 8.45.3.15 SetLogPath()

```
virtual void MessageReceiver::SetLogPath (
 const std::string & pathname,
 bool append = false) [pure virtual]
```

Implemented in [ConsoleMessageReceiver](#).

#### 8.45.3.16 ShowMessage() [1/2]

```
virtual void MessageReceiver::ShowMessage (
 const std::string & msg) [pure virtual]
```

Implemented in [ConsoleMessageReceiver](#).

#### 8.45.3.17 ShowMessage() [2/2]

```
virtual void MessageReceiver::ShowMessage (
 const char * msg,
 ...) [pure virtual]
```

Implemented in [ConsoleMessageReceiver](#).

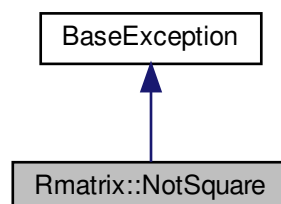
The documentation for this class was generated from the following files:

- GMATsrc/util/[MessageReceiver.hpp](#)
- GMATsrc/util/[MessageReceiver.cpp](#)

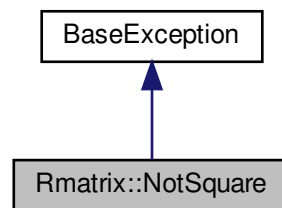
## 8.46 Rmatrix::NotSquare Class Reference

```
#include <Rmatrix.hpp>
```

Inheritance diagram for Rmatrix::NotSquare:



Collaboration diagram for Rmatrix::NotSquare:



### Public Member Functions

- [NotSquare](#) (const std::string &message="Rmatrix error: matrix not square\')

### Additional Inherited Members

#### 8.46.1 Constructor & Destructor Documentation

##### 8.46.1.1 NotSquare()

```
Rmatrix::NotSquare::NotSquare (
 const std::string & message = "Rmatrix error: matrix not square\n") [inline]
```

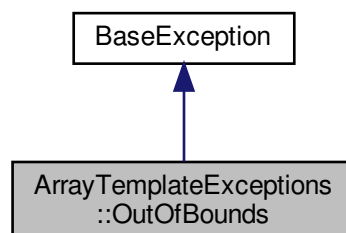
The documentation for this class was generated from the following file:

- GMATsrc/util/[Rmatrix.hpp](#)

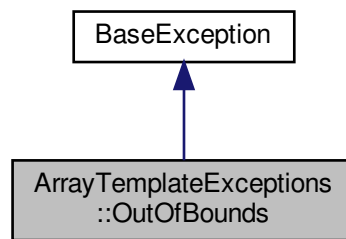
## 8.47 ArrayTemplateExceptions::OutOfBounds Class Reference

```
#include <ArrayTemplate.hpp>
```

Inheritance diagram for ArrayTemplateExceptions::OutOfBounds:



Collaboration diagram for ArrayTemplateExceptions::OutOfBounds:



### Public Member Functions

- [OutOfBounds](#) (const std::string &message="ArrayTemplate error : out-of-bounds.")

### Additional Inherited Members

## 8.47.1 Constructor & Destructor Documentation

### 8.47.1.1 OutOfBounds()

```
ArrayTemplateExceptions::OutOfBounds::OutOfBounds (
 const std::string & message = "ArrayTemplate error : out-of-bounds.") [inline]
```

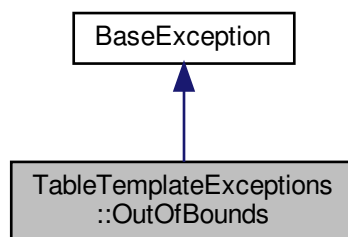
The documentation for this class was generated from the following file:

- GMATsrc/util/[ArrayTemplate.hpp](#)

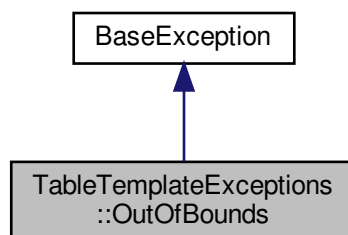
## 8.48 TableTemplateExceptions::OutOfBounds Class Reference

```
#include <TableTemplate.hpp>
```

Inheritance diagram for TableTemplateExceptions::OutOfBounds:



Collaboration diagram for TableTemplateExceptions::OutOfBounds:



## Public Member Functions

- [OutOfBounds](#) (const std::string &message="TableTemplate error : index out-of-bounds.\n")

## Additional Inherited Members

### 8.48.1 Constructor & Destructor Documentation

#### 8.48.1.1 OutOfBounds()

```
TableTemplateExceptions::OutOfBounds::OutOfBounds (
 const std::string & message = "TableTemplate error : index out-of-bounds.\n")
[inline]
```

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

- GMATsrc/util/[TableTemplate.hpp](#)

## 8.49 Gmat::PluginResource Struct Reference

```
#include <gmatdefs.hpp>
```

### Public Member Functions

- [PluginResource](#) ()

### Public Attributes

- `std::string` [nodeName](#)
- `std::string` [parentNodeName](#)
- [ObjectType](#) `type`
- `std::string` [subtype](#)
- `std::string` [toolkit](#)
- `std::string` [widgetType](#)
- [Integer](#) `trigger`
- [Integer](#) `firstId`
- [Integer](#) `lastId`
- `GmatEventHandler *` [handler](#)

### 8.49.1 Constructor & Destructor Documentation

#### 8.49.1.1 PluginResource()

```
Gmat::PluginResource::PluginResource () [inline]
```

### 8.49.2 Member Data Documentation

#### 8.49.2.1 firstId

```
Integer Gmat::PluginResource::firstId
```

#### 8.49.2.2 handler

```
GmatEventHandler* Gmat::PluginResource::handler
```



#### 8.49.2.3 lastId

`Integer` Gmat::PluginResource::lastId

#### 8.49.2.4 nodeName

`std::string` Gmat::PluginResource::nodeName

#### 8.49.2.5 parentNodeName

`std::string` Gmat::PluginResource::parentNodeName

#### 8.49.2.6 subtype

`std::string` Gmat::PluginResource::subtype

#### 8.49.2.7 toolkit

`std::string` Gmat::PluginResource::toolkit

#### 8.49.2.8 trigger

`Integer` Gmat::PluginResource::trigger

#### 8.49.2.9 type

`ObjectType` Gmat::PluginResource::type

#### 8.49.2.10 widgetType

```
std::string Gmat::PluginResource::widgetType
```

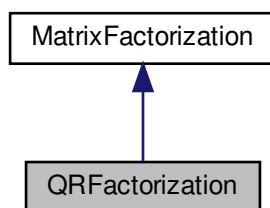
The documentation for this struct was generated from the following file:

- GMATsrc/include/gmatdefs.hpp

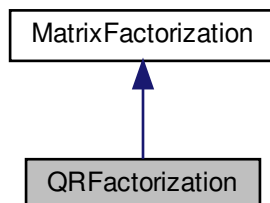
## 8.50 QRFactorization Class Reference

```
#include <QRFactorization.hpp>
```

Inheritance diagram for QRFactorization:



Collaboration diagram for QRFactorization:



## Public Member Functions

- [QRFactorization](#) (bool pivotOption=true)
- [QRFactorization](#) (const [QRFactorization](#) &qrfactorization)
- [~QRFactorization](#) ()
- [QRFactorization](#) & [operator=](#) (const [QRFactorization](#) &qrfactorization)
- void [Factor](#) (const [Rmatrix](#) A, [Rmatrix](#) &R, [Rmatrix](#) &Q)
- void [RemoveFromQR](#) ([Rmatrix](#) R, [Rmatrix](#) Q, std::string dimensionToRemove, [Integer](#) locationToRemove, [Rmatrix](#) &R1, [Rmatrix](#) &Q1)
- void [AddToQR](#) ([Rmatrix](#) R, [Rmatrix](#) Q, std::string dimensionToInsert, [Integer](#) locationToInsert, [Rvector](#) new↵ Elements, [Rmatrix](#) &R1, [Rmatrix](#) &Q1)
- void [Invert](#) ([Rmatrix](#) &inputMatrix)
- [Real Determinant](#) ([Rmatrix](#) A)
- [Rmatrix](#) [GetParameterMatrix](#) ()

## Additional Inherited Members

### 8.50.1 Detailed Description

Declares QR factorization methods. This class strictly uses std::vector operations to create and factor the matrix.

### 8.50.2 Constructor & Destructor Documentation

#### 8.50.2.1 QRFactorization() [1/2]

```
QRFactorization::QRFactorization (
 bool pivotOption = true)
```

Defines QR factorization methods. Constructor

#### 8.50.2.2 QRFactorization() [2/2]

```
QRFactorization::QRFactorization (
 const QRFactorization & qrfactorizationgmat)
```

Copy Constructor

#### 8.50.2.3 ~QRFactorization()

```
QRFactorization::~~QRFactorization ()
```

Class destructor

### 8.50.3 Member Function Documentation

#### 8.50.3.1 AddToQR()

```
void QRFactorization::AddToQR (
 Rmatrix R,
 Rmatrix Q,
 std::string dimensionToInsert,
 Integer locationToInsert,
 Rvector newElements,
 Rmatrix & R1,
 Rmatrix & Q1)
```

Method used to update the QR factorization of a matrix by adding a row or column from the original matrix. Algorithms are based on section 12.5 of Matrix Computations (3rd ed.) by Gene H. Golub and Charles F. Van Loan.

##### Parameters

|                          |                                                                                                        |
|--------------------------|--------------------------------------------------------------------------------------------------------|
| <i>R</i>                 | Original upper triangular matrix from initial QR factorization                                         |
| <i>Q</i>                 | Original orthogonal matrix from initial QR factorization                                               |
| <i>dimensionToInsert</i> | String entry determining whether a row (entered as "row") or column (entered as "col") should be added |
| <i>locationToInsert</i>  | Index of which row/column from the original factored matrix should be added                            |
| <i>&amp;R1</i>           | New upper triangular matrix after removing row/column from A                                           |
| <i>&amp;Q1</i>           | New orthogonal matrix after removing row/column from A                                                 |

#### 8.50.3.2 Determinant()

```
Real QRFactorization::Determinant (
 Rmatrix A)
```

Method that determines the determinant of a square matrix

##### Parameters

|          |                                                        |
|----------|--------------------------------------------------------|
| <i>A</i> | The square matrix a determinant will be calculated for |
|----------|--------------------------------------------------------|

##### Returns

det The determinant of the matrix

## 8.50.3.3 Factor()

```
void QRFactorization::Factor (
 const Rmatrix A,
 Rmatrix & R,
 Rmatrix & Q) [virtual]
```

Method used to factor the matrix with QR factorization ( $A = QR$ ) based off the third option in algorithm 5.2.2 from Gene H. Golub and Chales F. Van Loan with modifications added for column pivoting similar to what is used in algorithm 5.4.1.

## Parameters

|       |                                                                              |
|-------|------------------------------------------------------------------------------|
| $A$   | The matrix that will be factored                                             |
| $\&R$ | The upper triangular matrix, same dimensions as A                            |
| $\&Q$ | The orthogonal matrix, has dimensions mxm where m is the number of rows in A |

Implements [MatrixFactorization](#).

## 8.50.3.4 GetParameterMatrix()

```
Rmatrix QRFactorization::GetParameterMatrix ()
```

Method that returns the parameter matrix

## 8.50.3.5 Invert()

```
void QRFactorization::Invert (
 Rmatrix & inputMatrix) [virtual]
```

Method used to invert the matrix with QR factorization ( $\text{inv}(A) = \text{inv}(Q) * \text{inv}(R) = \text{inv}(R) * \text{transpose}(Q)$ )

## Parameters

|       |                                                                              |
|-------|------------------------------------------------------------------------------|
| $A$   | The matrix that will be factored                                             |
| $\&R$ | The upper triangular matrix, same dimensions as A                            |
| $\&Q$ | The orthogonal matrix, has dimensions mxm where m is the number of rows in A |

Implements [MatrixFactorization](#).

## 8.50.3.6 operator=()

```
QRFactorization & QRFactorization::operator= (
 const QRFactorization & qrfactorization)
```

Assignment operator

### 8.50.3.7 RemoveFromQR()

```
void QRFactorization::RemoveFromQR (
 Rmatrix R,
 Rmatrix Q,
 std::string dimensionToRemove,
 Integer locationToRemove,
 Rmatrix & R1,
 Rmatrix & Q1)
```

Method used to update the QR factorization of a matrix by removing a row or column from the original matrix. Algorithms are based on section 12.5 of Matrix Computations (3rd ed.) by Gene H. Golub and Charles F. Van Loan.

#### Parameters

|                          |                                                                                                          |
|--------------------------|----------------------------------------------------------------------------------------------------------|
| <i>R</i>                 | Original upper triangular matrix from initial QR factorization                                           |
| <i>Q</i>                 | Original orthogonal matrix from initial QR factorization                                                 |
| <i>dimensionToRemove</i> | String entry determining whether a row (entered as "row") or column (entered as "col") should be removed |
| <i>locationToRemove</i>  | Index of which row/column from the original factored matrix should be removed                            |
| <i>&amp;R1</i>           | New upper triangular matrix after removing row/column from A                                             |
| <i>&amp;Q1</i>           | New orthogonal matrix after removing row/column from A                                                   |

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

- [GMATsrc/util/matrixoperations/QRFactorization.hpp](#)
- [GMATsrc/util/matrixoperations/QRFactorization.cpp](#)

## 8.51 GmatRealUtil::RaCodec Struct Reference

```
#include <Linear.hpp>
```

### Public Attributes

- [Real radiusD](#)
- [Real rightAscensionD](#)
- [Real coDeclinationD](#)

### 8.51.1 Member Data Documentation

#### 8.51.1.1 coDeclinationD

```
Real GmatRealUtil::RaCodec::coDeclinationD
```

### 8.51.1.2 radiusD

`Real GmatRealUtil::RaCodec::radiusD`

### 8.51.1.3 rightAscensionD

`Real GmatRealUtil::RaCodec::rightAscensionD`

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

- GMATsrc/util/[Linear.hpp](#)

## 8.52 GmatRealUtil::RaDec Struct Reference

```
#include <Linear.hpp>
```

### Public Attributes

- [Real radiusD](#)
- [Real rightAscensionD](#)
- [Real declinationD](#)

### 8.52.1 Member Data Documentation

#### 8.52.1.1 declinationD

`Real GmatRealUtil::RaDec::declinationD`

#### 8.52.1.2 radiusD

`Real GmatRealUtil::RaDec::radiusD`

### 8.52.1.3 rightAscensionD

`Real GmatRealUtil::RaDec::rightAscensionD`

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

- GMATsrc/util/[Linear.hpp](#)

## 8.53 RandomNumber Class Reference

```
#include <RandomNumber.hpp>
```

### Public Member Functions

- [~RandomNumber](#) ()
- void [SetSeed](#) (unsigned int s)
- void [SetClockSeed](#) ()
- [Real Gaussian](#) ()
- [Real Gaussian](#) (const [Real](#) mean, const [Real](#) stdev)
- void [GaussianArray](#) ([Real](#) \*myArray, const [Integer](#) size)
- void [GaussianArray](#) ([Real](#) \*myArray, const [Integer](#) size, const [Real](#) mean, const [Real](#) stdev)
- [Real Uniform](#) ()
- [Real Uniform](#) (const [Real](#) a, const [Real](#) b)
- void [UniformArray](#) ([Real](#) \*myArray, const [Integer](#) size)
- void [UniformArray](#) ([Real](#) \*myArray, const [Integer](#) size, const [Real](#) a, const [Real](#) b)

### Static Public Member Functions

- static [RandomNumber](#) \* [Instance](#) ()

### 8.53.1 Detailed Description

This class provides random number

### 8.53.2 Constructor & Destructor Documentation

#### 8.53.2.1 ~RandomNumber()

```
RandomNumber::~~RandomNumber ()
```

Class destructor.



### 8.53.3 Member Function Documentation

#### 8.53.3.1 Gaussian() [1/2]

```
Real RandomNumber::Gaussian ()
```

Returns a normally distributed Gaussian random deviate (zero mean, unit var)

#### 8.53.3.2 Gaussian() [2/2]

```
Real RandomNumber::Gaussian (
 const Real mean,
 const Real stdev)
```

Returns a normally distributed Gaussian random deviate.

##### Parameters

|         |                                             |
|---------|---------------------------------------------|
| <mean>  | Mean of Gaussian distribution               |
| <stdev> | Standard deviation of Gaussian distribution |

##### Returns

The random deviate.

#### 8.53.3.3 GaussianArray() [1/2]

```
void RandomNumber::GaussianArray (
 Real * myArray,
 const Integer size)
```

Returns a normally distributed Gaussian random deviate (zero mean, unit var)

##### Parameters

|           |                                                       |
|-----------|-------------------------------------------------------|
| <myArray> | Pointer to array where random deviates will be stored |
| <size>    | size of the array of deviates                         |

#### 8.53.3.4 GaussianArray() [2/2]

```
void RandomNumber::GaussianArray (
 Real * myArray,
```

```

const Integer size,
const Real mean,
const Real stdev)

```

Returns an array of normally distributed Gaussian random deviates with specified mean and variance.

#### Parameters

|           |                                                       |
|-----------|-------------------------------------------------------|
| <myArray> | Pointer to array where random deviates will be stored |
| <size>    | size of the array of deviates                         |
| <mean>    | Mean of Gaussian distribution                         |
| <stdev>   | Standard deviation of Gaussian distribution           |

#### 8.53.3.5 Instance()

```
RandomNumber * RandomNumber::Instance () [static]
```

#### 8.53.3.6 SetClockSeed()

```
void RandomNumber::SetClockSeed ()
```

Set the seed value based upon the current clock time.

#### 8.53.3.7 SetSeed()

```
void RandomNumber::SetSeed (
 unsigned int s)
```

Set the seed for the random number generator using a specified value.

#### Parameters

|     |            |
|-----|------------|
| <s> | input seed |
|-----|------------|

#### 8.53.3.8 Uniform() <sup>[1/2]</sup>

```
Real RandomNumber::Uniform ()
```

Returns an uniformly distributed random deviate in the range [0,1) The range includes 0.0 but excludes 1.0;

#### Returns

The random deviate.

**8.53.3.9 Uniform()** [2/2]

```
Real RandomNumber::Uniform (
 const Real a,
 const Real b)
```

Returns an uniformly distributed random deviate in the range [a,b) The mean of this distribution is (a+b)/2. The variance of this distribution is (b-a)^2/12.

**Parameters**

|     |                    |
|-----|--------------------|
| <a> | Distribution start |
| <b> | Distribution end   |

**Returns**

The random deviate.

**8.53.3.10 UniformArray()** [1/2]

```
void RandomNumber::UniformArray (
 Real * myArray,
 const Integer size)
```

Returns a uniformly distributed random deviate in the range [0,1) The range includes 0.0 but excludes 1.0;

**Parameters**

|           |                                                       |
|-----------|-------------------------------------------------------|
| <myArray> | Pointer to array where random deviates will be stored |
| <size>    | size of the array of deviates                         |

**8.53.3.11 UniformArray()** [2/2]

```
void RandomNumber::UniformArray (
 Real * myArray,
 const Integer size,
 const Real a,
 const Real b)
```

Returns an uniformly distributed random deviate in the range [a,b) The mean of this distribution is (a+b)/2. The variance of this distribution is (b-a)^2/12.

**Parameters**

|           |                                                       |
|-----------|-------------------------------------------------------|
| <myArray> | Pointer to array where random deviates will be stored |
| <size>    | size of the array of deviates                         |
| <a>       | Distribution start                                    |
| <b>       | Distribution end                                      |

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

- GMATsrc/util/[RandomNumber.hpp](#)
- GMATsrc/util/[RandomNumber.cpp](#)

## 8.54 RealUtilitiesExceptions Struct Reference

```
#include <RealUtilities.hpp>
```

### Classes

- class [ArgumentError](#)
- class [IllegalTime](#)

### 8.54.1 Detailed Description

This file provides measurement conversion constantrns and Math Utilities that are not provided in the C++ Library or provides call-throughs to the routines of the C++ (C) math.h

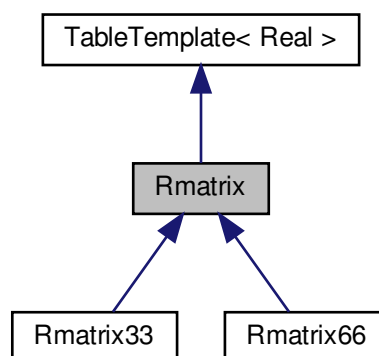
The documentation for this struct was generated from the following file:

- GMATsrc/util/[RealUtilities.hpp](#)

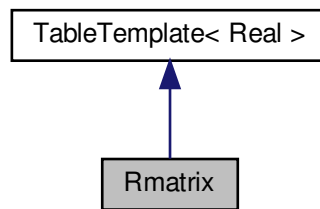
## 8.55 Rmatrix Class Reference

```
#include <Rmatrix.hpp>
```

Inheritance diagram for Rmatrix:



Collaboration diagram for Rmatrix:



## Classes

- class [DivideByZero](#)
- class [IsSingular](#)
- class [NotSquare](#)

## Public Member Functions

- [Rmatrix](#) ()
- [Rmatrix](#) (int r, int c)
- [Rmatrix](#) (int r, int c, [Real](#) a1,...)
- [Rmatrix](#) (const [Rmatrix](#) &m)
- virtual [~Rmatrix](#) ()
- virtual bool [IsOrthogonal](#) ([Real](#) accuracyRequired=[GmatRealConstants::REAL\\_EPSILON](#)) const
- virtual bool [IsOrthonormal](#) ([Real](#) accuracyRequired=[GmatRealConstants::REAL\\_EPSILON](#)) const
- const [Rmatrix](#) & [operator=](#) (const [Rmatrix](#) &m)
- bool [operator==](#) (const [Rmatrix](#) &m) const
- bool [operator!=](#) (const [Rmatrix](#) &m) const
- [Rmatrix](#) [operator+](#) (const [Rmatrix](#) &RHSRmatrix) const
- const [Rmatrix](#) & [operator+=](#) (const [Rmatrix](#) &RHSRmatrix)
- [Rmatrix](#) [operator-](#) (const [Rmatrix](#) &RHSRmatrix) const
- const [Rmatrix](#) & [operator-=](#) (const [Rmatrix](#) &RHSRmatrix)
- [Rmatrix](#) [operator\\*](#) (const [Rmatrix](#) &RHSRmatrix) const
- const [Rmatrix](#) & [operator\\*=](#) (const [Rmatrix](#) &RHSRmatrix)
- [Rmatrix](#) [operator/](#) (const [Rmatrix](#) &RHSRmatrix) const
- const [Rmatrix](#) & [operator/=](#) (const [Rmatrix](#) &RHSRmatrix)
- [Rmatrix](#) [ElementWiseMultiply](#) (const [Rmatrix](#) &m)
- [Rmatrix](#) [ElementWiseDivide](#) (const [Rmatrix](#) &m)
- [Rmatrix](#) [operator+](#) ([Real](#) scalar) const
- const [Rmatrix](#) & [operator+=](#) ([Real](#) scalar)
- [Rmatrix](#) [operator-](#) ([Real](#) scalar) const
- const [Rmatrix](#) & [operator-=](#) ([Real](#) scalar)
- [Rmatrix](#) [operator\\*](#) ([Real](#) scalar) const
- const [Rmatrix](#) & [operator\\*=](#) ([Real](#) scalar)
- [Rmatrix](#) [operator/](#) ([Real](#) scalar) const
- const [Rmatrix](#) & [operator/=](#) ([Real](#) scalar)

- [Rmatrix operator-](#) () const
- [Rvector operator\\*](#) (const [Rvector](#) &v) const
- virtual [Real Trace](#) () const
- virtual [Real Determinant](#) () const
- virtual [Real Cofactor](#) (int r, int c) const
- [Rmatrix Transpose](#) () const
- [Rmatrix Inverse](#) () const
- [Rmatrix Inverse](#) ([Real](#) zeroValue) const
- virtual [Rmatrix Pseudoinverse](#) ([Real](#) zeroValue=1e-12) const
- [Rmatrix Symmetric](#) () const
- [Rmatrix AntiSymmetric](#) () const
- [Rvector GetRow](#) (int r) const
- [Rvector GetColumn](#) (int c) const
- [Rvector GetRowOrColumn](#) () const
- void [MakeOneRowMatrix](#) (const [Rvector](#) &vec)
- void [MakeOneColumnMatrix](#) (const [Rvector](#) &vec)
- const [StringArray](#) & [GetStringVals](#) ([Integer](#) p=[GmatGlobal::DATA\\_PRECISION](#), [Integer](#) w=[GmatGlobal::DATA\\_WIDTH](#))
- virtual std::string [ToString](#) ([Integer](#) precision, [Integer](#) width=1, bool horizontal=false, const std::string &prefix="", bool appendEol=true) const
- virtual std::string [ToString](#) (bool useCurrentFormat=true, bool scientific=false, bool showPoint=false, [Integer](#) precision=[GmatGlobal::DATA\\_PRECISION](#), [Integer](#) width=[GmatGlobal::DATA\\_WIDTH](#), bool horizontal=true, [Integer](#) spacing=1, const std::string &prefix="", bool appendEol=true) const
- virtual std::string [ToRowString](#) ([Integer](#) row, [Integer](#) precision, [Integer](#) width=1, bool zeroFill=false) const

### Static Public Member Functions

- static [Rmatrix Identity](#) (unsigned int size)
- static [Rmatrix Diagonal](#) (unsigned int size, [Rvector](#) data)

### Protected Attributes

- [StringArray](#) stringVals

### Friends

- class [Rvector](#)
- class [Rvector3](#)
- [GMATUTIL\\_API](#) friend [Rmatrix operator+](#) ([Real](#) scalar, const [Rmatrix](#) &m)
- [GMATUTIL\\_API](#) friend [Rmatrix operator-](#) ([Real](#) scalar, const [Rmatrix](#) &m)
- [GMATUTIL\\_API](#) friend [Rmatrix operator\\*](#) ([Real](#) scalar, const [Rmatrix](#) &m)
- [GMATUTIL\\_API](#) friend [Rmatrix operator/](#) ([Real](#) scalar, const [Rmatrix](#) &m)
- [Rmatrix SkewSymmetric4by4](#) (const [Rvector3](#) &v)
- [Rmatrix TransposeTimesMatrix](#) (const [Rmatrix](#) &m1, const [Rmatrix](#) &m2)
- [Rmatrix MatrixTimesTranspose](#) (const [Rmatrix](#) &m1, const [Rmatrix](#) &m2)
- [Rmatrix TransposeTimesTranspose](#) (const [Rmatrix](#) &m1, const [Rmatrix](#) &m2)
- std::istream & [operator>>](#) (std::istream &input, [Rmatrix](#) &a)
- std::ostream & [operator<<](#) (std::ostream &output, const [Rmatrix](#) &a)

## Additional Inherited Members

### 8.55.1 Constructor & Destructor Documentation

#### 8.55.1.1 Rmatrix() [1/4]

```
Rmatrix::Rmatrix ()
```

Defines Matrix operations.

#### 8.55.1.2 Rmatrix() [2/4]

```
Rmatrix::Rmatrix (
 int r,
 int c)
```

#### 8.55.1.3 Rmatrix() [3/4]

```
Rmatrix::Rmatrix (
 int r,
 int c,
 Real a1,
 ...)
```

#### 8.55.1.4 Rmatrix() [4/4]

```
Rmatrix::Rmatrix (
 const Rmatrix & m)
```

#### 8.55.1.5 ~Rmatrix()

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

### 8.55.2 Member Function Documentation

#### 8.55.2.1 AntiSymmetric()

```
Rmatrix Rmatrix::AntiSymmetric () const
```

#### 8.55.2.2 Cofactor()

```
Real Rmatrix::Cofactor (
 int r,
 int c) const [virtual]
```

#### 8.55.2.3 Determinant()

```
Real Rmatrix::Determinant () const [virtual]
```

Reimplemented in [Rmatrix33](#), and [Rmatrix66](#).

#### 8.55.2.4 Diagonal()

```
Rmatrix Rmatrix::Diagonal (
 unsigned int size,
 Rvector data) [static]
```

#### 8.55.2.5 ElementWiseDivide()

```
Rmatrix Rmatrix::ElementWiseDivide (
 const Rmatrix & m)
```

#### 8.55.2.6 ElementWiseMultiply()

```
Rmatrix Rmatrix::ElementWiseMultiply (
 const Rmatrix & m)
```



#### 8.55.2.7 GetColumn()

```
Rvector Rmatrix::GetColumn (
 int c) const
```

#### 8.55.2.8 GetRow()

```
Rvector Rmatrix::GetRow (
 int r) const
```

#### 8.55.2.9 GetRowOrColumn()

```
Rvector Rmatrix::GetRowOrColumn () const
```

Returns row or column vector if matrix is one dimensional array (1xN or Mx1 matrix)

#### 8.55.2.10 GetStringVals()

```
const StringArray & Rmatrix::GetStringVals (
 Integer p = GmatGlobal::DATA_PRECISION,
 Integer w = GmatGlobal::DATA_WIDTH)
```

#### 8.55.2.11 Identity()

```
Rmatrix Rmatrix::Identity (
 unsigned int size) [static]
```

#### 8.55.2.12 Inverse() [1/2]

```
Rmatrix Rmatrix::Inverse () const
```

#### 8.55.2.13 Inverse() [2/2]

```
Rmatrix Rmatrix::Inverse (
 Real zeroValue) const
```

#### 8.55.2.14 IsOrthogonal()

```
bool Rmatrix::IsOrthogonal (
 Real accuracyRequired = GmatRealConstants::REAL_EPSILON) const [virtual]
```

Reimplemented in [Rmatrix33](#), and [Rmatrix66](#).

#### 8.55.2.15 IsOrthonormal()

```
bool Rmatrix::IsOrthonormal (
 Real accuracyRequired = GmatRealConstants::REAL_EPSILON) const [virtual]
```

Reimplemented in [Rmatrix33](#), and [Rmatrix66](#).

#### 8.55.2.16 MakeOneColumnMatrix()

```
void Rmatrix::MakeOneColumnMatrix (
 const Rvector & vec)
```

#### 8.55.2.17 MakeOneRowMatrix()

```
void Rmatrix::MakeOneRowMatrix (
 const Rvector & vec)
```

#### 8.55.2.18 operator!=(=)

```
bool Rmatrix::operator!= (
 const Rmatrix & m) const
```

#### 8.55.2.19 operator\*() [1/3]

```
Rmatrix Rmatrix::operator* (
 const Rmatrix & RHSRmatrix) const
```

**8.55.2.20 operator\*()** [2/3]

```
Rmatrix Rmatrix::operator* (
 Real scalar) const
```

**8.55.2.21 operator\*()** [3/3]

```
Rvector Rmatrix::operator* (
 const Rvector & v) const
```

**8.55.2.22 operator\*=( )** [1/2]

```
const Rmatrix & Rmatrix::operator*= (
 const Rmatrix & RHSRmatrix)
```

**8.55.2.23 operator\*=( )** [2/2]

```
const Rmatrix & Rmatrix::operator*= (
 Real scalar)
```

**8.55.2.24 operator+( )** [1/2]

```
Rmatrix Rmatrix::operator+ (
 const Rmatrix & RHSRmatrix) const
```

**8.55.2.25 operator+( )** [2/2]

```
Rmatrix Rmatrix::operator+ (
 Real scalar) const
```

**8.55.2.26 operator+=( )** [1/2]

```
const Rmatrix & Rmatrix::operator+= (
 const Rmatrix & RHSRmatrix)
```

**8.55.2.27 operator+=( )** [2/2]

```
const Rmatrix & Rmatrix::operator+= (
 Real scalar)
```

**8.55.2.28 operator-( )** [1/3]

```
Rmatrix Rmatrix::operator- (
 const Rmatrix & RHSRmatrix) const
```

**8.55.2.29 operator-( )** [2/3]

```
Rmatrix Rmatrix::operator- (
 Real scalar) const
```

**8.55.2.30 operator-( )** [3/3]

```
Rmatrix Rmatrix::operator- () const
```

**8.55.2.31 operator-=( )** [1/2]

```
const Rmatrix & Rmatrix::operator-= (
 const Rmatrix & RHSRmatrix)
```

**8.55.2.32 operator-=( )** [2/2]

```
const Rmatrix & Rmatrix::operator-= (
 Real scalar)
```

**8.55.2.33 operator/( )** [1/2]

```
Rmatrix Rmatrix::operator/ (
 const Rmatrix & RHSRmatrix) const
```

**8.55.2.34 operator/()** [2/2]

```
Rmatrix Rmatrix::operator/ (
 Real scalar) const
```

**8.55.2.35 operator/=( )** [1/2]

```
const Rmatrix & Rmatrix::operator/= (
 const Rmatrix & RHSRmatrix)
```

**8.55.2.36 operator/=( )** [2/2]

```
const Rmatrix & Rmatrix::operator/= (
 Real scalar)
```

**8.55.2.37 operator=( )**

```
const Rmatrix & Rmatrix::operator= (
 const Rmatrix & m)
```

**8.55.2.38 operator==( )**

```
bool Rmatrix::operator== (
 const Rmatrix & m) const
```

**8.55.2.39 Pseudoinverse()**

```
Rmatrix Rmatrix::Pseudoinverse (
 Real zeroValue = 1e-12) const [virtual]
```

**8.55.2.40 Symmetric()**

```
Rmatrix Rmatrix::Symmetric () const
```

**8.55.2.41 ToRowString()**

```
std::string Rmatrix::ToRowString (
 Integer row,
 Integer precision,
 Integer width = 1,
 bool zeroFill = false) const [virtual]
```

**8.55.2.42 ToString()** [1/2]

```
std::string Rmatrix::ToString (
 Integer precision,
 Integer width = 1,
 bool horizontal = false,
 const std::string & prefix = "",
 bool appendEol = true) const [virtual]
```

**8.55.2.43 ToString()** [2/2]

```
std::string Rmatrix::ToString (
 bool useCurrentFormat = true,
 bool scientific = false,
 bool showPoint = false,
 Integer precision = GmatGlobal::DATA_PRECISION,
 Integer width = GmatGlobal::DATA_WIDTH,
 bool horizontal = true,
 Integer spacing = 1,
 const std::string & prefix = "",
 bool appendEol = true) const [virtual]
```

**8.55.2.44 Trace()**

```
Real Rmatrix::Trace () const [virtual]
```

Reimplemented in [Rmatrix33](#), and [Rmatrix66](#).

**8.55.2.45 Transpose()**

```
Rmatrix Rmatrix::Transpose () const
```

### 8.55.3 Friends And Related Function Documentation

#### 8.55.3.1 MatrixTimesTranspose

```
Rmatrix MatrixTimesTranspose (
 const Rmatrix & m1,
 const Rmatrix & m2) [friend]
```

#### 8.55.3.2 operator\*

```
GMATUTIL_API friend Rmatrix operator* (
 Real scalar,
 const Rmatrix & m) [friend]
```

#### 8.55.3.3 operator+

```
GMATUTIL_API friend Rmatrix operator+ (
 Real scalar,
 const Rmatrix & m) [friend]
```

#### 8.55.3.4 operator-

```
GMATUTIL_API friend Rmatrix operator- (
 Real scalar,
 const Rmatrix & m) [friend]
```

#### 8.55.3.5 operator/

```
GMATUTIL_API friend Rmatrix operator/ (
 Real scalar,
 const Rmatrix & m) [friend]
```

#### 8.55.3.6 operator<<

```
std::ostream& operator<< (
 std::ostream & output,
 const Rmatrix & a) [friend]
```

#### 8.55.3.7 operator>>

```
std::istream& operator>> (
 std::istream & input,
 Rmatrix & a) [friend]
```

#### 8.55.3.8 Rvector

```
friend class Rvector [friend]
```

#### 8.55.3.9 Rvector3

```
friend class Rvector3 [friend]
```

#### 8.55.3.10 SkewSymmetric4by4

```
Rmatrix SkewSymmetric4by4 (
 const Rvector3 & v) [friend]
```

#### 8.55.3.11 TransposeTimesMatrix

```
Rmatrix TransposeTimesMatrix (
 const Rmatrix & m1,
 const Rmatrix & m2) [friend]
```

#### 8.55.3.12 TransposeTimesTranspose

```
Rmatrix TransposeTimesTranspose (
 const Rmatrix & m1,
 const Rmatrix & m2) [friend]
```



### 8.55.4 Member Data Documentation

#### 8.55.4.1 stringVals

```
StringArray Rmatrix::stringVals [protected]
```

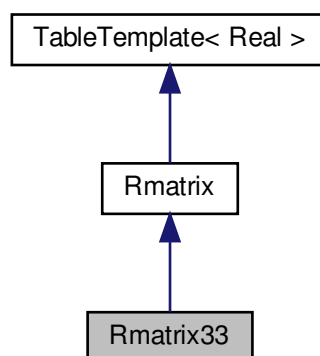
The documentation for this class was generated from the following files:

- GMATsrc/util/[Rmatrix.hpp](#)
- GMATsrc/util/[Rmatrix.cpp](#)

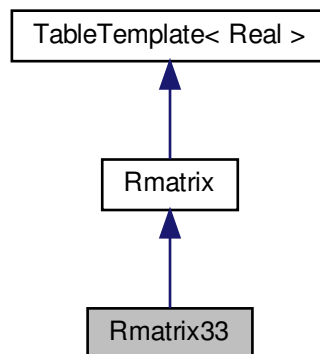
## 8.56 Rmatrix33 Class Reference

```
#include <Rmatrix33.hpp>
```

Inheritance diagram for Rmatrix33:



Collaboration diagram for Rmatrix33:



## Public Member Functions

- [Rmatrix33](#) (bool IsIdentityMatrix=true)
- [Rmatrix33](#) (Real a00, Real a01, Real a02, Real a10, Real a11, Real a12, Real a20, Real a21, Real a22)
- [Rmatrix33](#) (const [Rmatrix33](#) &m)
- [Rmatrix33](#) (const [Rmatrix](#) &m)
- virtual [~Rmatrix33](#) ()
- const [Rmatrix33](#) & [operator=](#) (const [Rmatrix33](#) &m)
- bool [operator==](#) (const [Rmatrix33](#) &m) const
- bool [operator!=](#) (const [Rmatrix33](#) &m) const
- void [Set](#) (Real a00, Real a01, Real a02, Real a10, Real a11, Real a12, Real a20, Real a21, Real a22)
- bool [IsOrthogonal](#) (Real accuracyRequired=[GmatRealConstants::REAL\\_EPSILON](#)) const
- bool [IsOrthonormal](#) (Real accuracyRequired=[GmatRealConstants::REAL\\_EPSILON](#)) const
- [Rmatrix33 operator+](#) (const [Rmatrix33](#) &m) const
- const [Rmatrix33](#) & [operator+=](#) (const [Rmatrix33](#) &m)
- [Rmatrix33 operator-](#) (const [Rmatrix33](#) &m) const
- const [Rmatrix33](#) & [operator-=](#) (const [Rmatrix33](#) &m)
- [Rmatrix33 operator\\*](#) (const [Rmatrix33](#) &m) const
- const [Rmatrix33](#) & [operator\\*=](#) (const [Rmatrix33](#) &m)
- [Rmatrix33 operator/](#) (const [Rmatrix33](#) &m) const
- const [Rmatrix33](#) & [operator/=](#) (const [Rmatrix33](#) &m)
- [Rmatrix33 operator\\*](#) (Real scalar) const
- const [Rmatrix33](#) & [operator\\*=](#) (Real scalar)
- [Rmatrix33 operator/](#) (Real scalar) const
- const [Rmatrix33](#) & [operator/=](#) (Real scalar)
- [Rmatrix33 operator-](#) () const
- [Rvector3 operator\\*](#) (const [Rvector3](#) &v) const
- Real [Trace](#) () const
- Real [Determinant](#) () const
- virtual [Rmatrix33 Transpose](#) () const
- virtual [Rmatrix33 Inverse](#) () const
- virtual [Rmatrix33 Symmetric](#) () const
- [Rmatrix33 AntiSymmetric](#) () const
- const std::string \* [GetDataDescriptions](#) () const

## Friends

- class [Rvector3](#)
- [Rmatrix33 operator\\*](#) ([Real](#) scalar, const [Rmatrix33](#) &m)
- [Rmatrix33 SkewSymmetric](#) (const [Rvector3](#) &v)
- [Rmatrix33 TransposeTimesMatrix](#) (const [Rmatrix33](#) &m1, const [Rmatrix33](#) &m2)
- [Rmatrix33 MatrixTimesTranspose](#) (const [Rmatrix33](#) &m1, const [Rmatrix33](#) &m2)
- [Rmatrix33 TransposeTimesTranspose](#) (const [Rmatrix33](#) &m1, const [Rmatrix33](#) &m2)

## Additional Inherited Members

### 8.56.1 Constructor & Destructor Documentation

#### 8.56.1.1 [Rmatrix33\(\)](#) [1/4]

```
Rmatrix33::Rmatrix33 (
 bool IsIdentityMatrix = true)
```

#### 8.56.1.2 [Rmatrix33\(\)](#) [2/4]

```
Rmatrix33::Rmatrix33 (
 Real a00,
 Real a01,
 Real a02,
 Real a10,
 Real a11,
 Real a12,
 Real a20,
 Real a21,
 Real a22)
```

#### 8.56.1.3 [Rmatrix33\(\)](#) [3/4]

```
Rmatrix33::Rmatrix33 (
 const Rmatrix33 & m)
```

#### 8.56.1.4 [Rmatrix33\(\)](#) [4/4]

```
Rmatrix33::Rmatrix33 (
 const Rmatrix & m)
```

#### 8.56.1.5 ~Rmatrix33()

```
Rmatrix33::~~Rmatrix33 () [virtual]
```

### 8.56.2 Member Function Documentation

#### 8.56.2.1 AntiSymmetric()

```
Rmatrix33 Rmatrix33::AntiSymmetric () const
```

#### 8.56.2.2 Determinant()

```
Real Rmatrix33::Determinant () const [virtual]
```

Reimplemented from [Rmatrix](#).

#### 8.56.2.3 GetDataDescriptions()

```
const std::string * Rmatrix33::GetDataDescriptions () const
```

#### 8.56.2.4 Inverse()

```
Rmatrix33 Rmatrix33::Inverse () const [virtual]
```

#### 8.56.2.5 IsOrthogonal()

```
bool Rmatrix33::IsOrthogonal (
 Real accuracyRequired = GmatRealConstants::REAL_EPSILON) const [virtual]
```

Reimplemented from [Rmatrix](#).

### 8.56.2.6 IsOrthonormal()

```
bool Rmatrix33::IsOrthonormal (
 Real accuracyRequired = GmatRealConstants::REAL_EPSILON) const [virtual]
```

Reimplemented from [Rmatrix](#).

### 8.56.2.7 operator!=(())

```
bool Rmatrix33::operator!= (
 const Rmatrix33 & m) const
```

### 8.56.2.8 operator\*() [1/3]

```
Rmatrix33 Rmatrix33::operator* (
 const Rmatrix33 & m) const
```

### 8.56.2.9 operator\*() [2/3]

```
Rmatrix33 Rmatrix33::operator* (
 Real scalar) const
```

### 8.56.2.10 operator\*() [3/3]

```
Rvector3 Rmatrix33::operator* (
 const Rvector3 & v) const
```

### 8.56.2.11 operator\*=(()) [1/2]

```
const Rmatrix33 & Rmatrix33::operator*= (
 const Rmatrix33 & m)
```

**8.56.2.12 operator\*=( )** [2/2]

```
const Rmatrix33 & Rmatrix33::operator*= (
 Real scalar)
```

**8.56.2.13 operator+( )**

```
Rmatrix33 Rmatrix33::operator+ (
 const Rmatrix33 & m) const
```

**8.56.2.14 operator+=( )**

```
const Rmatrix33 & Rmatrix33::operator+= (
 const Rmatrix33 & m)
```

**8.56.2.15 operator-( )** [1/2]

```
Rmatrix33 Rmatrix33::operator- (
 const Rmatrix33 & m) const
```

**8.56.2.16 operator-( )** [2/2]

```
Rmatrix33 Rmatrix33::operator- () const
```

**8.56.2.17 operator-=( )**

```
const Rmatrix33 & Rmatrix33::operator-= (
 const Rmatrix33 & m)
```

**8.56.2.18 operator/( )** [1/2]

```
Rmatrix33 Rmatrix33::operator/ (
 const Rmatrix33 & m) const
```

**8.56.2.19 operator/()** [2/2]

```
Rmatrix33 Rmatrix33::operator/ (
 Real scalar) const
```

**8.56.2.20 operator/=( )** [1/2]

```
const Rmatrix33 & Rmatrix33::operator/= (
 const Rmatrix33 & m)
```

**8.56.2.21 operator/=( )** [2/2]

```
const Rmatrix33 & Rmatrix33::operator/= (
 Real scalar)
```

**8.56.2.22 operator=( )**

```
const Rmatrix33 & Rmatrix33::operator= (
 const Rmatrix33 & m)
```

**8.56.2.23 operator==( )**

```
bool Rmatrix33::operator== (
 const Rmatrix33 & m) const
```

**8.56.2.24 Set()**

```
void Rmatrix33::Set (
 Real a00,
 Real a01,
 Real a02,
 Real a10,
 Real a11,
 Real a12,
 Real a20,
 Real a21,
 Real a22)
```

#### 8.56.2.25 Symmetric()

```
Rmatrix33 Rmatrix33::Symmetric () const [virtual]
```

#### 8.56.2.26 Trace()

```
Real Rmatrix33::Trace () const [virtual]
```

Reimplemented from [Rmatrix](#).

#### 8.56.2.27 Transpose()

```
Rmatrix33 Rmatrix33::Transpose () const [virtual]
```

### 8.56.3 Friends And Related Function Documentation

#### 8.56.3.1 MatrixTimesTranspose

```
Rmatrix33 MatrixTimesTranspose (
 const Rmatrix33 & m1,
 const Rmatrix33 & m2) [friend]
```

#### 8.56.3.2 operator\*

```
Rmatrix33 operator* (
 Real scalar,
 const Rmatrix33 & m) [friend]
```

#### 8.56.3.3 Rvector3

```
friend class Rvector3 [friend]
```



#### 8.56.3.4 SkewSymmetric

```
Rmatrix33 SkewSymmetric (
 const Rvector3 & v) [friend]
```

#### 8.56.3.5 TransposeTimesMatrix

```
Rmatrix33 TransposeTimesMatrix (
 const Rmatrix33 & m1,
 const Rmatrix33 & m2) [friend]
```

#### 8.56.3.6 TransposeTimesTranspose

```
Rmatrix33 TransposeTimesTranspose (
 const Rmatrix33 & m1,
 const Rmatrix33 & m2) [friend]
```

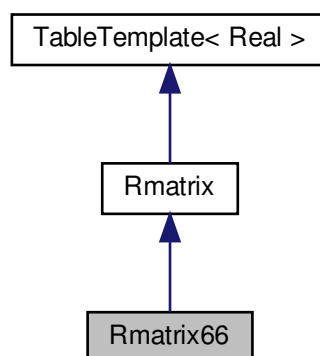
The documentation for this class was generated from the following files:

- GMATsrc/util/[Rmatrix33.hpp](#)
- GMATsrc/util/[Rmatrix33.cpp](#)

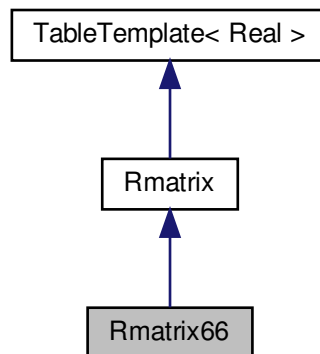
## 8.57 Rmatrix66 Class Reference

```
#include <Rmatrix66.hpp>
```

Inheritance diagram for Rmatrix66:



Collaboration diagram for Rmatrix66:



## Public Member Functions

- [Rmatrix66](#) (bool IsIdentityMatrix=true)
- [Rmatrix66](#) (int nArgs, [Real](#) a1,...)
- [Rmatrix66](#) (const [Rmatrix66](#) &m)
- [Rmatrix66](#) (const [Rmatrix](#) &m)
- virtual [~Rmatrix66](#) ()
- const [Rmatrix66](#) & [operator=](#) (const [Rmatrix66](#) &m)
- bool [operator==](#) (const [Rmatrix66](#) &m) const
- bool [operator!=](#) (const [Rmatrix66](#) &m) const
- void [Set](#) (int nArgs, [Real](#) a1,...)
- void [SetUndefined](#) ()
- [Rmatrix33](#) [UpperLeft](#) ()
- [Rmatrix33](#) [UpperRight](#) ()
- [Rmatrix33](#) [LowerLeft](#) ()
- [Rmatrix33](#) [LowerRight](#) ()
- bool [IsOrthogonal](#) ([Real](#) accuracyRequired=[GmatRealConstants::REAL\\_EPSILON](#)) const
- bool [IsOrthonormal](#) ([Real](#) accuracyRequired=[GmatRealConstants::REAL\\_EPSILON](#)) const
- [Rmatrix66](#) [operator+](#) (const [Rmatrix66](#) &m) const
- const [Rmatrix66](#) & [operator+=](#) (const [Rmatrix66](#) &m)
- [Rmatrix66](#) [operator-](#) (const [Rmatrix66](#) &m) const
- const [Rmatrix66](#) & [operator-=](#) (const [Rmatrix66](#) &m)
- [Rmatrix66](#) [operator\\*](#) (const [Rmatrix66](#) &m) const
- const [Rmatrix66](#) & [operator\\*=](#) (const [Rmatrix66](#) &m)
- [Rmatrix66](#) [operator/](#) (const [Rmatrix66](#) &m) const
- const [Rmatrix66](#) & [operator/=](#) (const [Rmatrix66](#) &m)
- [Rmatrix66](#) [operator\\*](#) ([Real](#) scalar) const
- const [Rmatrix66](#) & [operator\\*=](#) ([Real](#) scalar)
- [Rmatrix66](#) [operator/](#) ([Real](#) scalar) const
- const [Rmatrix66](#) & [operator/=](#) ([Real](#) scalar)
- [Rmatrix66](#) [operator-](#) () const
- [Rvector6](#) [operator\\*](#) (const [Rvector6](#) &v) const
- [Real](#) [Trace](#) () const

- [Real Determinant](#) () const
- virtual [Rmatrix66 Transpose](#) () const
- virtual [Rmatrix66 Inverse](#) () const
- virtual [Rmatrix66 Symmetric](#) () const
- [Rmatrix66 AntiSymmetric](#) () const

## Friends

- class [Rvector6](#)
- [Rmatrix66 operator\\*](#) ([Real](#) scalar, const [Rmatrix66](#) &m)
- [Rmatrix66 SkewSymmetric](#) (const [Rvector6](#) &v)
- [Rmatrix66 TransposeTimesMatrix](#) (const [Rmatrix66](#) &m1, const [Rmatrix66](#) &m2)
- [Rmatrix66 MatrixTimesTranspose](#) (const [Rmatrix66](#) &m1, const [Rmatrix66](#) &m2)
- [Rmatrix66 TransposeTimesTranspose](#) (const [Rmatrix66](#) &m1, const [Rmatrix66](#) &m2)

## Additional Inherited Members

### 8.57.1 Constructor & Destructor Documentation

#### 8.57.1.1 [Rmatrix66\(\)](#) [1/4]

```
Rmatrix66::Rmatrix66 (
 bool isIdentityMatrix = true)
```

Defines linear algebra operations for 6x6 matrices.

#### 8.57.1.2 [Rmatrix66\(\)](#) [2/4]

```
Rmatrix66::Rmatrix66 (
 int nArgs,
 Real a1,
 ...)
```

#### 8.57.1.3 [Rmatrix66\(\)](#) [3/4]

```
Rmatrix66::Rmatrix66 (
 const Rmatrix66 & m)
```

#### 8.57.1.4 Rmatrix66() [4/4]

```
Rmatrix66::Rmatrix66 (
 const Rmatrix & m)
```

#### 8.57.1.5 ~Rmatrix66()

```
Rmatrix66::~~Rmatrix66 () [virtual]
```

### 8.57.2 Member Function Documentation

#### 8.57.2.1 AntiSymmetric()

```
Rmatrix66 Rmatrix66::AntiSymmetric () const
```

#### 8.57.2.2 Determinant()

```
Real Rmatrix66::Determinant () const [virtual]
```

Reimplemented from [Rmatrix](#).

#### 8.57.2.3 Inverse()

```
Rmatrix66 Rmatrix66::Inverse () const [virtual]
```

#### 8.57.2.4 IsOrthogonal()

```
bool Rmatrix66::IsOrthogonal (
 Real accuracyRequired = GmatRealConstants::REAL_EPSILON) const [virtual]
```

Reimplemented from [Rmatrix](#).

#### 8.57.2.5 IsOrthonormal()

```
bool Rmatrix66::IsOrthonormal (
 Real accuracyRequired = GmatRealConstants::REAL_EPSILON) const [virtual]
```

Reimplemented from [Rmatrix](#).

#### 8.57.2.6 LowerLeft()

```
Rmatrix33 Rmatrix66::LowerLeft ()
```

#### 8.57.2.7 LowerRight()

```
Rmatrix33 Rmatrix66::LowerRight ()
```

#### 8.57.2.8 operator!=(())

```
bool Rmatrix66::operator!= (
 const Rmatrix66 & m) const
```

#### 8.57.2.9 operator\*() [1/3]

```
Rmatrix66 Rmatrix66::operator* (
 const Rmatrix66 & m) const
```

#### 8.57.2.10 operator\*() [2/3]

```
Rmatrix66 Rmatrix66::operator* (
 Real scalar) const
```

#### 8.57.2.11 operator\*() [3/3]

```
Rvector6 Rmatrix66::operator* (
 const Rvector6 & v) const
```

**8.57.2.12 operator\*=( )** [1/2]

```
const Rmatrix66 & Rmatrix66::operator*= (
 const Rmatrix66 & m)
```

**8.57.2.13 operator\*=( )** [2/2]

```
const Rmatrix66 & Rmatrix66::operator*= (
 Real scalar)
```

**8.57.2.14 operator+( )**

```
Rmatrix66 Rmatrix66::operator+ (
 const Rmatrix66 & m) const
```

**8.57.2.15 operator+=( )**

```
const Rmatrix66 & Rmatrix66::operator+= (
 const Rmatrix66 & m)
```

**8.57.2.16 operator-( )** [1/2]

```
Rmatrix66 Rmatrix66::operator- (
 const Rmatrix66 & m) const
```

**8.57.2.17 operator-( )** [2/2]

```
Rmatrix66 Rmatrix66::operator- () const
```

**8.57.2.18 operator-=( )**

```
const Rmatrix66 & Rmatrix66::operator-= (
 const Rmatrix66 & m)
```

**8.57.2.19 operator/()** [1/2]

```
Rmatrix66 Rmatrix66::operator/ (
 const Rmatrix66 & m) const
```

**8.57.2.20 operator/()** [2/2]

```
Rmatrix66 Rmatrix66::operator/ (
 Real scalar) const
```

**8.57.2.21 operator/=()** [1/2]

```
const Rmatrix66 & Rmatrix66::operator/= (
 const Rmatrix66 & m)
```

**8.57.2.22 operator/=()** [2/2]

```
const Rmatrix66 & Rmatrix66::operator/= (
 Real scalar)
```

**8.57.2.23 operator=()**

```
const Rmatrix66 & Rmatrix66::operator= (
 const Rmatrix66 & m)
```

**8.57.2.24 operator==()**

```
bool Rmatrix66::operator== (
 const Rmatrix66 & m) const
```

**8.57.2.25 Set()**

```
void Rmatrix66::Set (
 int nArgs,
 Real a1,
 ...)
```

#### 8.57.2.26 SetUndefined()

```
void Rmatrix66::SetUndefined ()
```

#### 8.57.2.27 Symmetric()

```
Rmatrix66 Rmatrix66::Symmetric () const [virtual]
```

#### 8.57.2.28 Trace()

```
Real Rmatrix66::Trace () const [virtual]
```

Reimplemented from [Rmatrix](#).

#### 8.57.2.29 Transpose()

```
Rmatrix66 Rmatrix66::Transpose () const [virtual]
```

#### 8.57.2.30 UpperLeft()

```
Rmatrix33 Rmatrix66::UpperLeft ()
```

#### 8.57.2.31 UpperRight()

```
Rmatrix33 Rmatrix66::UpperRight ()
```

### 8.57.3 Friends And Related Function Documentation

#### 8.57.3.1 MatrixTimesTranspose

```
Rmatrix66 MatrixTimesTranspose (
 const Rmatrix66 & m1,
 const Rmatrix66 & m2) [friend]
```



### 8.57.3.2 operator\*

```
Rmatrix66 operator* (
 Real scalar,
 const Rmatrix66 & m) [friend]
```

### 8.57.3.3 Rvector6

```
friend class Rvector6 [friend]
```

### 8.57.3.4 SkewSymmetric

```
Rmatrix66 SkewSymmetric (
 const Rvector6 & v) [friend]
```

### 8.57.3.5 TransposeTimesMatrix

```
Rmatrix66 TransposeTimesMatrix (
 const Rmatrix66 & m1,
 const Rmatrix66 & m2) [friend]
```

### 8.57.3.6 TransposeTimesTranspose

```
Rmatrix66 TransposeTimesTranspose (
 const Rmatrix66 & m1,
 const Rmatrix66 & m2) [friend]
```

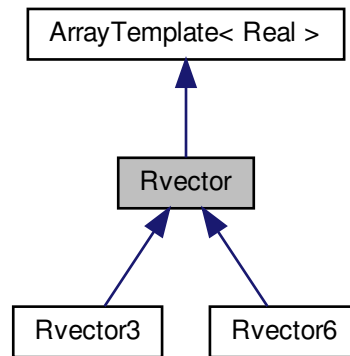
The documentation for this class was generated from the following files:

- GMATsrc/util/[Rmatrix66.hpp](#)
- GMATsrc/util/[Rmatrix66.cpp](#)

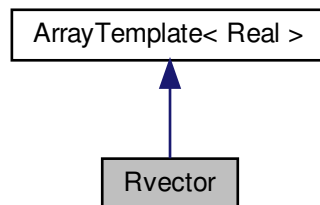
## 8.58 Rvector Class Reference

```
#include <Rvector.hpp>
```

Inheritance diagram for Rvector:



Collaboration diagram for Rvector:



### Classes

- class [ZeroVector](#)

### Public Member Functions

- [Rvector](#) ()
- [Rvector](#) (int size)
- [Rvector](#) (int size, [Real](#) a1,...)
- [Rvector](#) (const [RealArray](#) &ra)
- [Rvector](#) (const [Rvector](#) &v)

- virtual `~Rvector` ()
- void `Set` (int numElem, `Real` a1,...)
- void `Set` (`Real` \*data, int size=0)
- virtual `RealArray GetRealArray` () const
- virtual `Real GetMagnitude` () const
- `Rvector GetUnitRvector` () const
- const `Rvector` & `Normalize` ()
- const `Rvector` & `operator=` (const `Rvector` &v)
- bool `operator==` (const `Rvector` &v) const
- bool `operator!=` (const `Rvector` &v) const
- `Rvector operator-` () const
- `Rvector operator+` (const `Rvector` &v) const
- const `Rvector` & `operator+=` (const `Rvector` &v)
- `Rvector operator-` (const `Rvector` &v) const
- const `Rvector` & `operator-=` (const `Rvector` &v)
- `Rvector operator*` (`Real` s) const
- const `Rvector` & `operator*=` (`Real` s)
- `Real operator*` (const `Rvector` &v) const
- `Rvector operator/` (`Real` s) const
- const `Rvector` & `operator/=` (`Real` s)
- `Real Max` ()
- `Real Min` ()
- void `Sort` (bool ascending=true)
- void `Sort` (`IntegerArray` &indices, bool ascending=true)
- `Rvector operator*` (const `Rmatrix` &m) const
- const `Rvector` & `operator*=` (const `Rmatrix` &m)
- `Rvector operator/` (const `Rmatrix` &m) const
- const `Rvector` & `operator/=` (const `Rmatrix` &m)
- virtual bool `MakeZeroVector` ()
- virtual bool `IsZeroVector` () const
- `Real Norm` ()
- virtual std::string `ToString` (const std::string &format, `Integer` col) const
- virtual std::string `ToString` (`Integer` precision, bool horizontal=true, const std::string &prefix="") const
- virtual std::string `ToString` (bool useCurrentFormat=true, bool scientific=false, bool showPoint=false, `Integer` precision=`GmatGlobal::DATA_PRECISION`, `Integer` width=`GmatGlobal::DATA_WIDTH`, bool horizontal=true, `Integer` spacing=1, const std::string &prefix="", bool appendEol=false) const

## Friends

- class `Rmatrix`
- `Rvector GMATUTIL_API operator*` (`Real` s, const `Rvector` &v)
- `Rmatrix GMATUTIL_API Outerproduct` (const `Rvector` &v1, const `Rvector` &v2)
- std::istream & `operator>>` (std::istream &input, `Rvector` &a)
- std::ostream & `operator<<` (std::ostream &output, const `Rvector` &a)

## Additional Inherited Members

### 8.58.1 Constructor & Destructor Documentation

#### 8.58.1.1 Rvector() [1/5]

```
Rvector::Rvector ()
```

Declarations for the [Rvector](#) class, providing linear algebra operations for the general n-element Real vector.

#### 8.58.1.2 Rvector() [2/5]

```
Rvector::Rvector (
 int size)
```

#### 8.58.1.3 Rvector() [3/5]

```
Rvector::Rvector (
 int size,
 Real a1,
 ...)
```

#### Note

. is required for variable length Real values. eg) 123., 100.

#### 8.58.1.4 Rvector() [4/5]

```
Rvector::Rvector (
 const RealArray & ra)
```

#### 8.58.1.5 Rvector() [5/5]

```
Rvector::Rvector (
 const Rvector & v)
```

#### 8.58.1.6 ~Rvector()

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

### 8.58.2 Member Function Documentation

### 8.58.2.1 GetMagnitude()

```
Real Rvector::GetMagnitude () const [virtual]
```

Reimplemented in [Rvector3](#).

### 8.58.2.2 GetRealArray()

```
RealArray Rvector::GetRealArray () const [virtual]
```

### 8.58.2.3 GetUnitRvector()

```
Rvector Rvector::GetUnitRvector () const
```

### 8.58.2.4 IsZeroVector()

```
bool Rvector::IsZeroVector () const [virtual]
```

### 8.58.2.5 MakeZeroVector()

```
bool Rvector::MakeZeroVector () [virtual]
```

### 8.58.2.6 Max()

```
Real Rvector::Max ()
```

returns maximum value in vector

#### Returns

maximum value contained in this vector

#### 8.58.2.7 Min()

```
Real Rvector::Min ()
```

returns minimum value in vector

##### Returns

minimum value contained in this vector

#### 8.58.2.8 Norm()

```
Real Rvector::Norm ()
```

This method calls returns the square root of the sum of the squares.

#### 8.58.2.9 Normalize()

```
const Rvector & Rvector::Normalize ()
```

#### 8.58.2.10 operator!=(=)

```
bool Rvector::operator!= (
 const Rvector & v) const
```

#### 8.58.2.11 operator\*() [1/3]

```
Rvector Rvector::operator* (
 Real s) const
```

#### 8.58.2.12 operator\*() [2/3]

```
Real Rvector::operator* (
 const Rvector & v) const
```

**8.58.2.13** `operator*()` [3/3]

```
Rvector Rvector::operator* (
 const Rmatrix & m) const
```

**Note**

Here vector is treated as a row matrix 1 x N.

**8.58.2.14** `operator*=( )` [1/2]

```
const Rvector & Rvector::operator*= (
 Real s)
```

**8.58.2.15** `operator*=( )` [2/2]

```
const Rvector & Rvector::operator*= (
 const Rmatrix & m)
```

**Note**

Here vector is treated as a row matrix 1 x N.

**8.58.2.16** `operator+( )`

```
Rvector Rvector::operator+ (
 const Rvector & v) const
```

**8.58.2.17** `operator+=( )`

```
const Rvector & Rvector::operator+= (
 const Rvector & v)
```

**8.58.2.18** `operator-( )` [1/2]

```
Rvector Rvector::operator- () const
```

**8.58.2.19** `operator-()` [2/2]

```
Rvector Rvector::operator- (
 const Rvector & v) const
```

**8.58.2.20** `operator-=()`

```
const Rvector & Rvector::operator-= (
 const Rvector & v)
```

**8.58.2.21** `operator/()` [1/2]

```
Rvector Rvector::operator/ (
 Real s) const
```

**8.58.2.22** `operator/()` [2/2]

```
Rvector Rvector::operator/ (
 const Rmatrix & m) const
```

**8.58.2.23** `operator/=()` [1/2]

```
const Rvector & Rvector::operator/= (
 Real s)
```

**8.58.2.24** `operator/=()` [2/2]

```
const Rvector & Rvector::operator/= (
 const Rmatrix & m)
```

**8.58.2.25** `operator=()`

```
const Rvector & Rvector::operator= (
 const Rvector & v)
```



**8.58.2.26 operator==()**

```
bool Rvector::operator== (
 const Rvector & v) const
```

**8.58.2.27 Set()** [1/2]

```
void Rvector::Set (
 int numElem,
 Real a1,
 ...)
```

**Note**

. is required for variable length Real values. eg) 123., 100.

**8.58.2.28 Set()** [2/2]

```
void Rvector::Set (
 Real * data,
 int size = 0)
```

**8.58.2.29 Sort()** [1/2]

```
void Rvector::Sort (
 bool ascending = true)
```

sorts an [Rvector](#) in ascending or descending order

**Parameters**

|                  |                                                           |
|------------------|-----------------------------------------------------------|
| <i>ascending</i> | input of true => ascending sort, false => descending sort |
|------------------|-----------------------------------------------------------|

**8.58.2.30 Sort()** [2/2]

```
void Rvector::Sort (
 IntegerArray & indices,
 bool ascending = true)
```

sorts an [Rvector](#) in ascending or descending order, returning an array containing the original indices of the values in the updated [Rvector](#)

**Note**

this sort matches the Matlab sort that returns an index array

**Parameters**

|                  |                                                           |
|------------------|-----------------------------------------------------------|
| <i>indices</i>   | array containing original indices after sort is done      |
| <i>ascending</i> | input of true => ascending sort, false => descending sort |

**8.58.2.31 ToString()** [1/3]

```
std::string Rvector::ToString (
 const std::string & format,
 Integer col) const [virtual]
```

**8.58.2.32 ToString()** [2/3]

```
std::string Rvector::ToString (
 Integer precision,
 bool horizontal = true,
 const std::string & prefix = "") const [virtual]
```

**8.58.2.33 ToString()** [3/3]

```
std::string Rvector::ToString (
 bool useCurrentFormat = true,
 bool scientific = false,
 bool showPoint = false,
 Integer precision = GmatGlobal::DATA_PRECISION,
 Integer width = GmatGlobal::DATA_WIDTH,
 bool horizontal = true,
 Integer spacing = 1,
 const std::string & prefix = "",
 bool appendEol = false) const [virtual]
```

**8.58.3 Friends And Related Function Documentation**

#### 8.58.3.1 operator\*

```
Rvector GMATUTIL_API operator* (
 Real s,
 const Rvector & v) [friend]
```

#### 8.58.3.2 operator<<

```
std::ostream& operator<< (
 std::ostream & output,
 const Rvector & a) [friend]
```

#### 8.58.3.3 operator>>

```
std::istream& operator>> (
 std::istream & input,
 Rvector & a) [friend]
```

#### 8.58.3.4 Outerproduct

```
Rmatrix GMATUTIL_API Outerproduct (
 const Rvector & v1,
 const Rvector & v2) [friend]
```

#### Note

[Rmatrix](#) multiplication, with first vector considered a Nx1 matrix and a second vector considered a 1xM matrix.

#### 8.58.3.5 Rmatrix

```
friend class Rmatrix [friend]
```

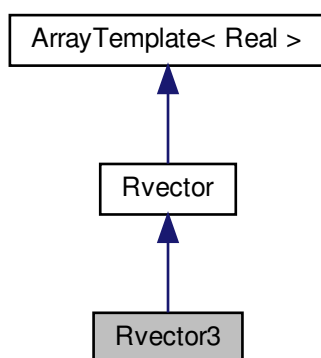
The documentation for this class was generated from the following files:

- GMATsrc/util/[Rvector.hpp](#)
- GMATsrc/util/[Rvector.cpp](#)

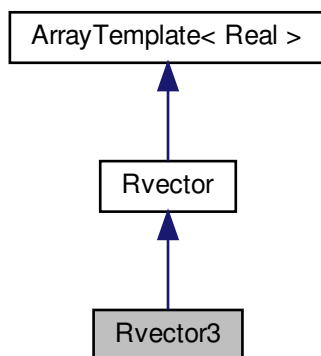
## 8.59 Rvector3 Class Reference

```
#include <Rvector3.hpp>
```

Inheritance diagram for Rvector3:



Collaboration diagram for Rvector3:



### Public Member Functions

- [Rvector3](#) ()
- [Rvector3](#) (const [Real](#) e1, const [Real](#) e2, const [Real](#) e3)
- [Rvector3](#) (const [Rvector3](#) &v)
- [Rvector3](#) (const [RealArray](#) &ra)
- virtual [~Rvector3](#) ()

- [Real Get](#) ([Integer](#) index) const
- void [Set](#) (const [Real](#) e1, const [Real](#) e2, const [Real](#) e3)
- virtual [Real GetMagnitude](#) () const
- [Rvector3 GetUnitVector](#) () const
- const [Rvector3](#) & [Normalize](#) ()
- void [ComputeLongitudeLatitude](#) ([Real](#) &lon, [Real](#) &lat)
- [Rvector3](#) & [operator=](#) (const [Rvector3](#) &v)
- bool [operator==](#) (const [Rvector3](#) &v) const
- bool [operator!=](#) (const [Rvector3](#) &v) const
- [Rvector3 operator-](#) () const
- [Rvector3 operator+](#) (const [Rvector3](#) &v) const
- const [Rvector3](#) & [operator+=](#) (const [Rvector3](#) &v)
- [Rvector3 operator-](#) (const [Rvector3](#) &v) const
- const [Rvector3](#) & [operator-=](#) (const [Rvector3](#) &v)
- [Rvector3 operator\\*](#) ([Real](#) s) const
- const [Rvector3](#) & [operator\\*=](#) ([Real](#) s)
- [Real operator\\*](#) (const [Rvector3](#) &v) const
- [Rvector3 operator/](#) ([Real](#) s) const
- const [Rvector3](#) & [operator/=](#) ([Real](#) s)
- [Rvector3 operator\\*](#) (const [Rmatrix33](#) &m) const
- const [Rvector3](#) & [operator\\*=](#) (const [Rmatrix33](#) &m)
- [Rvector3 operator/](#) (const [Rmatrix33](#) &m) const
- const [Rvector3](#) & [operator/=](#) (const [Rmatrix33](#) &m)
- [Integer GetNumData](#) () const
- const std::string \* [GetDataDescriptions](#) () const

### Static Public Member Functions

- static [Real Normalize](#) (const [Real](#) from[3], [Real](#) to[3])
- static void [Copy](#) (const [Real](#) from[3], [Real](#) to[3])

### Friends

- class [Rmatrix33](#)
- [Rvector3 GMATUTIL\\_API operator\\*](#) ([Real](#) s, const [Rvector3](#) &v)
- [Rmatrix33 GMATUTIL\\_API Outerproduct](#) (const [Rvector3](#) &v1, const [Rvector3](#) &v2)
- [Rvector3 GMATUTIL\\_API Cross](#) (const [Rvector3](#) &v1, const [Rvector3](#) &v2)

### Additional Inherited Members

## 8.59.1 Constructor & Destructor Documentation

### 8.59.1.1 [Rvector3\(\)](#) [1/4]

```
Rvector3::Rvector3 ()
```

**8.59.1.2 Rvector3()** [2/4]

```
Rvector3::Rvector3 (
 const Real e1,
 const Real e2,
 const Real e3)
```

**8.59.1.3 Rvector3()** [3/4]

```
Rvector3::Rvector3 (
 const Rvector3 & v)
```

**8.59.1.4 Rvector3()** [4/4]

```
Rvector3::Rvector3 (
 const RealArray & ra)
```

**8.59.1.5 ~Rvector3()**

```
Rvector3::~~Rvector3 () [virtual]
```

**8.59.2 Member Function Documentation****8.59.2.1 ComputeLongitudeLatitude()**

```
void Rvector3::ComputeLongitudeLatitude (
 Real & lon,
 Real & lat)
```

Computes longitude and latitude in radians. Longitude returns value between -PI and +PI. Latitude returns value between -PI/2 and +PI/2

**Note**

- Consider using [BodyFixedStateConverter::CartesianToSpherical\(\)](#)

### 8.59.2.2 Copy()

```
void Rvector3::Copy (
 const Real from[3],
 Real to[3]) [static]
```

### 8.59.2.3 Get()

```
Real Rvector3::Get (
 Integer index) const
```

### 8.59.2.4 GetDataDescriptions()

```
const std::string * Rvector3::GetDataDescriptions () const
```

### 8.59.2.5 GetMagnitude()

```
Real Rvector3::GetMagnitude () const [virtual]
```

Reimplemented from [Rvector](#).

### 8.59.2.6 GetNumData()

```
Integer Rvector3::GetNumData () const
```

### 8.59.2.7 GetUnitVector()

```
Rvector3 Rvector3::GetUnitVector () const
```

### 8.59.2.8 Normalize() <sup>[1/2]</sup>

```
const Rvector3 & Rvector3::Normalize ()
```

**8.59.2.9 Normalize()** [2/2]

```
Real Rvector3::Normalize (
 const Real from[3],
 Real to[3]) [static]
```

**8.59.2.10 operator!=(=)**

```
bool Rvector3::operator!= (
 const Rvector3 & v) const
```

**8.59.2.11 operator\*()** [1/3]

```
Rvector3 Rvector3::operator* (
 Real s) const
```

**8.59.2.12 operator\*()** [2/3]

```
Real Rvector3::operator* (
 const Rvector3 & v) const
```

**8.59.2.13 operator\*()** [3/3]

```
Rvector3 Rvector3::operator* (
 const Rmatrix33 & m) const
```

**8.59.2.14 operator\*=(=)** [1/2]

```
const Rvector3 & Rvector3::operator*= (
 Real s)
```

**8.59.2.15 operator\*=(=)** [2/2]

```
const Rvector3 & Rvector3::operator*= (
 const Rmatrix33 & m)
```



**8.59.2.16 operator+()**

```
Rvector3 Rvector3::operator+ (
 const Rvector3 & v) const
```

**8.59.2.17 operator+=()**

```
const Rvector3 & Rvector3::operator+= (
 const Rvector3 & v)
```

**8.59.2.18 operator-()** [1/2]

```
Rvector3 Rvector3::operator- () const
```

**8.59.2.19 operator-()** [2/2]

```
Rvector3 Rvector3::operator- (
 const Rvector3 & v) const
```

**8.59.2.20 operator-=()**

```
const Rvector3 & Rvector3::operator-= (
 const Rvector3 & v)
```

**8.59.2.21 operator/()** [1/2]

```
Rvector3 Rvector3::operator/ (
 Real s) const
```

**8.59.2.22 operator/()** [2/2]

```
Rvector3 Rvector3::operator/ (
 const Rmatrix33 & m) const
```

**8.59.2.23 operator/=( )** [1/2]

```
const Rvector3 & Rvector3::operator/= (
 Real s)
```

**8.59.2.24 operator/=( )** [2/2]

```
const Rvector3 & Rvector3::operator/= (
 const Rmatrix33 & m)
```

**8.59.2.25 operator=( )**

```
Rvector3 & Rvector3::operator= (
 const Rvector3 & v)
```

**8.59.2.26 operator==( )**

```
bool Rvector3::operator== (
 const Rvector3 & v) const
```

**8.59.2.27 Set( )**

```
void Rvector3::Set (
 const Real e1,
 const Real e2,
 const Real e3)
```

**8.59.3 Friends And Related Function Documentation****8.59.3.1 Cross**

```
Rvector3 GMATUTIL_API Cross (
 const Rvector3 & v1,
 const Rvector3 & v2) [friend]
```

## 8.59.3.2 operator\*

```
Rvector3 GMATUTIL_API operator* (
 Real s,
 const Rvector3 & v) [friend]
```

## 8.59.3.3 Outerproduct

```
Rmatrix33 GMATUTIL_API Outerproduct (
 const Rvector3 & v1,
 const Rvector3 & v2) [friend]
```

## 8.59.3.4 Rmatrix33

```
friend class Rmatrix33 [friend]
```

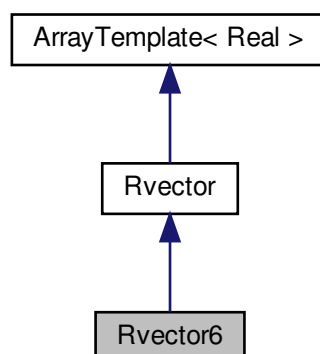
The documentation for this class was generated from the following files:

- GMATsrc/util/Rvector3.hpp
- GMATsrc/util/Rvector3.cpp

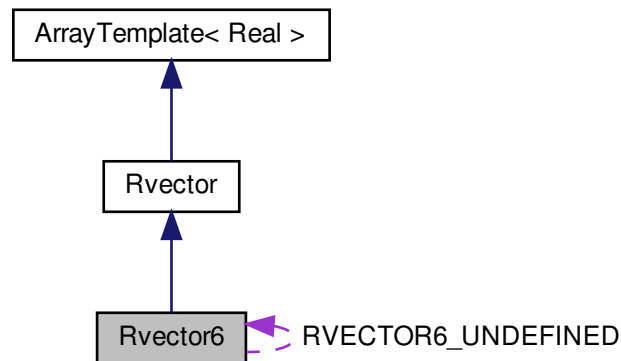
## 8.60 Rvector6 Class Reference

```
#include <Rvector6.hpp>
```

Inheritance diagram for Rvector6:



Collaboration diagram for Rvector6:



## Public Member Functions

- [Rvector6](#) ()
- [Rvector6](#) (const [Real](#) e1, const [Real](#) e2, const [Real](#) e3, const [Real](#) e4, const [Real](#) e5, const [Real](#) e6)
- [Rvector6](#) (const [Rvector3](#) &r, const [Rvector3](#) &v)
- [Rvector6](#) (const [Real](#) vec[6])
- [Rvector6](#) (const [Rvector6](#) &v)
- [Rvector6](#) (const [RealArray](#) &ra)
- [Rvector6](#) & operator= (const [Rvector6](#) &v)
- [Rvector6](#) \* Clone () const
- virtual ~[Rvector6](#) ()
- [Real](#) Get (Integer index) const
- [Rvector3](#) GetR () const
- [Rvector3](#) GetV () const
- void GetR ([Real](#) \*r) const
- void GetV ([Real](#) \*v) const
- void Set (const [Real](#) e1, const [Real](#) e2, const [Real](#) e3, const [Real](#) e4, const [Real](#) e5, const [Real](#) e6)
- void Set (const [Real](#) v[6])
- void SetR (const [Rvector3](#) &v)
- void SetV (const [Rvector3](#) &v)
- bool operator== (const [Rvector6](#) &v) const
- bool operator!= (const [Rvector6](#) &v) const
- [Rvector6](#) operator- () const
- [Rvector6](#) operator+ (const [Rvector6](#) &v) const
- const [Rvector6](#) & operator+= (const [Rvector6](#) &v)
- [Rvector6](#) operator- (const [Rvector6](#) &v) const
- const [Rvector6](#) & operator-= (const [Rvector6](#) &v)
- [Rvector6](#) operator\* ([Real](#) s) const
- const [Rvector6](#) & operator\*= ([Real](#) s)
- [Real](#) operator\* (const [Rvector6](#) &v) const
- [Rvector6](#) operator/ ([Real](#) s) const
- const [Rvector6](#) & operator/= ([Real](#) s)
- [Rvector6](#) operator\* (const [Rmatrix66](#) &m) const

- const [Rvector6](#) & [operator\\*=\(const Rmatrix66 &m\)](#)
- [Rvector6](#) [operator/](#) (const [Rmatrix66](#) &m) const
- const [Rvector6](#) & [operator/=](#) (const [Rmatrix66](#) &m)
- [Integer](#) [GetNumData](#) () const
- const std::string \* [GetDataDescriptions](#) () const
- bool [IsValid](#) (const [Real](#) val)

### Static Public Attributes

- static const [Real](#) [UTIL\\_REAL\\_UNDEFINED](#) = [GmatRealConstants::REAL\\_UNDEFINED](#)
- static const [Rvector6](#) [RVECTOR6\\_UNDEFINED](#)

### Friends

- class [Rmatrix66](#)

### Additional Inherited Members

## 8.60.1 Constructor & Destructor Documentation

### 8.60.1.1 [Rvector6\(\)](#) [1/6]

```
Rvector6::Rvector6 ()
```

Default constructor. Initializes elements to 0.0.

### 8.60.1.2 [Rvector6\(\)](#) [2/6]

```
Rvector6::Rvector6 (
 const Real e1,
 const Real e2,
 const Real e3,
 const Real e4,
 const Real e5,
 const Real e6)
```

Constructor. Creates an object from 6 Real elements.

### 8.60.1.3 [Rvector6\(\)](#) [3/6]

```
Rvector6::Rvector6 (
 const Rvector3 & r,
 const Rvector3 & v)
```

Constructor. Creates an object from two [Rvector3](#) object.

#### 8.60.1.4 Rvector6() [4/6]

```
Rvector6::Rvector6 (
 const Real vec[6])
```

Constructor. Creates an object from Real array

#### 8.60.1.5 Rvector6() [5/6]

```
Rvector6::Rvector6 (
 const Rvector6 & v)
```

Copy constructor.

#### 8.60.1.6 Rvector6() [6/6]

```
Rvector6::Rvector6 (
 const RealArray & ra)
```

#### 8.60.1.7 ~Rvector6()

```
Rvector6::~~Rvector6 () [virtual]
```

Destructor.

### 8.60.2 Member Function Documentation

#### 8.60.2.1 Clone()

```
Rvector6 * Rvector6::Clone () const
```

#### 8.60.2.2 Get()

```
Real Rvector6::Get (
 Integer index) const
```

### 8.60.2.3 GetDataDescriptions()

```
const std::string * Rvector6::GetDataDescriptions () const
```

#### Returns

data description pointer.

### 8.60.2.4 GetNumData()

```
Integer Rvector6::GetNumData () const
```

#### Returns

number of data elements.

### 8.60.2.5 GetR() [1/2]

```
Rvector3 Rvector6::GetR () const
```

#### Returns

[Rvector3](#) object created from first three elements.

### 8.60.2.6 GetR() [2/2]

```
void Rvector6::GetR (
 Real * r) const
```

### 8.60.2.7 GetV() [1/2]

```
Rvector3 Rvector6::GetV () const
```

#### Returns

[Rvector3](#) object created from last three elements.

#### 8.60.2.8 GetV() [2/2]

```
void Rvector6::GetV (
 Real * v) const
```

#### 8.60.2.9 IsValid()

```
bool Rvector6::IsValid (
 const Real val)
```

##### Returns

true if all 6 elements are not equal to input value.

#### 8.60.2.10 operator!=(())

```
bool Rvector6::operator!= (
 const Rvector6 & v) const
```

##### Returns

true if all 6 elements of v are not equal to this object.

#### 8.60.2.11 operator\*(()) [1/3]

```
Rvector6 Rvector6::operator* (
 Real s) const
```

Multiplies a Real number and return a object.

#### 8.60.2.12 operator\*(()) [2/3]

```
Real Rvector6::operator* (
 const Rvector6 & v) const
```

#### 8.60.2.13 operator\*(()) [3/3]

```
Rvector6 Rvector6::operator* (
 const Rmatrix66 & m) const
```



**8.60.2.14 operator\*=( )** [1/2]

```
const Rvector6 & Rvector6::operator*= (
 Real s)
```

Multiplies a Real number and return the same object.

**8.60.2.15 operator\*=( )** [2/2]

```
const Rvector6& Rvector6::operator*= (
 const Rmatrix66 & m)
```

**8.60.2.16 operator+( )**

```
Rvector6 Rvector6::operator+ (
 const Rvector6 & v) const
```

Adds an object and return a new object.

**8.60.2.17 operator+=( )**

```
const Rvector6 & Rvector6::operator+= (
 const Rvector6 & v)
```

Adds an object and return the same object.

**8.60.2.18 operator-( )** [1/2]

```
Rvector6 Rvector6::operator- () const
```

Negates all elements and return a new object.

**8.60.2.19 operator-( )** [2/2]

```
Rvector6 Rvector6::operator- (
 const Rvector6 & v) const
```

Subtracts a object and return a new object.

Returns

a new object.

**8.60.2.20 operator-=( )**

```
const Rvector6 & Rvector6::operator-= (
 const Rvector6 & v)
```

Subtracts an object and return the same object.

**8.60.2.21 operator/( )** [1/2]

```
Rvector6 Rvector6::operator/ (
 Real s) const
```

Divides by a Real number and return a new object.

## Exceptions

|               |                       |
|---------------|-----------------------|
| <i>thrown</i> | when divider is zero. |
|---------------|-----------------------|

8.60.2.22 `operator/()` [2/2]

```
Rvector6 Rvector6::operator/ (
 const Rmatrix66 & m) const
```

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

```
const Rvector6 & Rvector6::operator/= (
 Real s)
```

Divides by a Real number and return the same object.

## Exceptions

|               |                       |
|---------------|-----------------------|
| <i>thrown</i> | when divider is zero. |
|---------------|-----------------------|

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

```
const Rvector6& Rvector6::operator/= (
 const Rmatrix66 & m)
```

8.60.2.25 `operator=( )`

```
Rvector6 & Rvector6::operator= (
 const Rvector6 & v)
```

Assignment operator.

8.60.2.26 `operator==( )`

```
bool Rvector6::operator== (
 const Rvector6 & v) const
```

## Returns

true if all 6 elements of v are equal to this object.

**8.60.2.27 Set()** [1/2]

```
void Rvector6::Set (
 const Real e1,
 const Real e2,
 const Real e3,
 const Real e4,
 const Real e5,
 const Real e6)
```

**8.60.2.28 Set()** [2/2]

```
void Rvector6::Set (
 const Real v[6])
```

**8.60.2.29 SetR()**

```
void Rvector6::SetR (
 const Rvector3 & v)
```

Sets first three elements.

**8.60.2.30 SetV()**

```
void Rvector6::SetV (
 const Rvector3 & v)
```

Sets last three elements.

**8.60.3 Friends And Related Function Documentation****8.60.3.1 Rmatrix66**

```
friend class Rmatrix66 [friend]
```

**8.60.4 Member Data Documentation**

#### 8.60.4.1 RVECTOR6\_UNDEFINED

```
const Rvector6 Rvector6::RVECTOR6_UNDEFINED [static]
```

**Initial value:**

```
= Rvector6(UTIL_REAL_UNDEFINED, UTIL_REAL_UNDEFINED,
 UTIL_REAL_UNDEFINED, UTIL_REAL_UNDEFINED,
 UTIL_REAL_UNDEFINED, UTIL_REAL_UNDEFINED)
```

#### 8.60.4.2 UTIL\_REAL\_UNDEFINED

```
const Real Rvector6::UTIL_REAL_UNDEFINED = GmatRealConstants::REAL_UNDEFINED [static]
```

Provides definitions for the [Rvector6](#) class, providing linear algebra operations for the 6-element Real vector

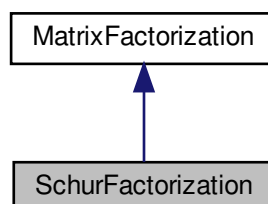
The documentation for this class was generated from the following files:

- GMATsrc/util/[Rvector6.hpp](#)
- GMATsrc/util/[Rvector6.cpp](#)

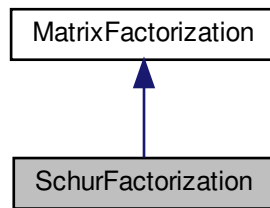
## 8.61 SchurFactorization Class Reference

```
#include <SchurFactorization.hpp>
```

Inheritance diagram for SchurFactorization:



Collaboration diagram for SchurFactorization:



## Public Member Functions

- [SchurFactorization](#) ()
- [SchurFactorization](#) (const [SchurFactorization](#) &schurfactorization)
- [~SchurFactorization](#) ()
- [SchurFactorization & operator=](#) (const [SchurFactorization](#) &schurfactorization)
- virtual void [Factor](#) (const [Rmatrix](#) inputMatrix, [Rmatrix](#) &sdUnitary, [Rmatrix](#) &sdUpper)
- virtual void [Invert](#) ([Rmatrix](#) &inputMatrix)
- virtual void [Invert](#) ([Real](#) \*sum1, [Integer](#) array\_size)
- virtual void [RemoveRowCol](#) ([Real](#) \*sum1, [Integer](#) num\_rows, [Integer](#) row\_to\_remove, [IntegerArray](#) &removed\_rows)
- virtual void [RestoreAllRowCols](#) ([Real](#) \*sum1, [Integer](#) num\_rows, [IntegerArray](#) &removed\_rows)

## Additional Inherited Members

### 8.61.1 Detailed Description

Declares [SchurFactorization](#) class.

### 8.61.2 Constructor & Destructor Documentation

#### 8.61.2.1 SchurFactorization() [1/2]

```
SchurFactorization::SchurFactorization ()
```

Declares [SchurFactorization](#) class. Constructor

### 8.61.2.2 SchurFactorization() [2/2]

```
SchurFactorization::SchurFactorization (
 const SchurFactorization & schurfactorization)
```

Copy Constructor

### 8.61.2.3 ~SchurFactorization()

```
SchurFactorization::~~SchurFactorization ()
```

Destructor

## 8.61.3 Member Function Documentation

### 8.61.3.1 Factor()

```
void SchurFactorization::Factor (
 const Rmatrix inputMatrix,
 Rmatrix & sdUnitary,
 Rmatrix & sdUpper) [virtual]
```

Matrix factorization routine using the Schur identity

Factorization is completed by using QR factorization:  $A \rightarrow Q, R$   $A = R*Q$  This pattern is used until a tolerance is reached.  $A$  will become *sdUpper* through this process and *sdUnitary* will come from multiplying all  $Q$  values calculated together. Original equation by Padraic Bartlett at the end of the document "Lecture 5: The Schur Decomposition".

#### Parameters

|                       |                                     |
|-----------------------|-------------------------------------|
| <i>inputMatrix</i>    | The matrix to factor                |
| <i>&amp;sdUnitary</i> | Unitary matrix $Q$ where $Q'AQ = U$ |
| <i>&amp;sdUpper</i>   | Upper triangular matrix $U$         |

#### Returns

0 on success, anything else indicates a problem

Implements [MatrixFactorization](#).

### 8.61.3.2 Invert() [1/2]

```
void SchurFactorization::Invert (
 Rmatrix & inputMatrix) [virtual]
```

Matrix inversion routine using the Schur identity

This method is a port of the inversion code from GTDS, as ported by Angel Wang of Thinking Systems and then integrated into GMAT by D. Conway.

#### Parameters

|                    |                           |
|--------------------|---------------------------|
| <i>inputMatrix</i> | The matrix to be inverted |
|--------------------|---------------------------|

#### Returns

0 on success, anything else indicates a problem

Implements [MatrixFactorization](#).

#### 8.61.3.3 Invert() [2/2]

```
void SchurFactorization::Invert (
 Real * sum1,
 Integer array_size) [virtual]
```

Matrix inversion routine using the Schur identity

This method is a port of the inversion code from GTDS, as ported by Angel Wang of Thinking Systems and then integrated into GMAT by D. Conway.

#### Parameters

|                   |                                                            |
|-------------------|------------------------------------------------------------|
| <i>sum1</i>       | The matrix to be inverted, packed in upper triangular form |
| <i>array_size</i> | The size of the sum1 array                                 |

#### 8.61.3.4 operator=()

```
SchurFactorization & SchurFactorization::operator= (
 const SchurFactorization & schurfactorization)
```

Assignment operator

#### 8.61.3.5 RemoveRowCol()

```
void SchurFactorization::RemoveRowCol (
 Real * sum1,
 Integer num_rows,
 Integer row_to_remove,
 IntegerArray & removed_rows) [virtual]
```

Remove a row/column from sum1, the upper triangle of a square symmetric matrix packed into an array. Move elements past the removed row/column down, and fill the spaces created with zeros.

## Parameters

|                      |                                                                                                                                             |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <i>sum1</i>          | The upper triangle of a square symmetric matrix packed into an array                                                                        |
| <i>num_rows</i>      | The number of rows and columns in the matrix                                                                                                |
| <i>row_to_remove</i> | The index of the row/column to remove from the matrix. $0 \leq \text{row\_to\_remove} < \text{num\_rows}$                                   |
| <i>removed_rows</i>  | An array containing the indexes of the rows/columns which were removed from the matrix. <i>row_to_remove</i> will be appended to this array |

## 8.61.3.6 RestoreAllRowCols()

```
void SchurFactorization::RestoreAllRowCols (
 Real * sum1,
 Integer num_rows,
 IntegerArray & removed_rows) [virtual]
```

*sum1* is the upper triangle of a square symmetric matrix packed into an array. *removed\_rows* contains the indexes of the rows/columns which were removed from *sum1*. *RestoreAllRowCols* will expand *sum1* back to its original size, filling in rows/columns of zeros at the row indexes where rows/columns were previously removed.

## Parameters

|                     |                                                                                                                               |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------|
| <i>sum1</i>         | The upper triangle of a square symmetric matrix packed into an array                                                          |
| <i>num_rows</i>     | The number of rows and columns in the matrix                                                                                  |
| <i>removed_rows</i> | An array containing the indexes of the rows/columns which were removed from the matrix. <i>row_to_remove</i> will be appended |

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

- GMATsrc/util/matrixoperations/[SchurFactorization.hpp](#)
- GMATsrc/util/matrixoperations/[SchurFactorization.cpp](#)

## 8.62 StateConversionUtil Class Reference

```
#include <StateConversionUtil.hpp>
```

## Public Types

- enum [StateType](#) {  
[CARTESIAN](#), [KEPLERIAN](#), [MOD\\_KEPLERIAN](#), [SPH\\_AZFPA](#),  
[SPH\\_RADEC](#), [EQUINOCTIAL](#), [MOD\\_EQUINOCTIAL](#), [ALT\\_EQUINOCTIAL](#),  
[DELAUNAY](#), [PLANETODETIC](#), [OUT\\_ASYM](#), [IN\\_ASYM](#),  
[BROLYD\\_SHORT](#), [BROLYD\\_LONG](#), [StateTypeCount](#) }
- enum [AnomalyType](#) {  
[TA](#), [MA](#), [EA](#), [HA](#),  
[AnomalyTypeCount](#) }



## Static Public Member Functions

- static [Rvector6 Convert](#) (const [Real](#) \*state, const std::string &fromType, const std::string &toType, [Real](#) mu=[GmatSolarSystemDefaults::PLANET\\_MU](#)[[GmatSolarSystemDefaults::EARTH](#)], [Real](#) flattening=[GmatSolarSystemDefaults::PLANET\\_FLATTENING](#)[[GmatSolarSystemDefaults::EARTH](#)], [Real](#) eqRadius=[GmatSolarSystemDefaults::PLANET\\_EQUATORIAL\\_RADIUS](#)[[GmatSolarSystemDefaults::EARTH](#)], const std::string &anomalyType="TA")
- static [Rvector6 Convert](#) (const [Rvector6](#) &state, const std::string &fromType, const std::string &toType, [Real](#) mu=[GmatSolarSystemDefaults::PLANET\\_MU](#)[[GmatSolarSystemDefaults::EARTH](#)], [Real](#) flattening=[GmatSolarSystemDefaults::PLANET\\_FLATTENING](#)[[GmatSolarSystemDefaults::EARTH](#)], [Real](#) eqRadius=[GmatSolarSystemDefaults::PLANET\\_EQUATORIAL\\_RADIUS](#)[[GmatSolarSystemDefaults::EARTH](#)], const std::string &anomalyType="TA")
- static [Rvector6 ConvertFromCartesian](#) (const std::string &toType, const [Rvector6](#) &state, [Real](#) mu=[EARTH\\_MU](#), const std::string &anomalyType="TA", [Real](#) flattening=[EARTH\\_FLATTENING](#), [Real](#) eqRadius=[EARTH\\_EQ\\_RADIUS](#))
- static [Rvector6 ConvertFromKeplerian](#) (const std::string &toType, const [Rvector6](#) &state, [Real](#) mu=[EARTH\\_MU](#), const std::string &anomalyType="TA", [Real](#) flattening=[EARTH\\_FLATTENING](#), [Real](#) eqRadius=[EARTH\\_EQ\\_RADIUS](#))
- static [Rvector6 ConvertFromModKeplerian](#) (const std::string &toType, const [Rvector6](#) &state, [Real](#) mu=[EARTH\\_MU](#), const std::string &anomalyType="TA", [Real](#) flattening=[EARTH\\_FLATTENING](#), [Real](#) eqRadius=[EARTH\\_EQ\\_RADIUS](#))
- static [Rvector6 ConvertFromSphericalIAZFPA](#) (const std::string &toType, const [Rvector6](#) &state, [Real](#) mu=[EARTH\\_MU](#), const std::string &anomalyType="TA", [Real](#) flattening=[EARTH\\_FLATTENING](#), [Real](#) eqRadius=[EARTH\\_EQ\\_RADIUS](#))
- static [Rvector6 ConvertFromSphericalIADEC](#) (const std::string &toType, const [Rvector6](#) &state, [Real](#) mu=[EARTH\\_MU](#), const std::string &anomalyType="TA", [Real](#) flattening=[EARTH\\_FLATTENING](#), [Real](#) eqRadius=[EARTH\\_EQ\\_RADIUS](#))
- static [Rvector6 ConvertFromEquinoctial](#) (const std::string &toType, const [Rvector6](#) &state, [Real](#) mu=[EARTH\\_MU](#), const std::string &anomalyType="TA", [Real](#) flattening=[EARTH\\_FLATTENING](#), [Real](#) eqRadius=[EARTH\\_EQ\\_RADIUS](#))
- static [Rvector6 ConvertFromModEquinoctial](#) (const std::string &toType, const [Rvector6](#) &state, [Real](#) mu=[EARTH\\_MU](#), const std::string &anomalyType="TA", [Real](#) flattening=[EARTH\\_FLATTENING](#), [Real](#) eqRadius=[EARTH\\_EQ\\_RADIUS](#))
- static [Rvector6 ConvertFromAltEquinoctial](#) (const std::string &toType, const [Rvector6](#) &state, [Real](#) mu=[EARTH\\_MU](#), const std::string &anomalyType="TA", [Real](#) flattening=[EARTH\\_FLATTENING](#), [Real](#) eqRadius=[EARTH\\_EQ\\_RADIUS](#))
- static [Rvector6 ConvertFromDelaunay](#) (const std::string &toType, const [Rvector6](#) &state, [Real](#) mu=[EARTH\\_MU](#), const std::string &anomalyType="TA", [Real](#) flattening=[EARTH\\_FLATTENING](#), [Real](#) eqRadius=[EARTH\\_EQ\\_RADIUS](#))
- static [Rvector6 ConvertFromPlanetodetic](#) (const std::string &toType, const [Rvector6](#) &state, [Real](#) mu=[EARTH\\_MU](#), const std::string &anomalyType="TA", [Real](#) flattening=[EARTH\\_FLATTENING](#), [Real](#) eqRadius=[EARTH\\_EQ\\_RADIUS](#))
- static [Rvector6 ConvertFromIncomingAsymptote](#) (const std::string &toType, const [Rvector6](#) &state, [Real](#) mu=[EARTH\\_MU](#), const std::string &anomalyType="TA", [Real](#) flattening=[EARTH\\_FLATTENING](#), [Real](#) eqRadius=[EARTH\\_EQ\\_RADIUS](#))
- static [Rvector6 ConvertFromOutgoingAsymptote](#) (const std::string &toType, const [Rvector6](#) &state, [Real](#) mu=[EARTH\\_MU](#), const std::string &anomalyType="TA", [Real](#) flattening=[EARTH\\_FLATTENING](#), [Real](#) eqRadius=[EARTH\\_EQ\\_RADIUS](#))
- static [Rvector6 ConvertFromBrouwerMeanShort](#) (const std::string &toType, const [Rvector6](#) &state, [Real](#) mu=[EARTH\\_MU](#), const std::string &anomalyType="TA", [Real](#) flattening=[EARTH\\_FLATTENING](#), [Real](#) eqRadius=[EARTH\\_EQ\\_RADIUS](#))
- static [Rvector6 ConvertFromBrouwerMeanLong](#) (const std::string &toType, const [Rvector6](#) &state, [Real](#) mu=[EARTH\\_MU](#), const std::string &anomalyType="TA", [Real](#) flattening=[EARTH\\_FLATTENING](#), [Real](#) eqRadius=[EARTH\\_EQ\\_RADIUS](#))
- static [Rvector6 CartesianToKeplerian](#) ([Real](#) mu, const [Rvector3](#) &pos, const [Rvector3](#) &vel, [AnomalyType](#))
- static [Rvector6 CartesianToKeplerian](#) ([Real](#) mu, const [Rvector3](#) &pos, const [Rvector3](#) &vel, const std::string &anomalyType="TA")

- static [Rvector6](#) CartesianToKeplerian ([Real](#) mu, const [Rvector6](#) &state, [AnomalyType](#) anomalyType)
- static [Rvector6](#) CartesianToKeplerian ([Real](#) mu, const [Rvector6](#) &state, const std::string &anomalyType="TA")
- static [Rvector6](#) CartesianToKeplerian ([Real](#) mu, const [Rvector6](#) &state, [Real](#) \*ma)
- static [Rvector6](#) KeplerianToCartesian ([Real](#) mu, const [Rvector6](#) &state, [AnomalyType](#) anomalyType)
- static [Rvector6](#) KeplerianToCartesian ([Real](#) mu, const [Rvector6](#) &state, const std::string &anomalyType="TA")
- static [Rvector6](#) CartesianToSphericalAZFPA (const [Rvector6](#) &cartesian)
- static [Rvector6](#) SphericalAZFPAToCartesian (const [Rvector6](#) &spherical)
- static [Rvector6](#) CartesianToSphericalRADEC (const [Rvector6](#) &cartesian)
- static [Rvector6](#) SphericalRADECToCartesian (const [Rvector6](#) &spherical)
- static [Rvector6](#) KeplerianToModKeplerian (const [Rvector6](#) &keplerian)
- static [Rvector6](#) ModKeplerianToKeplerian (const [Rvector6](#) &modKeplerian)
- static [Rvector6](#) CartesianToEquinoctial (const [Rvector6](#) &cartesian, const [Real](#) &mu)
- static [Rvector6](#) EquinoctialToCartesian (const [Rvector6](#) &equinoctial, const [Real](#) &mu)
- static [Rvector6](#) CartesianToModEquinoctial (const [Rvector6](#) &cartesian, const [Real](#) &mu)
- static [Rvector6](#) ModEquinoctialToCartesian (const [Rvector6](#) &modequinoctial, const [Real](#) &mu)
- static [Rvector6](#) KeplerianToDelaunay (const [Rvector6](#) &keplerian, const [Real](#) &mu)
- static [Rvector6](#) DelaunayToKeplerian (const [Rvector6](#) &delaunay, const [Real](#) &mu)
- static [Rvector6](#) CartesianToPlanetodetic (const [Rvector6](#) &cartesian, [Real](#) flattening, [Real](#) eqRadius)
- static [Rvector6](#) PlanetodeticToCartesian (const [Rvector6](#) &planetodetic, [Real](#) flattening, [Real](#) eqRadius)
- static [Rvector6](#) CartesianToIncomingAsymptote ([Real](#) mu, const [Rvector6](#) &cartesian)
- static [Rvector6](#) IncomingAsymptoteToCartesian ([Real](#) mu, const [Rvector6](#) &inasym)
- static [Rvector6](#) CartesianToOutgoingAsymptote ([Real](#) mu, const [Rvector6](#) &cartesian)
- static [Rvector6](#) OutgoingAsymptoteToCartesian ([Real](#) mu, const [Rvector6](#) &outasym)
- static [Rvector6](#) CartesianToBrouwerMeanShort ([Real](#) mu, const [Rvector6](#) &cartesian)
- static [Rvector6](#) BrouwerMeanShortToOsculatingElements ([Real](#) mu, const [Rvector6](#) &blms)
- static [Rvector6](#) BrouwerMeanShortToCartesian ([Real](#) mu, const [Rvector6](#) &blms)
- static [Rvector6](#) CartesianToBrouwerMeanLong ([Real](#) mu, const [Rvector6](#) &cartesian)
- static [Rvector6](#) BrouwerMeanLongToOsculatingElements ([Real](#) mu, const [Rvector6](#) &blml)
- static [Rvector6](#) BrouwerMeanLongToCartesian ([Real](#) mu, const [Rvector6](#) &blml)
- static [Rvector6](#) EquinoctialToAltEquinoctial (const [Rvector6](#) &equinoctial)
- static [Rvector6](#) AltEquinoctialToEquinoctial (const [Rvector6](#) &altequinoctial)
- static [Real](#) TrueToMeanAnomaly ([Real](#) taRadians, [Real](#) ecc, bool modBy2Pi=false)
- static [Real](#) TrueToEccentricAnomaly ([Real](#) taRadians, [Real](#) ecc, bool modBy2Pi=false)
- static [Real](#) TrueToHyperbolicAnomaly ([Real](#) taRadians, [Real](#) ecc, bool modBy2Pi=false)
- static [Real](#) MeanToTrueAnomaly ([Real](#) maRadians, [Real](#) ecc, [Real](#) tol=1.0e-08)
- static [Real](#) EccentricToTrueAnomaly ([Real](#) eaRadians, [Real](#) ecc, bool modBy2Pi=false)
- static [Real](#) HyperbolicToTrueAnomaly ([Real](#) haRadians, [Real](#) ecc, bool modBy2Pi=false)
- static [Real](#) ConvertFromTrueAnomaly (const std::string &toType, [Real](#) taRadians, [Real](#) ecc, bool modBy2Pi=false)
- static [Real](#) ConvertFromTrueAnomaly ([AnomalyType](#) toType, [Real](#) taRadians, [Real](#) ecc, bool modBy2Pi=false)
- static [Real](#) ConvertToTrueAnomaly (const std::string &fromType, [Real](#) taRadians, [Real](#) ecc, bool modBy2Pi=false)
- static [Real](#) ConvertToTrueAnomaly ([AnomalyType](#) fromType, [Real](#) taRadians, [Real](#) ecc, bool modBy2Pi=false)
- static [Real](#) CartesianToTA ([Real](#) mu, const [Rvector3](#) &pos, const [Rvector3](#) &vel, bool inRadian=false)
- static [Real](#) CartesianToMA ([Real](#) mu, const [Rvector3](#) &pos, const [Rvector3](#) &vel, bool inRadian=false)
- static [Real](#) CartesianToEA ([Real](#) mu, const [Rvector3](#) &pos, const [Rvector3](#) &vel, bool inRadian=false)
- static [Real](#) CartesianToHA ([Real](#) mu, const [Rvector3](#) &pos, const [Rvector3](#) &vel, bool inRadian=false)
- static [Real](#) CartesianToSMA ([Real](#) mu, const [Rvector3](#) &pos, const [Rvector3](#) &vel)
- static [Real](#) CartesianToECC ([Real](#) mu, const [Rvector3](#) &pos, const [Rvector3](#) &vel)
- static [Real](#) CartesianToINC ([Real](#) mu, const [Rvector3](#) &pos, const [Rvector3](#) &vel, bool inRadian=false)
- static [Real](#) CartesianToRAAN ([Real](#) mu, const [Rvector3](#) &pos, const [Rvector3](#) &vel, bool inRadian=false)
- static [Real](#) CartesianToAOP ([Real](#) mu, const [Rvector3](#) &pos, const [Rvector3](#) &vel, bool inRadian=false)
- static [Rvector3](#) CartesianToEccVector ([Real](#) mu, const [Rvector3](#) &pos, const [Rvector3](#) &vel)
- static [Rvector3](#) CartesianToDirOfLineOfNode (const [Rvector3](#) &pos, const [Rvector3](#) &vel)

- static [Rvector6 CartesianToAngularMomentum](#) ([Real](#) mu, const [Rvector3](#) &pos, const [Rvector3](#) &vel)
- static [Real CalculateEccentricAnomaly](#) ([Real](#) e, [Real](#) M)
- static [Rmatrix66 CartesianToKeplerianDerivativeConversion](#) ([Real](#) mu, const [Rvector6](#) &cartesianState)
- static [Rmatrix66 CartesianToKeplerianDerivativeConversionWithKeplInput](#) ([Real](#) mu, const [Rvector6](#) &keplerState)
- static [Rmatrix66 CartesianToKeplerianDerivativeConversion\\_FiniteDiff](#) ([Real](#) mu, const [Rvector6](#) &cartesianState)
- static [Rmatrix66 CartesianToKeplerianDerivativeConversionWithKeplInput\\_FiniteDiff](#) ([Real](#) mu, const [Rvector6](#) &keplerianState)
- static bool [ValidateValue](#) (const std::string &label, [Real](#) value, const std::string &errorMsgFmt, [Integer](#) dataPrecision, const std::string &compareTo="", [Real](#) compareValue=0.0)
- static [Integer GetTypeCount](#) ()
- static const std::string \* [GetStateTypeList](#) ()
- static bool [RequiresCelestialBodyOrigin](#) (const std::string &type)
- static bool [RequiresFixedCoordinateSystem](#) (const std::string &type)
- static [AnomalyType GetAnomalyType](#) (const std::string &typeStr)
- static bool [IsValidAnomalyType](#) (const std::string &anomType)
- static bool [IsRvValid](#) ([Real](#) \*r, [Real](#) \*v)
- static const std::string \* [GetLongTypeNameList](#) ()
- static std::string [GetAnomalyShortText](#) (const std::string &anomalyType)
- static std::string [GetAnomalyLongText](#) (const std::string &anomalyType)

### 8.62.1 Detailed Description

Definition of the static class containing methods to convert between orbit state representations. This is a static class: No instances of this class may be declared.

for BodyFixed states: Cartesian states are (x,y,z) Spherical and Spherical-Ellipsoid states are (latitude, longitude, height)

### 8.62.2 Member Enumeration Documentation

#### 8.62.2.1 AnomalyType

```
enum StateConversionUtil::AnomalyType
```

##### Enumerator

|                  |  |
|------------------|--|
| TA               |  |
| MA               |  |
| EA               |  |
| HA               |  |
| AnomalyTypeCount |  |

### 8.62.2.2 StateType

```
enum StateConversionUtil::StateType
```

#### Enumerator

|                 |  |
|-----------------|--|
| CARTESIAN       |  |
| KEPLERIAN       |  |
| MOD_KEPLERIAN   |  |
| SPH_AZFPA       |  |
| SPH_RADEC       |  |
| EQUINOCTIAL     |  |
| MOD_EQUINOCTIAL |  |
| ALT_EQUINOCTIAL |  |
| DELAUNAY        |  |
| PLANETODETIC    |  |
| OUT_ASYM        |  |
| IN_ASYM         |  |
| BROLYD_SHORT    |  |
| BROLYD_LONG     |  |
| StateTypeCount  |  |

## 8.62.3 Member Function Documentation

### 8.62.3.1 AltEquinoctialToEquinoctial()

```
Rvector6 StateConversionUtil::AltEquinoctialToEquinoctial (
 const Rvector6 & Altequinoctial) [static]
```

Converts from AlternateEquinoctial to Equinoctial.

#### Parameters

|                  |                                             |
|------------------|---------------------------------------------|
| <Altequinoctial> | Alternate Equinoctial state                 |
| <mu>             | Gravitational constant for the central body |

#### Returns

Spacecraft orbit state converted from Equinoctial to AlternateEquinoctial

### 8.62.3.2 BrouwerMeanLongToCartesian()

```
Rvector6 StateConversionUtil::BrouwerMeanLongToCartesian (
 Real mu,
 const Rvector6 & blms) [static]
```

Converts from Brouwer-Lyddane Mean Elements to Cartesian.

#### Parameters

|        |                               |
|--------|-------------------------------|
| <blms> | Brouwer-Lyddane Mean Elements |
|--------|-------------------------------|

#### Returns

Spacecraft orbit state converted from BrouwerMeanLong to Cartesian

#### 8.62.3.3 BrouwerMeanLongToOsculatingElements()

```
Rvector6 StateConversionUtil::BrouwerMeanLongToOsculatingElements (
 Real mu,
 const Rvector6 & blml) [static]
```

Converts from BrouwerMeanLong to Osculating Keplerian Elements.

#### Parameters

|        |                                                            |
|--------|------------------------------------------------------------|
| <blml> | Brouwer-Lyddane Mean Elements (Long and long period terms) |
|--------|------------------------------------------------------------|

#### Returns

Spacecraft orbit state converted from Keplerian to BrouwerMeanLong

#### 8.62.3.4 BrouwerMeanShortToCartesian()

```
Rvector6 StateConversionUtil::BrouwerMeanShortToCartesian (
 Real mu,
 const Rvector6 & blms) [static]
```

Converts from Brouwer-Lyddane Mean Elements (short period terms only) to Cartesian.

#### Parameters

|        |                                                         |
|--------|---------------------------------------------------------|
| <blms> | Brouwer-Lyddane Mean Elements (short period terms only) |
|--------|---------------------------------------------------------|

#### Returns

Spacecraft orbit state converted from BrouwerMeanShort to Cartesian

### 8.62.3.5 BrouwerMeanShortToOsculatingElements()

```
Rvector6 StateConversionUtil::BrouwerMeanShortToOsculatingElements (
 Real mu,
 const Rvector6 & blms) [static]
```

Converts from Brouwer-Lyddane Mean Elements (short period terms only) to Osculating Keplerian Elements.

#### Parameters

|        |                                                         |
|--------|---------------------------------------------------------|
| <blms> | Brouwer-Lyddane Mean Elements (short period terms only) |
|--------|---------------------------------------------------------|

#### Returns

Spacecraft orbit state converted from BrouwerMeanShort to Keplerian

### 8.62.3.6 CalculateEccentricAnomaly()

```
Real StateConversionUtil::CalculateEccentricAnomaly (
 Real e,
 Real M) [static]
```

This function is used to calculate eccentric anomaly E based on values of eccentricity and mean anomaly

#### Parameters

|          |                                               |
|----------|-----------------------------------------------|
| <i>e</i> | eccentricity                                  |
| <i>M</i> | mean anomaly return value of ecentric anomaly |

### 8.62.3.7 CartesianToAngularMomentum()

```
Rvector6 StateConversionUtil::CartesianToAngularMomentum (
 Real mu,
 const Rvector3 & pos,
 const Rvector3 & vel) [static]
```

CartesianTo angular momentum, its magnitude, and other related data.

#### Parameters

|       |                                            |
|-------|--------------------------------------------|
| <mu>  | gravitational constant of the central body |
| <pos> | input position vector                      |
| <vel> | input velocity vector                      |

**Returns**

[0] Angular momentum x unit vector [1] Angular momentum y unit vector [2] Angular momentum z unit vector  
 [3] Angular momentum magnitude [4] velocity magnitude squared [5] Orbit parameter

**8.62.3.8 CartesianToAOP()**

```
Real StateConversionUtil::CartesianToAOP (
 Real mu,
 const Rvector3 & pos,
 const Rvector3 & vel,
 bool inRadian = false) [static]
```

**8.62.3.9 CartesianToBrouwerMeanLong()**

```
Rvector6 StateConversionUtil::CartesianToBrouwerMeanLong (
 Real mu,
 const Rvector6 & cartesian) [static]
```

Converts from Brouwer-Lyddane Mean Elements (short and long period terms) to Cartesian.

**Parameters**

|             |                 |
|-------------|-----------------|
| <cartesian> | Cartesian state |
|-------------|-----------------|

**Returns**

Spacecraft orbit state converted from Cartesian to BrouwerMeanLong

**8.62.3.10 CartesianToBrouwerMeanShort()**

```
Rvector6 StateConversionUtil::CartesianToBrouwerMeanShort (
 Real mu,
 const Rvector6 & cartesian) [static]
```

Converts from Outgoing Aymptote to Cartesian.

**Parameters**

|             |                                             |
|-------------|---------------------------------------------|
| <mu>        | Gravitational constant for the central body |
| <cartesian> | Cartesian state                             |

**Returns**

Spacecraft orbit state converted from Cartesian to BrouwerMeanShort

**8.62.3.11 CartesianToDirOfLineOfNode()**

```
Rvector3 StateConversionUtil::CartesianToDirOfLineOfNode (
 const Rvector3 & pos,
 const Rvector3 & vel) [static]
```

**8.62.3.12 CartesianToEA()**

```
Real StateConversionUtil::CartesianToEA (
 Real mu,
 const Rvector3 & pos,
 const Rvector3 & vel,
 bool inRadian = false) [static]
```

**8.62.3.13 CartesianToECC()**

```
Real StateConversionUtil::CartesianToECC (
 Real mu,
 const Rvector3 & pos,
 const Rvector3 & vel) [static]
```

**8.62.3.14 CartesianToEccVector()**

```
Rvector3 StateConversionUtil::CartesianToEccVector (
 Real mu,
 const Rvector3 & pos,
 const Rvector3 & vel) [static]
```

**8.62.3.15 CartesianToEquinoctial()**

```
Rvector6 StateConversionUtil::CartesianToEquinoctial (
 const Rvector6 & cartesian,
 const Real & mu) [static]
```

Converts from Cartesian to Equinoctial.



## Parameters

|             |                                             |
|-------------|---------------------------------------------|
| <cartesian> | Cartesian state                             |
| <mu>        | Gravitational constant for the central body |

## Returns

Spacecraft orbit state converted from Cartesian to Equinoctial

## 8.62.3.16 CartesianToHA()

```
Real StateConversionUtil::CartesianToHA (
 Real mu,
 const Rvector3 & pos,
 const Rvector3 & vel,
 bool inRadian = false) [static]
```

## 8.62.3.17 CartesianToINC()

```
Real StateConversionUtil::CartesianToINC (
 Real mu,
 const Rvector3 & pos,
 const Rvector3 & vel,
 bool inRadian = false) [static]
```

## 8.62.3.18 CartesianToIncomingAsymptote()

```
Rvector6 StateConversionUtil::CartesianToIncomingAsymptote (
 Real mu,
 const Rvector6 & cartesian) [static]
```

Converts from Cartesian to Incoming Asymptote.

## Parameters

|             |                                             |
|-------------|---------------------------------------------|
| <mu>        | Gravitational constant for the central body |
| <cartesian> | Cartesian state                             |

## Returns

Spacecraft orbit state converted from Cartesian to Incoming Asymptote

### 8.62.3.19 CartesianToKeplerian() [1/5]

```
Rvector6 StateConversionUtil::CartesianToKeplerian (
 Real mu,
 const Rvector3 & pos,
 const Rvector3 & vel,
 AnomalyType anomalyType) [static]
```

Converts from Cartesian to Keplerian.

#### Parameters

|               |                                             |
|---------------|---------------------------------------------|
| <mu>          | Gravitational constant for the central body |
| <pos>         | Cartesian position                          |
| <vel>         | Cartesian velocity                          |
| <anomalyType> | Anomaly type                                |

#### Returns

Spacecraft orbit state converted from Cartesian to Keplerian

### 8.62.3.20 CartesianToKeplerian() [2/5]

```
Rvector6 StateConversionUtil::CartesianToKeplerian (
 Real mu,
 const Rvector3 & pos,
 const Rvector3 & vel,
 const std::string & anomalyType = "TA") [static]
```

Converts from Cartesian to Keplerian.

#### Parameters

|               |                                             |
|---------------|---------------------------------------------|
| <mu>          | Gravitational constant for the central body |
| <pos>         | Cartesian position                          |
| <vel>         | Cartesian velocity                          |
| <anomalyType> | Anomaly type                                |

#### Returns

Spacecraft orbit state converted from Cartesian to Keplerian

### 8.62.3.21 CartesianToKeplerian() [3/5]

```
Rvector6 StateConversionUtil::CartesianToKeplerian (
 Real mu,
```

```
const Rvector6 & state,
AnomalyType anomalyType) [static]
```

Converts from Cartesian to Keplerian.

#### Parameters

|               |                                             |
|---------------|---------------------------------------------|
| <mu>          | Gravitational constant for the central body |
| <state>       | Cartesian state                             |
| <anomalyType> | Anomaly type                                |

#### Returns

Spacecraft orbit state converted from Cartesian to Keplerian

#### 8.62.3.22 CartesianToKeplerian() [4/5]

```
Rvector6 StateConversionUtil::CartesianToKeplerian (
 Real mu,
 const Rvector6 & state,
 const std::string & anomalyType = "TA") [static]
```

Converts from Cartesian to Keplerian.

#### Parameters

|               |                                             |
|---------------|---------------------------------------------|
| <mu>          | Gravitational constant for the central body |
| <state>       | Cartesian state                             |
| <anomalyType> | Anomaly type                                |

#### Returns

Spacecraft orbit state converted from Cartesian to Keplerian

#### 8.62.3.23 CartesianToKeplerian() [5/5]

```
Rvector6 StateConversionUtil::CartesianToKeplerian (
 Real mu,
 const Rvector6 & state,
 Real * ma) [static]
```

Converts from Cartesian to Keplerian.

#### Parameters

|         |                                             |
|---------|---------------------------------------------|
| <mu>    | Gravitational constant for the central body |
| <state> | Cartesian state                             |
| <ma>    | Mean Anomaly                                |

**Returns**

Spacecraft orbit state converted from Cartesian to Keplerian

**8.62.3.24 CartesianToKeplerianDerivativeConversion()**

```
Rmatrix66 StateConversionUtil::CartesianToKeplerianDerivativeConversion (
 Real mu,
 const Rvector6 & cartesianState) [static]
```

This function is used to calculate Cartesian to Keplerian derivative state conversion matrix. Only apply for Keplerian state presented in mean anomaly only. (GTDS MathSpec Equation 3-229)

**Parameters**

|                       |                                                |
|-----------------------|------------------------------------------------|
| <i>mu</i>             | mu value of primary body                       |
| <i>cartesianState</i> | state presented in Cartesian coordiante system |

return derivative state conversion matrix  $[dX/dK]$  where X is Cartesian state and K is Keplerian state Use finite difference method for Hyperbolic and Parabolic case.

**8.62.3.25 CartesianToKeplerianDerivativeConversion\_FiniteDiff()**

```
Rmatrix66 StateConversionUtil::CartesianToKeplerianDerivativeConversion_FiniteDiff (
 Real mu,
 const Rvector6 & cartesianState) [static]
```

This function is used to calculate derivative conversion matrix  $[dX/dK]$  where X is Cartesian state and K is Keplerian state using finite difference method

**Parameters**

|                       |                                                |
|-----------------------|------------------------------------------------|
| <i>mu</i>             | mu value of primary body                       |
| <i>cartesianState</i> | state presented in Cartesian coordiante system |

return derivative state conversion matrix  $[dX/dK]$  where X is Cartesian state and K is Keplerian state

**8.62.3.26 CartesianToKeplerianDerivativeConversionWithKeplInput()**

```
Rmatrix66 StateConversionUtil::CartesianToKeplerianDerivativeConversionWithKeplInput (
 Real mu,
 const Rvector6 & keplerState) [static]
```

This function is used to calculate  $[dX/dK]$  derivative conversion matrix for a given Keplerian state. Only apply for Keplerian state presented in mean anomaly

## Parameters

|                    |                                                |
|--------------------|------------------------------------------------|
| <i>mu</i>          | mu value of primary body                       |
| <i>keplerState</i> | state presented in Keplerian coordiante system |

return derivative state conversion matrix

## 8.62.3.27 CartesianToKeplerianDerivativeConversionWithKeplInput\_FiniteDiff()

```
Rmatrix66 StateConversionUtil::CartesianToKeplerianDerivativeConversionWithKeplInput_FiniteDiff (
 Real mu,
 const Rvector6 & keplerianState) [static]
```

This function is used to calculate derivative conversion matrix  $[dX/dK]$  where X is Cartesian state and K is Keplerian state using finite difference method. Keplerian anomaly is in MA form.

## Parameters

|                       |                                                                        |
|-----------------------|------------------------------------------------------------------------|
| <i>mu</i>             | mu value of primary body                                               |
| <i>keplerianState</i> | state presented in Keplerian coordiante system with anomaly in MA form |

return derivative state conversion matrix  $[dX/dK]$  where X is Cartesian state and K is Keplerian state. Keplerian anomaly element is in MA form.

## 8.62.3.28 CartesianToMA()

```
Real StateConversionUtil::CartesianToMA (
 Real mu,
 const Rvector3 & pos,
 const Rvector3 & vel,
 bool inRadian = false) [static]
```

## 8.62.3.29 CartesianToModEquinoctial()

```
Rvector6 StateConversionUtil::CartesianToModEquinoctial (
 const Rvector6 & cartesian,
 const Real & mu) [static]
```

Converts from Cartesian to ModifiedEquinoctial.

## Parameters

|             |                                             |
|-------------|---------------------------------------------|
| <cartesian> | Cartesian state                             |
| <mu>        | Gravitational constant for the central body |

**Returns**

Spacecraft orbit state converted from Cartesian to ModifiedEquinoctial

**8.62.3.30 CartesianToOutgoingAsymptote()**

```
Rvector6 StateConversionUtil::CartesianToOutgoingAsymptote (
 Real mu,
 const Rvector6 & cartesian) [static]
```

Converts from Cartesian to Outgoing Asymptote.

**Parameters**

|             |                                             |
|-------------|---------------------------------------------|
| <mu>        | Gravitational constant for the central body |
| <cartesian> | Cartesian state                             |

**Returns**

Spacecraft orbit state converted from Cartesian to Outgoing Asymptote

**8.62.3.31 CartesianToPlanetodetic()**

```
Rvector6 StateConversionUtil::CartesianToPlanetodetic (
 const Rvector6 & cartesian,
 Real flattening,
 Real eqRadius) [static]
```

Converts from Planetocentric to Cartesian.

**Parameters**

|              |                                             |
|--------------|---------------------------------------------|
| <cartesian>  | Cartesian state                             |
| <flattening> | flattening coefficient for the central body |
| <eqRadius>   | equatorial radius for the central body      |

**Returns**

Spacecraft orbit state converted from Cartesian to Planetodetic

**8.62.3.32 CartesianToRAAN()**

```
Real StateConversionUtil::CartesianToRAAN (
 Real mu,
```

```
const Rvector3 & pos,
const Rvector3 & vel,
bool inRadian = false) [static]
```

#### 8.62.3.33 CartesianToSMA()

```
Real StateConversionUtil::CartesianToSMA (
 Real mu,
 const Rvector3 & pos,
 const Rvector3 & vel) [static]
```

#### 8.62.3.34 CartesianToSphericalAZFPA()

```
Rvector6 StateConversionUtil::CartesianToSphericalAZFPA (
 const Rvector6 & cartesian) [static]
```

Converts from Cartesian to SphericalAZFPA.

##### Parameters

|             |                 |
|-------------|-----------------|
| <cartesian> | Cartesian state |
|-------------|-----------------|

##### Returns

Spacecraft orbit state converted from Cartesian to SphericalAZFPA

#### 8.62.3.35 CartesianToSphericalRADEC()

```
Rvector6 StateConversionUtil::CartesianToSphericalRADEC (
 const Rvector6 & cartesian) [static]
```

Converts from Cartesian to SphericalRADEC.

##### Parameters

|             |                 |
|-------------|-----------------|
| <cartesian> | Cartesian state |
|-------------|-----------------|

##### Returns

Spacecraft orbit state converted from Cartesian to SphericalRADEC

## 8.62.3.36 CartesianToTA()

```
Real StateConversionUtil::CartesianToTA (
 Real mu,
 const Rvector3 & pos,
 const Rvector3 & vel,
 bool inRadian = false) [static]
```

## 8.62.3.37 Convert() [1/2]

```
Rvector6 StateConversionUtil::Convert (
 const Real * state,
 const std::string & fromType,
 const std::string & toType,
 Real mu = GmatSolarSystemDefaults::PLANET_MU[GmatSolarSystemDefaults::EARTH],
 Real flattening = GmatSolarSystemDefaults::PLANET_FLATTENING[GmatSolarSystem↵
Defaults::EARTH],
 Real eqRadius = GmatSolarSystemDefaults::PLANET_EQUATORIAL_RADIUS[GmatSolar↵
SystemDefaults::EARTH],
 const std::string & anomalyType = "TA") [static]
```

Converts from fromType to toType.

## Parameters

|               |                                                |
|---------------|------------------------------------------------|
| <state>       | state to convert                               |
| <fromType>    | state type to convert from                     |
| <toType>      | state type to convert to                       |
| <mu>          | gravitational constant for the central body    |
| <flattening>  | flattening coefficient for the central body    |
| <eqRadius>    | equatorial radius for the central body         |
| <anomalyType> | anomaly type string if toType is Mod/Keplerian |

## Returns

Converted states from the specific element type

## 8.62.3.38 Convert() [2/2]

```
Rvector6 StateConversionUtil::Convert (
 const Rvector6 & state,
 const std::string & fromType,
 const std::string & toType,
 Real mu = GmatSolarSystemDefaults::PLANET_MU[GmatSolarSystemDefaults::EARTH],
 Real flattening = GmatSolarSystemDefaults::PLANET_FLATTENING[GmatSolarSystem↵
Defaults::EARTH],
```



```
Real eqRadius = GmatSolarSystemDefaults::PLANET_EQUATORIAL_RADIUS[GmatSolarSystemDefaults::EARTH],
 const std::string & anomalyType = "TA") [static]
```

Converts state from fromType to toType.

## Parameters

|               |                                                |
|---------------|------------------------------------------------|
| <state>       | state to convert                               |
| <fromType>    | state type to convert from                     |
| <toType>      | state type to convert to                       |
| <mu>          | gravitational constant for the central body    |
| <flattening>  | flattening coefficient for the central body    |
| <eqRadius>    | equatorial radius for the central body         |
| <anomalyType> | anomaly type string if toType is Mod/Keplerian |

## Returns

Converted states from the specific element type

## 8.62.3.39 ConvertFromAltEquinoctial()

```
Rvector6 StateConversionUtil::ConvertFromAltEquinoctial (
 const std::string & toType,
 const Rvector6 & state,
 Real mu = EARTH_MU,
 const std::string & anomalyType = "TA",
 Real flattening = EARTH_FLATTENING,
 Real eqRadius = EARTH_EQ_RADIUS) [static]
```

## 8.62.3.40 ConvertFromBrouwerMeanLong()

```
Rvector6 StateConversionUtil::ConvertFromBrouwerMeanLong (
 const std::string & toType,
 const Rvector6 & state,
 Real mu = EARTH_MU,
 const std::string & anomalyType = "TA",
 Real flattening = EARTH_FLATTENING,
 Real eqRadius = EARTH_EQ_RADIUS) [static]
```

## 8.62.3.41 ConvertFromBrouwerMeanShort()

```
Rvector6 StateConversionUtil::ConvertFromBrouwerMeanShort (
 const std::string & toType,
 const Rvector6 & state,
 Real mu = EARTH_MU,
 const std::string & anomalyType = "TA",
 Real flattening = EARTH_FLATTENING,
 Real eqRadius = EARTH_EQ_RADIUS) [static]
```

#### 8.62.3.42 ConvertFromCartesian()

```
Rvector6 StateConversionUtil::ConvertFromCartesian (
 const std::string & toType,
 const Rvector6 & state,
 Real mu = EARTH_MU,
 const std::string & anomalyType = "TA",
 Real flattening = EARTH_FLATTENING,
 Real eqRadius = EARTH_EQ_RADIUS) [static]
```

#### 8.62.3.43 ConvertFromDelaunay()

```
Rvector6 StateConversionUtil::ConvertFromDelaunay (
 const std::string & toType,
 const Rvector6 & state,
 Real mu = EARTH_MU,
 const std::string & anomalyType = "TA",
 Real flattening = EARTH_FLATTENING,
 Real eqRadius = EARTH_EQ_RADIUS) [static]
```

#### 8.62.3.44 ConvertFromEquinoctial()

```
Rvector6 StateConversionUtil::ConvertFromEquinoctial (
 const std::string & toType,
 const Rvector6 & state,
 Real mu = EARTH_MU,
 const std::string & anomalyType = "TA",
 Real flattening = EARTH_FLATTENING,
 Real eqRadius = EARTH_EQ_RADIUS) [static]
```

#### 8.62.3.45 ConvertFromIncomingAsymptote()

```
Rvector6 StateConversionUtil::ConvertFromIncomingAsymptote (
 const std::string & toType,
 const Rvector6 & state,
 Real mu = EARTH_MU,
 const std::string & anomalyType = "TA",
 Real flattening = EARTH_FLATTENING,
 Real eqRadius = EARTH_EQ_RADIUS) [static]
```

#### 8.62.3.46 ConvertFromKeplerian()

```
Rvector6 StateConversionUtil::ConvertFromKeplerian (
 const std::string & toType,
 const Rvector6 & state,
 Real mu = EARTH_MU,
 const std::string & anomalyType = "TA",
 Real flattening = EARTH_FLATTENING,
 Real eqRadius = EARTH_EQ_RADIUS) [static]
```

#### 8.62.3.47 ConvertFromModEquinoctial()

```
Rvector6 StateConversionUtil::ConvertFromModEquinoctial (
 const std::string & toType,
 const Rvector6 & state,
 Real mu = EARTH_MU,
 const std::string & anomalyType = "TA",
 Real flattening = EARTH_FLATTENING,
 Real eqRadius = EARTH_EQ_RADIUS) [static]
```

#### 8.62.3.48 ConvertFromModKeplerian()

```
Rvector6 StateConversionUtil::ConvertFromModKeplerian (
 const std::string & toType,
 const Rvector6 & state,
 Real mu = EARTH_MU,
 const std::string & anomalyType = "TA",
 Real flattening = EARTH_FLATTENING,
 Real eqRadius = EARTH_EQ_RADIUS) [static]
```

#### 8.62.3.49 ConvertFromOutgoingAsymptote()

```
Rvector6 StateConversionUtil::ConvertFromOutgoingAsymptote (
 const std::string & toType,
 const Rvector6 & state,
 Real mu = EARTH_MU,
 const std::string & anomalyType = "TA",
 Real flattening = EARTH_FLATTENING,
 Real eqRadius = EARTH_EQ_RADIUS) [static]
```

**8.62.3.50 ConvertFromPlanetodetic()**

```
Rvector6 StateConversionUtil::ConvertFromPlanetodetic (
 const std::string & toType,
 const Rvector6 & state,
 Real mu = EARTH_MU,
 const std::string & anomalyType = "TA",
 Real flattening = EARTH_FLATTENING,
 Real eqRadius = EARTH_EQ_RADIUS) [static]
```

**8.62.3.51 ConvertFromSphericalAZFPA()**

```
Rvector6 StateConversionUtil::ConvertFromSphericalAZFPA (
 const std::string & toType,
 const Rvector6 & state,
 Real mu = EARTH_MU,
 const std::string & anomalyType = "TA",
 Real flattening = EARTH_FLATTENING,
 Real eqRadius = EARTH_EQ_RADIUS) [static]
```

**8.62.3.52 ConvertFromSphericalRADEC()**

```
Rvector6 StateConversionUtil::ConvertFromSphericalRADEC (
 const std::string & toType,
 const Rvector6 & state,
 Real mu = EARTH_MU,
 const std::string & anomalyType = "TA",
 Real flattening = EARTH_FLATTENING,
 Real eqRadius = EARTH_EQ_RADIUS) [static]
```

**8.62.3.53 ConvertFromTrueAnomaly() [1/2]**

```
Real StateConversionUtil::ConvertFromTrueAnomaly (
 const std::string & toType,
 Real taRadians,
 Real ecc,
 bool modBy2Pi = false) [static]
```

**8.62.3.54 ConvertFromTrueAnomaly() [2/2]**

```
Real StateConversionUtil::ConvertFromTrueAnomaly (
 AnomalyType toType,
 Real taRadians,
 Real ecc,
 bool modBy2Pi = false) [static]
```

**8.62.3.55 ConvertToTrueAnomaly()** [1/2]

```
Real StateConversionUtil::ConvertToTrueAnomaly (
 const std::string & fromType,
 Real taRadians,
 Real ecc,
 bool modBy2Pi = false) [static]
```

**8.62.3.56 ConvertToTrueAnomaly()** [2/2]

```
Real StateConversionUtil::ConvertToTrueAnomaly (
 AnomalyType fromType,
 Real taRadians,
 Real ecc,
 bool modBy2Pi = false) [static]
```

**8.62.3.57 DelaunayToKeplerian()**

```
Rvector6 StateConversionUtil::DelaunayToKeplerian (
 const Rvector6 & delaunay,
 const Real & mu) [static]
```

Converts from Delaunay to Keplerian

**Parameters**

|                      |                                             |
|----------------------|---------------------------------------------|
| <keplerian> Delaunay | state                                       |
| <mu>                 | Gravitational constant for the central body |

**Returns**

Spacecraft orbit state converted from Delaunay to Keplerian

**8.62.3.58 EccentricToTrueAnomaly()**

```
Real StateConversionUtil::EccentricToTrueAnomaly (
 Real eaRadians,
 Real ecc,
 bool modBy2Pi = false) [static]
```

#### 8.62.3.59 EquinoctialToAltEquinoctial()

```
Rvector6 StateConversionUtil::EquinoctialToAltEquinoctial (
 const Rvector6 & equinoctial) [static]
```

Converts from Equinoctial to AlternateEquinoctial.

**Parameters**

|                        |                                             |
|------------------------|---------------------------------------------|
| < <i>equinoctial</i> > | Equinoctial state                           |
| < <i>mu</i> >          | Gravitational constant for the central body |

**Returns**

Spacecraft orbit state converted from Equinoctial to AlternateEquinoctial

**8.62.3.60 EquinoctialToCartesian()**

```
Rvector6 StateConversionUtil::EquinoctialToCartesian (
 const Rvector6 & equinoctial,
 const Real & mu) [static]
```

Converts from Equinoctial to Cartesian.

**Parameters**

|                        |                                             |
|------------------------|---------------------------------------------|
| < <i>equinoctial</i> > | Equinoctial state                           |
| < <i>mu</i> >          | Gravitational constant for the central body |

**Returns**

Spacecraft orbit state converted from Equinoctial to Cartesian

**8.62.3.61 GetAnomalyLongText()**

```
std::string StateConversionUtil::GetAnomalyLongText (
 const std::string & type) [static]
```

Returns the long anomaly name for the specified anomaly type

**Parameters**

|                        |              |
|------------------------|--------------|
| < <i>anomalyType</i> > | anomaly type |
|------------------------|--------------|

**Returns**

corresponding long anomaly name



#### 8.62.3.62 GetAnomalyShortText()

```
std::string StateConversionUtil::GetAnomalyShortText (
 const std::string & type) [static]
```

Returns the short anomaly name for the specified anomaly type

##### Parameters

|               |              |
|---------------|--------------|
| <anomalyType> | anomaly type |
|---------------|--------------|

##### Returns

corresponding short anomaly name

#### 8.62.3.63 GetAnomalyType()

```
StateConversionUtil::AnomalyType StateConversionUtil::GetAnomalyType (
 const std::string & typeStr) [static]
```

Returns the AnomalyType corresponding to the specified string

##### Parameters

|           |                     |
|-----------|---------------------|
| <typeStr> | Anomaly type string |
|-----------|---------------------|

##### Returns

anomaly type

#### 8.62.3.64 GetLongTypeNameList()

```
const std::string * StateConversionUtil::GetLongTypeNameList () [static]
```

Returns list of long anomaly names

##### Returns

list of long anomaly names

**8.62.3.65    GetStateTypeList()**

```
const std::string * StateConversionUtil::GetStateTypeList () [static]
```

Returns the list of state types

**Returns**

list of valid state types

**8.62.3.66    GetTypeCount()**

```
static Integer StateConversionUtil::GetTypeCount () [inline], [static]
```

**8.62.3.67    HyperbolicToTrueAnomaly()**

```
Real StateConversionUtil::HyperbolicToTrueAnomaly (
 Real haRadians,
 Real ecc,
 bool modBy2Pi = false) [static]
```

**8.62.3.68    IncomingAsymptoteToCartesian()**

```
Rvector6 StateConversionUtil::IncomingAsymptoteToCartesian (
 Real mu,
 const Rvector6 & inasym) [static]
```

Converts from Incoming Aymptote to Cartesian.

**Parameters**

|          |                                             |
|----------|---------------------------------------------|
| <mu>     | Gravitational constant for the central body |
| <inasym> | Incoming Asymptote state                    |

**Returns**

Spacecraft orbit state converted from Incoming Aymptote to Cartesian

**8.62.3.69 IsRvValid()**

```
bool StateConversionUtil::IsRvValid (
 Real * r,
 Real * v) [static]
```

Returns flag indicating whether or not the input position/velocity is valid

**Parameters**

|              |          |
|--------------|----------|
| < <i>r</i> > | Position |
| < <i>v</i> > | Velocity |

**Returns**

flag indicating whether or not the input position/velocity is valid

**8.62.3.70 IsValidAnomalyType()**

```
bool StateConversionUtil::IsValidAnomalyType (
 const std::string & anomType) [static]
```

Returns flag indicating whether or not the input string is a valid anomaly type

**Parameters**

|                    |                     |
|--------------------|---------------------|
| < <i>typeStr</i> > | Anomaly type string |
|--------------------|---------------------|

**Returns**

flag indicating whether or not the input string is a valid anomaly type

**8.62.3.71 KeplerianToCartesian()** [1/2]

```
Rvector6 StateConversionUtil::KeplerianToCartesian (
 Real mu,
 const Rvector6 & state,
 AnomalyType anomalyType) [static]
```

Converts from Keplerian to Cartesian.

**Parameters**

|                        |                                             |
|------------------------|---------------------------------------------|
| < <i>mu</i> >          | Gravitational constant for the central body |
| < <i>state</i> >       | Keplerian state                             |
| < <i>anomalyType</i> > | Anomaly Type                                |

**Returns**

Spacecraft orbit state converted from Keplerian to Cartesian

**8.62.3.72 KeplerianToCartesian()** [2/2]

```
Rvector6 StateConversionUtil::KeplerianToCartesian (
 Real mu,
 const Rvector6 & state,
 const std::string & anomalyType = "TA") [static]
```

Converts from Keplerian to Cartesian.

**Parameters**

|               |                                             |
|---------------|---------------------------------------------|
| <mu>          | Gravitational constant for the central body |
| <state>       | Keplerian state                             |
| <anomalyType> | Anomaly Type                                |

**Returns**

Spacecraft orbit state converted from Keplerian to Cartesian

**8.62.3.73 KeplerianToDelaunay()**

```
Rvector6 StateConversionUtil::KeplerianToDelaunay (
 const Rvector6 & keplerian,
 const Real & mu) [static]
```

Converts from Keplerian to Delaunay

**Parameters**

|                       |                                             |
|-----------------------|---------------------------------------------|
| <keplerian> Keplerian | state                                       |
| <mu>                  | Gravitational constant for the central body |

**Returns**

Spacecraft orbit state converted from Keplerian to Delaunay

**8.62.3.74 KeplerianToModKeplerian()**

```
Rvector6 StateConversionUtil::KeplerianToModKeplerian (
 const Rvector6 & keplerian) [static]
```

Converts from Keplerian to Modified Keplerian.

#### Parameters

|                      |                 |
|----------------------|-----------------|
| < <i>keplerian</i> > | Keplerian state |
|----------------------|-----------------|

#### Returns

Spacecraft orbit state converted from Keplerian to Modified Keplerian

#### 8.62.3.75 MeanToTrueAnomaly()

```
Real StateConversionUtil::MeanToTrueAnomaly (
 Real maRadians,
 Real ecc,
 Real tol = 1.0e-08) [static]
```

#### 8.62.3.76 ModEquinoctialToCartesian()

```
Rvector6 StateConversionUtil::ModEquinoctialToCartesian (
 const Rvector6 & modequinoctial,
 const Real & mu) [static]
```

Converts from ModifiedEquinoctial to Cartesian.

#### Parameters

|                           |                                             |
|---------------------------|---------------------------------------------|
| < <i>modequinoctial</i> > | Modified Equinoctial state                  |
| < <i>mu</i> >             | Gravitational constant for the central body |

#### Returns

Spacecraft orbit state converted from Modified Equinoctial to Cartesian

#### 8.62.3.77 ModKeplerianToKeplerian()

```
Rvector6 StateConversionUtil::ModKeplerianToKeplerian (
 const Rvector6 & modKeplerian) [static]
```

Converts from Modified Keplerian to Keplerian.

**Parameters**

|                                   |                          |
|-----------------------------------|--------------------------|
| <code>&lt;modKeplerian&gt;</code> | Modified Keplerian state |
|-----------------------------------|--------------------------|

**Returns**

Spacecraft orbit state converted from Modified Keplerian to Keplerian

**8.62.3.78 OutgoingAsymptoteToCartesian()**

```
Rvector6 StateConversionUtil::OutgoingAsymptoteToCartesian (
 Real mu,
 const Rvector6 & outasym) [static]
```

Converts from Outgoing Aymptote to Cartesian.

**Parameters**

|                              |                                             |
|------------------------------|---------------------------------------------|
| <code>&lt;mu&gt;</code>      | Gravitational constant for the central body |
| <code>&lt;outasym&gt;</code> | Outgoing Asymptote state                    |

**Returns**

Spacecraft orbit state converted from Outgoing Aymptote to Cartesian

**8.62.3.79 PlanetodeticToCartesian()**

```
Rvector6 StateConversionUtil::PlanetodeticToCartesian (
 const Rvector6 & planetodetic,
 Real flattening,
 Real eqRadius) [static]
```

Converts from Planetodetic to Cartesian.

**Parameters**

|                                   |                                             |
|-----------------------------------|---------------------------------------------|
| <code>&lt;planetodetic&gt;</code> | Planetodetic state                          |
| <code>&lt;flattening&gt;</code>   | flattening coefficient for the central body |
| <code>&lt;eqRadius&gt;</code>     | equatorial radius for the central body      |

**Returns**

Spacecraft orbit state converted from Planetodetic to Cartesian

#### 8.62.3.80 RequiresCelestialBodyOrigin()

```
bool StateConversionUtil::RequiresCelestialBodyOrigin (
 const std::string & type) [static]
```

Returns a flag indicating whether or not the specified state type requires a celestial body origin

##### Parameters

|        |            |
|--------|------------|
| <type> | State type |
|--------|------------|

##### Returns

flag indicating whether or not the specified state type requires a celestial body origin

#### 8.62.3.81 RequiresFixedCoordinateSystem()

```
bool StateConversionUtil::RequiresFixedCoordinateSystem (
 const std::string & type) [static]
```

Returns a flag indicating whether or not the specified state type requires a fixed coordinate system

##### Parameters

|        |            |
|--------|------------|
| <type> | State type |
|--------|------------|

##### Returns

flag indicating whether or not the specified state type requires a fixed coordinate system

#### 8.62.3.82 SphericalAZFPAToCartesian()

```
Rvector6 StateConversionUtil::SphericalAZFPAToCartesian (
 const Rvector6 & spherical) [static]
```

Converts from SphericalAZFPA to Cartesian.

##### Parameters

|             |                      |
|-------------|----------------------|
| <spherical> | SphericalAZFPA state |
|-------------|----------------------|

##### Returns

Spacecraft orbit state converted from SphericalAZFPA to Cartesian

### 8.62.3.83 SphericalRADECToCartesian()

```
Rvector6 StateConversionUtil::SphericalRADECToCartesian (
 const Rvector6 & spherical) [static]
```

Converts from SphericalRADEC to Cartesian.

#### Parameters

|                      |                      |
|----------------------|----------------------|
| < <i>spherical</i> > | SphericalRADEC state |
|----------------------|----------------------|

#### Returns

Spacecraft orbit state converted from SphericalRADEC to Cartesian

### 8.62.3.84 TrueToEccentricAnomaly()

```
Real StateConversionUtil::TrueToEccentricAnomaly (
 Real taRadians,
 Real ecc,
 bool modBy2Pi = false) [static]
```

### 8.62.3.85 TrueToHyperbolicAnomaly()

```
Real StateConversionUtil::TrueToHyperbolicAnomaly (
 Real taRadians,
 Real ecc,
 bool modBy2Pi = false) [static]
```

Converts true anomaly to hyperbolic anomaly.

#### Parameters

|                                        |                                                            |
|----------------------------------------|------------------------------------------------------------|
| <i>taRadians</i>                       | True anomaly in radians                                    |
| <i>ecc</i>                             | Eccentricity                                               |
| <i>mod<sub>↔</sub></i><br><i>By2Pi</i> | Flag specifying whether or not to mod the result by two pi |

#### Returns

Hyperbolic anomaly in radians, 0.0 if eccentricity is less than 1.0 + KEP\_TOL



## Exceptions

|                                         |                                                      |
|-----------------------------------------|------------------------------------------------------|
| <a href="#"><i>UtilityException</i></a> | is throw if "eccentricity * cos(trueAnomaly)" is -1. |
|-----------------------------------------|------------------------------------------------------|

## 8.62.3.86 TrueToMeanAnomaly()

```
Real StateConversionUtil::TrueToMeanAnomaly (
 Real taRadians,
 Real ecc,
 bool modBy2Pi = false) [static]
```

## 8.62.3.87 ValidateValue()

```
bool StateConversionUtil::ValidateValue (
 const std::string & label,
 Real value,
 const std::string & errorMsgFmt,
 Integer dataPrecision,
 const std::string & compareTo = "",
 Real compareValue = 0.0) [static]
```

Validates an input value for the specified orbit element. When requested, also validates the value against another input value, for coupled quantities.

## Parameters

|                |                                                    |
|----------------|----------------------------------------------------|
| <label>        | Orbit element label                                |
| <value>        | Input value                                        |
| <compareTo>    | Name of coupled orbit element to validate against  |
| <compareValue> | Value of coupled orbit element to validate against |

## Returns

true if value is allowed/valid; false otherwise

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

- GMATsrc/util/[StateConversionUtil.hpp](#)
- GMATsrc/util/[StateConversionUtil.cpp](#)

## 8.63 StringTokenizer Class Reference

```
#include <StringTokenizer.hpp>
```

## Public Member Functions

- [StringTokenizer](#) ()
- [StringTokenizer](#) (const std::string &str, const std::string &delim)
- [StringTokenizer](#) (const std::string &str, const std::string &delim, bool insertDelim)
- [~StringTokenizer](#) ()
- void [SetDelimiters](#) (const std::string &delim)
- std::string [GetDelimiters](#) ()
- [Integer](#) [CountTokens](#) () const
- std::string [GetToken](#) (const [Integer](#) loc) const
- const [StringArray](#) & [GetAllTokens](#) () const
- void [Set](#) (const std::string &str, const std::string &delim)
- void [Set](#) (const std::string &str, const std::string &delim, bool insertDelim)

### 8.63.1 Detailed Description

Definition of the [StringTokenizer](#) class base

### 8.63.2 Constructor & Destructor Documentation

#### 8.63.2.1 [StringTokenizer](#)() [1/3]

```
StringTokenizer::StringTokenizer ()
```

Definition of the [StringTokenizer](#) class base Creates default constructor.

#### 8.63.2.2 [StringTokenizer](#)() [2/3]

```
StringTokenizer::StringTokenizer (
 const std::string & str,
 const std::string & delim)
```

Creates constructor with parameters.

#### Parameters

|         |                  |
|---------|------------------|
| <str>   | Given String     |
| <delim> | Given delimiters |

#### 8.63.2.3 [StringTokenizer](#)() [3/3]

```
StringTokenizer::StringTokenizer (
 const std::string & str,
```

```
const std::string & delim,
bool insertDelim)
```

Creates constructor with parameters.

#### Parameters

|               |                                                                                             |
|---------------|---------------------------------------------------------------------------------------------|
| <str>         | Given String                                                                                |
| <delim>       | Given delimiters                                                                            |
| <insertDelim> | true if inserting delimiter back to tokenized string. it will insert if delimiter size is 1 |

#### Note

: I want to have constructor with [StringTokenizer](#)(const std::string &str, const std::string &delim, bool insertDelim = false) but this causing some numeric differences when routine test is running

#### 8.63.2.4 ~StringTokenizer()

```
StringTokenizer::~StringTokenizer ()
```

Destructor.

### 8.63.3 Member Function Documentation

#### 8.63.3.1 CountTokens()

```
Integer StringTokenizer::CountTokens () const
```

Get the number of tokens.

#### Returns

number of tokens.

#### 8.63.3.2 GetAllTokens()

```
const StringArray & StringTokenizer::GetAllTokens () const
```

Get all token string.

#### Returns

all token strings

#### 8.63.3.3 GetDelimiters()

```
std::string StringTokenizer::GetDelimiters () [inline]
```

#### 8.63.3.4 GetToken()

```
std::string StringTokenizer::GetToken (
 const Integer loc) const
```

Get the string from specifying the token number.

##### Parameters

|       |                        |
|-------|------------------------|
| <loc> | Specified token number |
|-------|------------------------|

##### Returns

return string in the specified token number

#### 8.63.3.5 Set() [1/2]

```
void StringTokenizer::Set (
 const std::string & str,
 const std::string & delim)
```

Set the string token.

##### Parameters

|         |                  |
|---------|------------------|
| <str>   | given string     |
| <delim> | given delimiters |

#### 8.63.3.6 Set() [2/2]

```
void StringTokenizer::Set (
 const std::string & str,
 const std::string & delim,
 bool insertDelim)
```

Set the string token.

## Parameters

|         |                  |
|---------|------------------|
| <str>   | given string     |
| <delim> | given delimiters |

## 8.63.3.7 SetDelimiters()

```
void StringTokenizer::SetDelimiters (
 const std::string & delim) [inline]
```

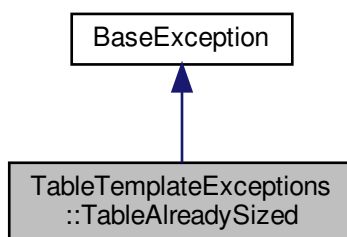
The documentation for this class was generated from the following files:

- GMATsrc/util/[StringTokenizer.hpp](#)
- GMATsrc/util/[StringTokenizer.cpp](#)

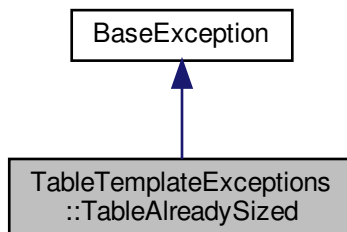
## 8.64 TableTemplateExceptions::TableAlreadySized Class Reference

```
#include <TableTemplate.hpp>
```

Inheritance diagram for TableTemplateExceptions::TableAlreadySized:



Collaboration diagram for TableTemplateExceptions::TableAlreadySized:



## Public Member Functions

- [TableAlreadySized](#) (const std::string &message="TableTemplate error : table already sized.\n")

## Additional Inherited Members

### 8.64.1 Constructor & Destructor Documentation

#### 8.64.1.1 TableAlreadySized()

```
TableTemplateExceptions::TableAlreadySized::TableAlreadySized(
 const std::string & message = "TableTemplate error : table already sized.\n")
[inline]
```

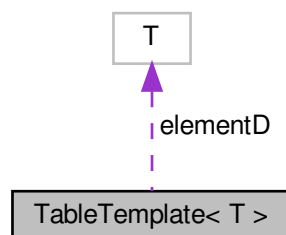
The documentation for this class was generated from the following file:

- GMATsrc/util/[TableTemplate.hpp](#)

### 8.65 TableTemplate< T > Class Template Reference

```
#include <TableTemplate.hpp>
```

Collaboration diagram for TableTemplate< T >:



## Public Member Functions

- [TableTemplate](#) ()
- [TableTemplate](#) ([Integer](#) r, [Integer](#) c)
- [TableTemplate](#) ([Integer](#) r, [Integer](#) c, const T \*array)
- [TableTemplate](#) (const [TableTemplate](#)< T > &table)
- virtual [~TableTemplate](#) ()
- T & [operator\(\)](#) ([Integer](#) r, [Integer](#) c)
- const T & [operator\(\)](#) ([Integer](#) r, [Integer](#) c) const
- [TableTemplate](#)< T > & [operator=](#) (const [TableTemplate](#)< T > &table)
- bool [operator==](#) (const [TableTemplate](#)< T > &table) const
- bool [operator!=](#) (const [TableTemplate](#)< T > &table) const
- virtual T [GetElement](#) ([Integer](#) r, [Integer](#) c) const
- virtual void [SetElement](#) ([Integer](#) r, [Integer](#) c, const T &value)
- virtual bool [IsSized](#) () const
- virtual void [SetSize](#) ([Integer](#) r, [Integer](#) c, bool zeroElements=true)
- virtual void [ChangeSize](#) (int r, int c, bool zeroElements)
- virtual void [GetSize](#) ([Integer](#) &r, [Integer](#) &c) const
- virtual [Integer](#) [GetNumColumns](#) () const
- virtual [Integer](#) [GetNumRows](#) () const
- const T \* [GetDataVector](#) ()

## Protected Member Functions

- void [init](#) ([Integer](#) r, [Integer](#) c)

## Protected Attributes

- T \* [elementD](#)
- [Integer](#) [rowsD](#)
- [Integer](#) [colsD](#)
- bool [isSizedD](#)

### 8.65.1 Constructor & Destructor Documentation

#### 8.65.1.1 [TableTemplate](#)() [ 1 / 4 ]

```
template<class T >
TableTemplate< T >::TableTemplate ()
```

Tables are stored in row major order. Assumptions about template parameter types: type has appropriate initializers (constructors and "=" operator) The exceptions are declared in a separate class because the current HP compiler cannot properly handle exceptions declared a template class and thrown in another template class.

Provides declarations for the [TableTemplate](#) template class, representing a 2-dimensional table of any type T (see assumptions about type T, below)

#### 8.65.1.2 TableTemplate() [2/4]

```
template<class T >
TableTemplate< T >::TableTemplate (
 Integer r,
 Integer c)
```

#### 8.65.1.3 TableTemplate() [3/4]

```
template<class T>
TableTemplate< T >::TableTemplate (
 Integer r,
 Integer c,
 const T * array)
```

#### 8.65.1.4 TableTemplate() [4/4]

```
template<class T>
TableTemplate< T >::TableTemplate (
 const TableTemplate< T > & table)
```

#### 8.65.1.5 ~TableTemplate()

```
template<class T >
TableTemplate< T >::~~TableTemplate () [virtual]
```

### 8.65.2 Member Function Documentation

#### 8.65.2.1 ChangeSize()

```
template<class T >
void TableTemplate< T >::ChangeSize (
 int r,
 int c,
 bool zeroElements) [virtual]
```



#### 8.65.2.2 GetDataVector()

```
template<class T>
const T* TableTemplate< T >::GetDataVector () [inline]
```

#### 8.65.2.3 GetElement()

```
template<class T >
T TableTemplate< T >::GetElement (
 Integer r,
 Integer c) const [virtual]
```

#### 8.65.2.4 GetNumColumns()

```
template<class T >
int TableTemplate< T >::GetNumColumns () const [virtual]
```

#### 8.65.2.5 GetNumRows()

```
template<class T >
int TableTemplate< T >::GetNumRows () const [virtual]
```

#### 8.65.2.6 GetSize()

```
template<class T >
void TableTemplate< T >::GetSize (
 Integer & r,
 Integer & c) const [virtual]
```

#### 8.65.2.7 init()

```
template<class T >
void TableTemplate< T >::init (
 Integer r,
 Integer c) [protected]
```

#### 8.65.2.8 IsSized()

```
template<class T >
bool TableTemplate< T >::IsSized () const [virtual]
```

#### 8.65.2.9 operator"!="()

```
template<class T>
bool TableTemplate< T >::operator!= (
 const TableTemplate< T > & table) const
```

#### 8.65.2.10 operator() [1/2]

```
template<class T >
T & TableTemplate< T >::operator() (
 Integer r,
 Integer c)
```

#### 8.65.2.11 operator() [2/2]

```
template<class T >
const T & TableTemplate< T >::operator() (
 Integer r,
 Integer c) const
```

#### 8.65.2.12 operator=()

```
template<class T>
TableTemplate< T > & TableTemplate< T >::operator= (
 const TableTemplate< T > & table)
```

#### 8.65.2.13 operator==()

```
template<class T>
bool TableTemplate< T >::operator== (
 const TableTemplate< T > & table) const
```

#### 8.65.2.14 SetElement()

```
template<class T>
void TableTemplate< T >::SetElement (
 Integer r,
 Integer c,
 const T & value) [virtual]
```

#### 8.65.2.15 SetSize()

```
template<class T >
void TableTemplate< T >::SetSize (
 Integer r,
 Integer c,
 bool zeroElements = true) [virtual]
```

### 8.65.3 Member Data Documentation

#### 8.65.3.1 colsD

```
template<class T>
Integer TableTemplate< T >::colsD [protected]
```

#### 8.65.3.2 elementD

```
template<class T>
T* TableTemplate< T >::elementD [protected]
```

#### 8.65.3.3 isSizedD

```
template<class T>
bool TableTemplate< T >::isSizedD [protected]
```

#### 8.65.3.4 rowsD

```
template<class T>
Integer TableTemplate< T >::rowsD [protected]
```

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

- GMATsrc/util/[TableTemplate.hpp](#)
- GMATsrc/util/[TableTemplate.cpp](#)

## 8.66 TableTemplateExceptions Class Reference

```
#include <TableTemplate.hpp>
```

### Classes

- class [DimensionError](#)
- class [IllegalSize](#)
- class [OutOfBounds](#)
- class [TableAlreadySized](#)
- class [UnsizeTable](#)

### 8.66.1 Detailed Description

Provides declarations for the [TableTemplate](#) template class, representing a 2-dimensional table of any type T (see assumptions about type T, below)

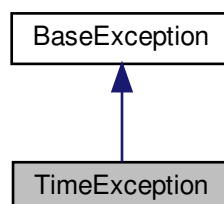
The documentation for this class was generated from the following file:

- GMATsrc/util/[TableTemplate.hpp](#)

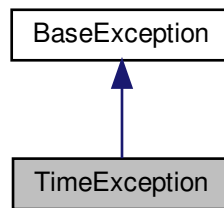
## 8.67 TimeException Class Reference

```
#include <UtilityException.hpp>
```

Inheritance diagram for TimeException:



Collaboration diagram for TimeException:



### Public Member Functions

- [TimeException](#) (const std::string &details="")

### Additional Inherited Members

#### 8.67.1 Constructor & Destructor Documentation

##### 8.67.1.1 TimeException()

```
TimeException::TimeException (
 const std::string & details = "") [inline]
```

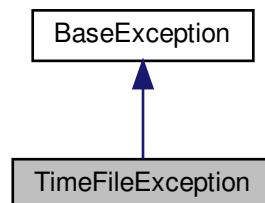
The documentation for this class was generated from the following file:

- GMATsrc/util/[UtilityException.hpp](#)

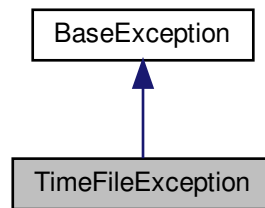
## 8.68 TimeFileException Class Reference

```
#include <TimeSystemConverter.hpp>
```

Inheritance diagram for TimeFileException:



Collaboration diagram for TimeFileException:



### Public Member Functions

- [TimeFileException](#) (const std::string &message="TimeSystemConverter: File is unknown: ")

### Additional Inherited Members

#### 8.68.1 Constructor & Destructor Documentation

##### 8.68.1.1 TimeFileException()

```
TimeFileException::TimeFileException (
 const std::string & message = "TimeSystemConverter: File is unknown: ") [inline]
```

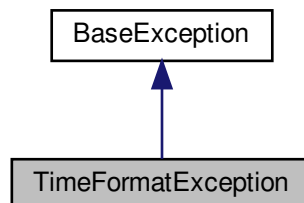
The documentation for this class was generated from the following file:

- GMATsrc/util/[TimeSystemConverter.hpp](#)

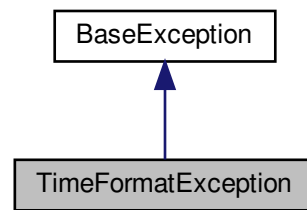
## 8.69 TimeFormatException Class Reference

```
#include <TimeSystemConverter.hpp>
```

Inheritance diagram for TimeFormatException:



Collaboration diagram for TimeFormatException:



### Public Member Functions

- [TimeFormatException](#) (const std::string &message="TimeSystemConverter: Requested format not implemented: ")

### Additional Inherited Members

#### 8.69.1 Constructor & Destructor Documentation

##### 8.69.1.1 TimeFormatException()

```
TimeFormatException::TimeFormatException (
 const std::string & message = "TimeSystemConverter: Requested format not implemented←
: ") [inline]
```

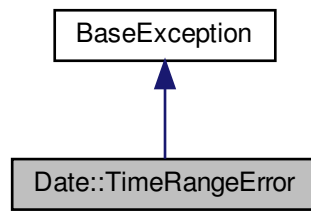
The documentation for this class was generated from the following file:

- GMATsrc/util/[TimeSystemConverter.hpp](#)

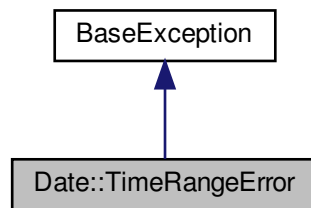
## 8.70 Date::TimeRangeError Class Reference

```
#include <Date.hpp>
```

Inheritance diagram for Date::TimeRangeError:



Collaboration diagram for Date::TimeRangeError:



## Public Member Functions

- [TimeRangeError](#) (const std::string &message="Date error: date or time out of specified range")

## Additional Inherited Members

### 8.70.1 Constructor & Destructor Documentation

#### 8.70.1.1 TimeRangeError()

```
Date::TimeRangeError::TimeRangeError (
 const std::string & message = "Date error: date or time out of specified range")
[inline]
```

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

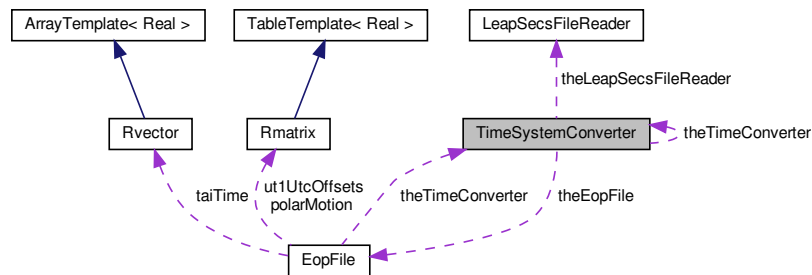
- GMATsrc/util/[Date.hpp](#)



## 8.71 TimeSystemConverter Class Reference

```
#include <TimeSystemConverter.hpp>
```

Collaboration diagram for TimeSystemConverter:



### Public Types

- enum `TimeSystemTypes` {  
`A1MJD = 0`, `TAIMJD`, `UTCMJD`, `UT1MJD`,  
`TDBMJD`, `TTMJD`, `A1`, `TAI`,  
`UTC`, `UT1`, `TDB`, `TT`,  
`TimeSystemCount` }

### Public Member Functions

- Integer `GetTimeTypeID` (std::string &str)
- Real `Convert` (const Real origValue, const Integer fromType, const Integer toType, Real refJd=GmatTimeConstants::JD\_JAN\_5\_1941, bool \*insideLeapSec=NULL)
- Real `ConvertToTaiMjd` (Integer fromType, Real origValue, Real refJd=GmatTimeConstants::JD\_NOV\_17\_1858, bool \*insideLeapSec=NULL)
- Real `ConvertFromTaiMjd` (Integer toType, Real origValue, Real refJd=GmatTimeConstants::JD\_NOV\_17\_1858, bool \*insideLeapSec=NULL)
- GmatTime `Convert` (const GmatTime origValue, const Integer fromType, const Integer toType, Real refJd=GmatTimeConstants::JD\_JAN\_5\_1941, bool \*insideLeapSec=NULL)
- GmatTime `ConvertToTaiMjd` (Integer fromType, GmatTime origValue, Real refJd=GmatTimeConstants::JD\_NOV\_17\_1858, bool \*insideLeapSec=NULL)
- GmatTime `ConvertFromTaiMjd` (Integer toType, GmatTime origValue, Real refJd=GmatTimeConstants::JD\_NOV\_17\_1858, bool \*insideLeapSec=NULL)
- Real `NumberOfLeapSecondsFrom` (Real utcMjd, Real jdOfMjdRef=GmatTimeConstants::JD\_JAN\_5\_1941)
- Real `GetFirstLeapSecondMJD` (Real fromUtcMjd, Real toUtcMjd, Real jdOfMjdRef=GmatTimeConstants::JD\_JAN\_5\_1941)
- void `SetEopFile` (EopFile \*eopFile)
- void `SetLeapSecsFileReader` (LeapSecsFileReader \*leapSecsFileReader)
- void `GetTimeSystemAndFormat` (const std::string &type, std::string &system, std::string &format)
- std::string `ConvertMjdToGregorian` (const Real mjd, bool handleLeapSecond=false, Integer format=1)
- Real `ConvertGregorianToMjd` (const std::string &greg)
- GmatTime `ConvertGregorianToMjdGT` (const std::string &greg)

- void [Convert](#) (const char \*fromType, [Real](#) fromMjd, const char \*fromStr, const char \*toType, [Real](#) &toMjd, std::string &toStr, [Integer](#) format=1, bool \*insideLeapSec=NULL)
- void [Convert](#) (const char \*fromType, [Real](#) fromMjd, const std::string &fromStr, const std::string &toType, [Real](#) &toMjd, std::string &toStr, [Integer](#) format=1, bool \*insideLeapSec=NULL)
- void [Convert](#) (const std::string &fromType, [Real](#) fromMjd, const std::string &fromStr, const std::string &toType, [Real](#) &toMjd, std::string &toStr, [Integer](#) format=1, bool \*insideLeapSec=NULL)
- void [Convert](#) (const char \*fromType, [GmatTime](#) fromMjd, const char \*fromStr, const char \*toType, [GmatTime](#) &toMjd, std::string &toStr, [Integer](#) format=1, bool \*insideLeapSec=NULL)
- void [Convert](#) (const char \*fromType, [GmatTime](#) fromMjd, const std::string &fromStr, const std::string &toType, [GmatTime](#) &toMjd, std::string &toStr, [Integer](#) format=1, bool \*insideLeapSec=NULL)
- void [Convert](#) (const std::string &fromType, [GmatTime](#) fromMjd, const std::string &fromStr, const std::string &toType, [GmatTime](#) &toMjd, std::string &toStr, [Integer](#) format=1, bool \*insideLeapSec=NULL)
- bool [ValidateTimeSystem](#) (std::string sys)
- bool [ValidateTimeFormat](#) (const std::string &format, const std::string &value, bool checkValue=true)
- [StringArray](#) [GetValidTimeRepresentations](#) ()
- bool [IsValidTimeSystem](#) (const std::string &system)

### Static Public Member Functions

- static [TimeSystemConverter](#) \* [Instance](#) ()

### Public Attributes

- const [Real](#) [TDB\\_COEFF1](#)
- const [Real](#) [TDB\\_COEFF2](#)
- const [Real](#) [M\\_E\\_OFFSET](#)
- const [Real](#) [M\\_E\\_COEFF1](#)
- const [Real](#) [T\\_TT\\_OFFSET](#)
- const [Real](#) [T\\_TT\\_COEFF1](#)
- const [Real](#) [L\\_B](#)
- const [Real](#) [NUM\\_SECS](#)

### Static Public Attributes

- static const std::string [TIME\\_SYSTEM\\_TEXT](#) [[TimeSystemCount](#)]

### Protected Member Functions

- bool [IsInLeapSecond](#) ([Real](#) theTaiMjd)
- bool [IsInLeapSecond](#) ([GmatTime](#) theTaiMjd)
- [TimeSystemConverter](#) ()
- [TimeSystemConverter](#) (const [TimeSystemConverter](#) &tcu)

### Protected Attributes

- [EopFile](#) \* [theEopFile](#)
- [LeapSecsFileReader](#) \* [theLeapSecsFileReader](#)

## Static Protected Attributes

- static [TimeSystemConverter](#) \* [theTimeConverter](#) = NULL

## 8.71.1 Member Enumeration Documentation

### 8.71.1.1 TimeSystemTypes

enum [TimeSystemConverter::TimeSystemTypes](#)

#### Enumerator

|                 |  |
|-----------------|--|
| A1MJD           |  |
| TAIMJD          |  |
| UTCMJD          |  |
| UT1MJD          |  |
| TDBMJD          |  |
| TTMJD           |  |
| A1              |  |
| TAI             |  |
| UTC             |  |
| UT1             |  |
| TDB             |  |
| TT              |  |
| TimeSystemCount |  |

## 8.71.2 Constructor & Destructor Documentation

### 8.71.2.1 TimeSystemConverter() [1/2]

[TimeSystemConverter::TimeSystemConverter](#) ( ) [protected]

### 8.71.2.2 TimeSystemConverter() [2/2]

[TimeSystemConverter::TimeSystemConverter](#) (  
const [TimeSystemConverter](#) & *tcu* ) [protected]

## 8.71.3 Member Function Documentation

**8.71.3.1 Convert()** [1/8]

```
Real TimeSystemConverter::Convert (
 const Real origValue,
 const Integer fromType,
 const Integer toType,
 Real refJd = GmatTimeConstants::JD_JAN_5_1941,
 bool * insideLeapSec = NULL)
```

Assignment operator for TimeConverter structures.

**Parameters**

|             |                                          |
|-------------|------------------------------------------|
| <origValue> | Given Time                               |
| <fromType>  | Time which is converted from date format |
| <toType>    | Time which is converted to date format   |
| <refJd>     | Reference Julian <a href="#">Date</a>    |

**Returns**

Converted time from the specific data format

**8.71.3.2 Convert()** [2/8]

```
GmatTime TimeSystemConverter::Convert (
 const GmatTime origValue,
 const Integer fromType,
 const Integer toType,
 Real refJd = GmatTimeConstants::JD_JAN_5_1941,
 bool * insideLeapSec = NULL)
```

**8.71.3.3 Convert()** [3/8]

```
void TimeSystemConverter::Convert (
 const char * fromType,
 Real fromMjd,
 const char * fromStr,
 const char * toType,
 Real & toMjd,
 std::string & toStr,
 Integer format = 1,
 bool * insideLeapSec = NULL)
```

#### 8.71.3.4 Convert() [4/8]

```
void TimeSystemConverter::Convert (
 const char * fromType,
 Real fromMjd,
 const std::string & fromStr,
 const std::string & toType,
 Real & toMjd,
 std::string & toStr,
 Integer format = 1,
 bool * insideLeapSec = NULL)
```

#### 8.71.3.5 Convert() [5/8]

```
void TimeSystemConverter::Convert (
 const std::string & fromType,
 Real fromMjd,
 const std::string & fromStr,
 const std::string & toType,
 Real & toMjd,
 std::string & toStr,
 Integer format = 1,
 bool * insideLeapSec = NULL)
```

#### 8.71.3.6 Convert() [6/8]

```
void TimeSystemConverter::Convert (
 const char * fromType,
 GmatTime fromMjd,
 const char * fromStr,
 const char * toType,
 GmatTime & toMjd,
 std::string & toStr,
 Integer format = 1,
 bool * insideLeapSec = NULL)
```

#### 8.71.3.7 Convert() [7/8]

```
void TimeSystemConverter::Convert (
 const char * fromType,
 GmatTime fromMjd,
 const std::string & fromStr,
 const std::string & toType,
 GmatTime & toMjd,
 std::string & toStr,
 Integer format = 1,
 bool * insideLeapSec = NULL)
```

**8.71.3.8 Convert()** [8/8]

```
void TimeSystemConverter::Convert (
 const std::string & fromType,
 GmatTime fromMjd,
 const std::string & fromStr,
 const std::string & toType,
 GmatTime & toMjd,
 std::string & toStr,
 Integer format = 1,
 bool * insideLeapSec = NULL)
```

**8.71.3.9 ConvertFromTaiMjd()** [1/2]

```
Real TimeSystemConverter::ConvertFromTaiMjd (
 Integer toType,
 Real origValue,
 Real refJd = GmatTimeConstants::JD_NOV_17_1858,
 bool * insideLeapSec = NULL)
```

Converts to the input time type from TAIMJD

**Parameters**

|             |                                       |
|-------------|---------------------------------------|
| <toType>    | Time type to convert to               |
| <origValue> | Original time value to convert        |
| <refJd>     | Reference Julian <a href="#">Date</a> |

**Exceptions**

|                     |                                                         |
|---------------------|---------------------------------------------------------|
| <TimeFileException> | thrown if <a href="#">LeapSecsFileReader</a> is not set |
| <TimeFileException> | thrown if <a href="#">EopFile</a> is not set            |

**Returns**

Time converted from TAIMJD to the input data format

**8.71.3.10 ConvertFromTaiMjd()** [2/2]

```
GmatTime TimeSystemConverter::ConvertFromTaiMjd (
 Integer toType,
 GmatTime origValue,
 Real refJd = GmatTimeConstants::JD_NOV_17_1858,
 bool * insideLeapSec = NULL)
```

### 8.71.3.11 ConvertGregorianToMjd()

```
Real TimeSystemConverter::ConvertGregorianToMjd (
 const std::string & greg)
```

Converts Gregorian to MJD date format.

#### Parameters

|        |                         |
|--------|-------------------------|
| <greg> | Input time in Gregorian |
|--------|-------------------------|

#### Exceptions

|                       |                                                                |
|-----------------------|----------------------------------------------------------------|
| <TimeFormatException> | thrown if input Gregorian date is not valid or is out of range |
|-----------------------|----------------------------------------------------------------|

#### Returns

Date in MJD format

### 8.71.3.12 ConvertGregorianToMjdGT()

```
GmatTime TimeSystemConverter::ConvertGregorianToMjdGT (
 const std::string & greg)
```

Converts Gregorian to MJD date format.

#### Parameters

|        |                         |
|--------|-------------------------|
| <greg> | Input time in Gregorian |
|--------|-------------------------|

#### Exceptions

|                       |                                                                |
|-----------------------|----------------------------------------------------------------|
| <TimeFormatException> | thrown if input Gregorian date is not valid or is out of range |
|-----------------------|----------------------------------------------------------------|

#### Returns

Date in MJD format

### 8.71.3.13 ConvertMjdToGregorian()

```
std::string TimeSystemConverter::ConvertMjdToGregorian (
 const Real mjd,
 bool handleLeapSecond = false,
 Integer format = 1)
```

Converts MJD to Gregorian date format.

## Parameters

|                             |                                                              |
|-----------------------------|--------------------------------------------------------------|
| < <i>mjd</i> >              | Input time in MJD                                            |
| < <i>handleLeapSecond</i> > | Do we need to handle a leap second (UTC only)?               |
| < <i>format</i> >           | 1 = "01 Jan 2000 11:59:28.000" 2 = "2000-01-01T11:59:28.000" |

## Returns

[Date](#) in Gregorian format

## 8.71.3.14 ConvertToTaiMjd() [1/2]

```
Real TimeSystemConverter::ConvertToTaiMjd (
 Integer fromType,
 Real origValue,
 Real refJd = GmatTimeConstants::JD_NOV_17_1858,
 bool * insideLeapSec = NULL)
```

Converts from the input time type to TAIMJD

## Parameters

|                      |                                       |
|----------------------|---------------------------------------|
| < <i>fromType</i> >  | Time type to convert from             |
| < <i>origValue</i> > | Original time value to convert        |
| < <i>refJd</i> >     | Reference Julian <a href="#">Date</a> |

## Exceptions

|                              |                                                         |
|------------------------------|---------------------------------------------------------|
| < <i>TimeFileException</i> > | thrown if <a href="#">LeapSecsFileReader</a> is not set |
| < <i>TimeFileException</i> > | thrown if <a href="#">EopFile</a> is not set            |

## Returns

Time converted from the input data format to TAIMJD

## 8.71.3.15 ConvertToTaiMjd() [2/2]

```
GmatTime TimeSystemConverter::ConvertToTaiMjd (
 Integer fromType,
 GmatTime origValue,
 Real refJd = GmatTimeConstants::JD_NOV_17_1858,
 bool * insideLeapSec = NULL)
```



**8.71.3.16 GetFirstLeapSecondMJD()**

```
Real TimeSystemConverter::GetFirstLeapSecondMJD (
 Real fromUtcMjd,
 Real toUtcMjd,
 Real jdOfMjdRef = GmatTimeConstants::JD_JAN_5_1941)
```

**8.71.3.17 GetTimeSystemAndFormat()**

```
void TimeSystemConverter::GetTimeSystemAndFormat (
 const std::string & type,
 std::string & system,
 std::string & format)
```

**8.71.3.18 GetTimeTypeID()**

```
Integer TimeSystemConverter::GetTimeTypeID (
 std::string & str)
```

**8.71.3.19 GetValidTimeRepresentations()**

```
StringArray TimeSystemConverter::GetValidTimeRepresentations ()
```

**8.71.3.20 Instance()**

```
TimeSystemConverter * TimeSystemConverter::Instance () [static]
```

**8.71.3.21 IsInLeapSecond()** [1/2]

```
bool TimeSystemConverter::IsInLeapSecond (
 Real theTaiMjd) [protected]
```

**8.71.3.22 IsInLeapSecond()** [2/2]

```
bool TimeSystemConverter::IsInLeapSecond (
 GmatTime theTaiMjd) [protected]
```

**8.71.3.23 IsValidTimeSystem()**

```
bool TimeSystemConverter::IsValidTimeSystem (
 const std::string & system)
```

Checks to see if a time system is valid

## Parameters

|               |                                    |
|---------------|------------------------------------|
| <i>system</i> | The descriptor for the time system |
|---------------|------------------------------------|

## Returns

true for valid systems, false for invalid systems

## 8.71.3.24 NumberOfLeapSecondsFrom()

```
Real TimeSystemConverter::NumberOfLeapSecondsFrom (
 Real utcMjd,
 Real jdOfMjdRef = GmatTimeConstants::JD_JAN_5_1941)
```

Retrives leap seconds from the leap second file.

## Parameters

|                   |                                                 |
|-------------------|-------------------------------------------------|
| <i>utcMjd</i>     | Modified julian days in UTC                     |
| <i>jdOfMjdRef</i> | Julidian days of modified julian days reference |

## Returns

Number of leap seconds

## 8.71.3.25 SetEopFile()

```
void TimeSystemConverter::SetEopFile (
 EopFile * eopFile)
```

Sets the EOP file for the TimsSystemConverter

## Parameters

|                    |                 |
|--------------------|-----------------|
| < <i>eopFile</i> > | EOP file to use |
|--------------------|-----------------|

## 8.71.3.26 SetLeapSecsFileReader()

```
void TimeSystemConverter::SetLeapSecsFileReader (
 LeapSecsFileReader * leapSecsFileReader)
```

Sets the Leap Seconds File reader

## Parameters

|                        |                                 |
|------------------------|---------------------------------|
| < leapSecsFileReader > | Leap Seconds File reader to use |
|------------------------|---------------------------------|

## 8.71.3.27 ValidateTimeFormat()

```
bool TimeSystemConverter::ValidateTimeFormat (
 const std::string & format,
 const std::string & value,
 bool checkValue = true)
```

## 8.71.3.28 ValidateTimeSystem()

```
bool TimeSystemConverter::ValidateTimeSystem (
 std::string sys)
```

## 8.71.4 Member Data Documentation

## 8.71.4.1 L\_B

```
const Real TimeSystemConverter::L_B
```

## 8.71.4.2 M\_E\_COEFF1

```
const Real TimeSystemConverter::M_E_COEFF1
```

## 8.71.4.3 M\_E\_OFFSET

```
const Real TimeSystemConverter::M_E_OFFSET
```

#### 8.71.4.4 NUM\_SECS

```
const Real TimeSystemConverter::NUM_SECS
```

#### 8.71.4.5 T\_TT\_COEFF1

```
const Real TimeSystemConverter::T_TT_COEFF1
```

#### 8.71.4.6 T\_TT\_OFFSET

```
const Real TimeSystemConverter::T_TT_OFFSET
```

#### 8.71.4.7 TDB\_COEFF1

```
const Real TimeSystemConverter::TDB_COEFF1
```

#### 8.71.4.8 TDB\_COEFF2

```
const Real TimeSystemConverter::TDB_COEFF2
```

#### 8.71.4.9 theEopFile

```
EopFile* TimeSystemConverter::theEopFile [protected]
```

#### 8.71.4.10 theLeapSecsFileReader

```
LeapSecsFileReader* TimeSystemConverter::theLeapSecsFileReader [protected]
```

#### 8.71.4.11 theTimeConverter

```
TimeSystemConverter * TimeSystemConverter::theTimeConverter = NULL [static], [protected]
```

## 8.71.4.12 TIME\_SYSTEM\_TEXT

```
const std::string TimeSystemConverter::TIME_SYSTEM_TEXT [static]
```

**Initial value:**

```
=
{
 "A1Mjd",
 "TaiMjd",
 "UtcMjd",
 "Ut1Mjd",
 "TdbMjd",
 "TtMjd",
 "A1",
 "TAI",
 "UTC",
 "UT1",
 "TDB",
 "TT",
}
```

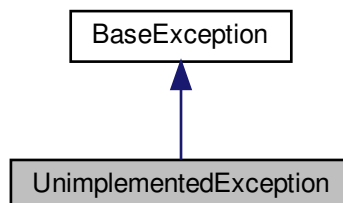
The documentation for this class was generated from the following files:

- [GMATsrc/util/TimeSystemConverter.hpp](#)
- [GMATsrc/util/TimeSystemConverter.cpp](#)

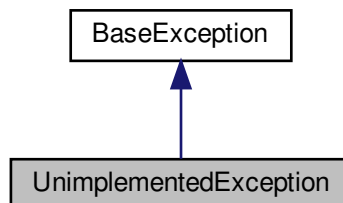
## 8.72 UnimplementedException Class Reference

```
#include <TimeSystemConverter.hpp>
```

Inheritance diagram for UnimplementedException:



Collaboration diagram for UnimplementedException:



## Public Member Functions

- [UnimplementedException](#) (const std::string &message="TimeSystemConverter: Conversion not implemented↵↵ : ")

## Additional Inherited Members

### 8.72.1 Detailed Description

Definition of the [TimeSystemConverter](#) class

### 8.72.2 Constructor & Destructor Documentation

#### 8.72.2.1 UnimplementedException()

```
UnimplementedException::UnimplementedException (
 const std::string & message = "TimeSystemConverter: Conversion not implemented: "
) [inline]
```

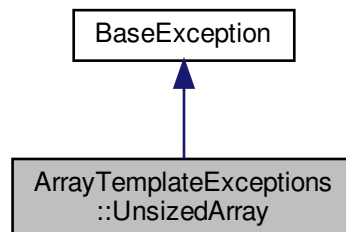
The documentation for this class was generated from the following file:

- GMATsrc/util/[TimeSystemConverter.hpp](#)

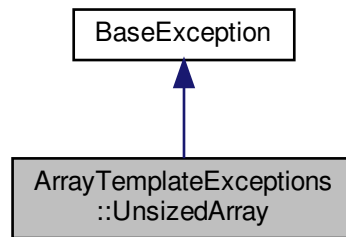
## 8.73 ArrayTemplateExceptions::UnsizeArray Class Reference

```
#include <ArrayTemplate.hpp>
```

Inheritance diagram for ArrayTemplateExceptions::UnsizeArray:



Collaboration diagram for ArrayTemplateExceptions::UnsizeArray:



### Public Member Functions

- [UnsizeArray](#) (const std::string &message="ArrayTemplate error : unsize array.")

### Additional Inherited Members

#### 8.73.1 Constructor & Destructor Documentation

##### 8.73.1.1 UnsizeArray()

```
ArrayTemplateExceptions::UnsizeArray::UnsizeArray (
 const std::string & message = "ArrayTemplate error : unsize array.") [inline]
```

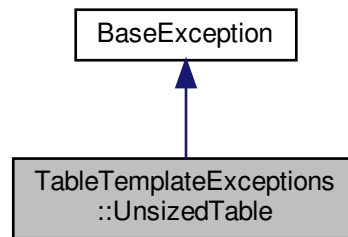
The documentation for this class was generated from the following file:

- GMATsrc/util/[ArrayTemplate.hpp](#)

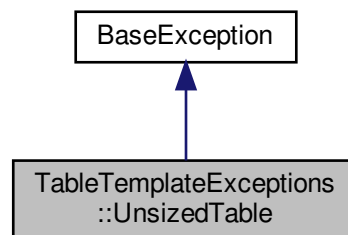
## 8.74 TableTemplateExceptions::UnsizeTable Class Reference

```
#include <TableTemplate.hpp>
```

Inheritance diagram for TableTemplateExceptions::UnsizeTable:



Collaboration diagram for TableTemplateExceptions::UnsizeTable:



## Public Member Functions

- [UnsizeTable](#) (const std::string &message="TableTemplate error : unsize table.\n")

## Additional Inherited Members

### 8.74.1 Constructor & Destructor Documentation

#### 8.74.1.1 UnsizeTable()

```
TableTemplateExceptions::UnsizeTable::UnsizeTable (
 const std::string & message = "TableTemplate error : unsize table.\n") [inline]
```

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

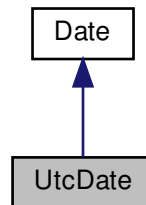
- GMATsrc/util/[TableTemplate.hpp](#)



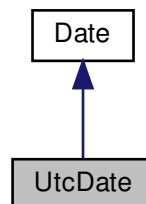
## 8.75 UtcDate Class Reference

```
#include <UtcDate.hpp>
```

Inheritance diagram for UtcDate:



Collaboration diagram for UtcDate:



### Public Member Functions

- `UtcDate ()`
- `UtcDate (Integer year, Integer month, Integer day, Integer hour, Integer minute, Real second)`
- `UtcDate (Integer year, Integer doy, Integer hour, Integer minute, Real second)`
- `UtcDate (Integer year, Integer month, Integer day, Real secondsOfDay)`
- `UtcDate (const GmatTimeUtil::CalDate &date)`
- `UtcDate (const std::string &time)`
- `UtcDate (const UtcDate &date)`
- `UtcDate operator= (const UtcDate &date)`
- `~UtcDate ()`
- `Real ToA1Mjd () const`

## Additional Inherited Members

### 8.75.1 Constructor & Destructor Documentation

#### 8.75.1.1 `UtcDate()` [1/7]

```
UtcDate::UtcDate ()
```

##### Note

Calls [A1Mjd](#) default constructor which creates an object with 0 second from reference.

#### 8.75.1.2 `UtcDate()` [2/7]

```
UtcDate::UtcDate (
 Integer year,
 Integer month,
 Integer day,
 Integer hour,
 Integer minute,
 Real second)
```

##### Note

Assumes input date is in UTC time system.

#### 8.75.1.3 `UtcDate()` [3/7]

```
UtcDate::UtcDate (
 Integer year,
 Integer dayOfYear,
 Integer hour,
 Integer minute,
 Real second)
```

##### Note

Assumes input date is in UTC time system.

#### 8.75.1.4 UtcDate() [4/7]

```
UtcDate::UtcDate (
 Integer year,
 Integer month,
 Integer day,
 Real secondsOfDay)
```

##### Note

Assumes input date is in UTC time system.

#### 8.75.1.5 UtcDate() [5/7]

```
UtcDate::UtcDate (
 const GmatTimeUtil::CalDate & date)
```

##### Note

Assumes input date is in UTC time system.

#### 8.75.1.6 UtcDate() [6/7]

```
UtcDate::UtcDate (
 const std::string & time)
```

##### Parameters

|        |                                   |
|--------|-----------------------------------|
| <time> | Time in "YYMMDD.hhmmssnnn" format |
|--------|-----------------------------------|

##### Note

Assumes input date is in UTC time system.

#### 8.75.1.7 UtcDate() [7/7]

```
UtcDate::UtcDate (
 const UtcDate & date)
```

#### 8.75.1.8 ~UtcDate()

```
UtcDate::~~UtcDate ()
```

### 8.75.2 Member Function Documentation

#### 8.75.2.1 operator=()

```
UtcDate UtcDate::operator= (
 const UtcDate & date)
```

#### 8.75.2.2 ToA1Mjd()

```
Real UtcDate::ToA1Mjd () const
```

#### Note

The two time systems label time differently. At any given moment, the A.1 system is several seconds ahead of the UTC system. This offset is constant between leap insertions. For example, the instant of time labeled July 1, 1992, 12:00:27.0343817 in the A.1 system will be labeled July 1, 1992, 12:00:00 (Noon) in the UTC system. Converts from [UtcDate](#) to A1 Modified Julian date.

#### Returns

A1 Modified Julian [Date](#)

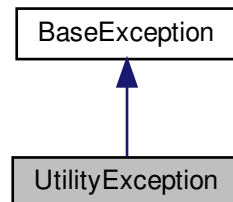
The documentation for this class was generated from the following files:

- GMATsrc/util/[UtcDate.hpp](#)
- GMATsrc/util/[UtcDate.cpp](#)

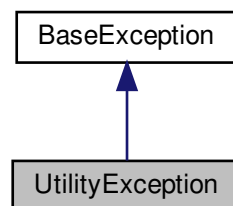
## 8.76 UtilityException Class Reference

```
#include <UtilityException.hpp>
```

Inheritance diagram for UtilityException:



Collaboration diagram for UtilityException:



### Public Member Functions

- [UtilityException](#) (const std::string &details="")

### Additional Inherited Members

#### 8.76.1 Detailed Description

Declares utility exception.

#### 8.76.2 Constructor & Destructor Documentation

### 8.76.2.1 `UtilityException()`

```
UtilityException::UtilityException (
 const std::string & details = "") [inline]
```

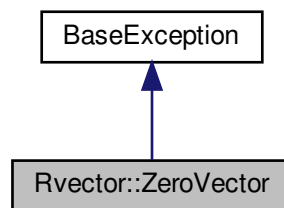
The documentation for this class was generated from the following file:

- GMATsrc/util/[UtilityException.hpp](#)

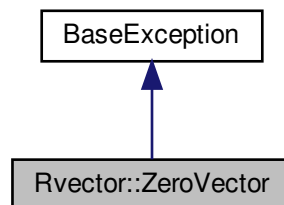
## 8.77 `Rvector::ZeroVector` Class Reference

```
#include <Rvector.hpp>
```

Inheritance diagram for `Rvector::ZeroVector`:



Collaboration diagram for `Rvector::ZeroVector`:



### Public Member Functions

- [ZeroVector](#) (const std::string &details="")

## Additional Inherited Members

### 8.77.1 Constructor & Destructor Documentation

#### 8.77.1.1 ZeroVector()

```
Rvector::ZeroVector::ZeroVector (
 const std::string & details = "") [inline]
```

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

- GMATsrc/util/[Rvector.hpp](#)





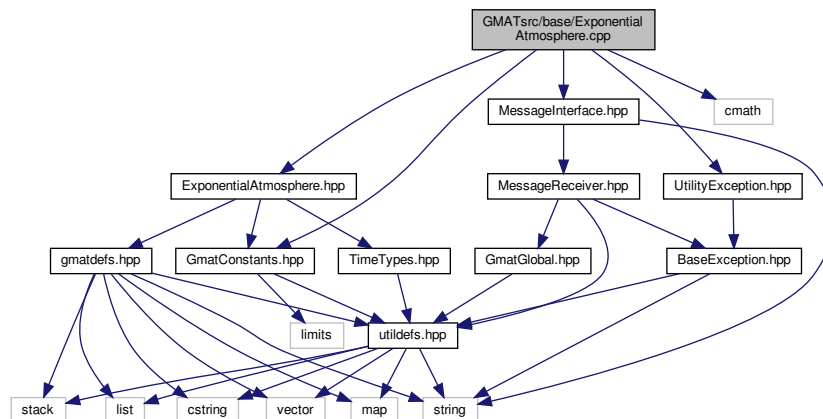
## Chapter 9

# File Documentation

### 9.1 GMATsrc/base/ExponentialAtmosphere.cpp File Reference

```
#include "ExponentialAtmosphere.hpp"
#include <cmath>
#include "MessageInterface.hpp"
#include "GmatConstants.hpp"
#include "UtilityException.hpp"
```

Include dependency graph for ExponentialAtmosphere.cpp:

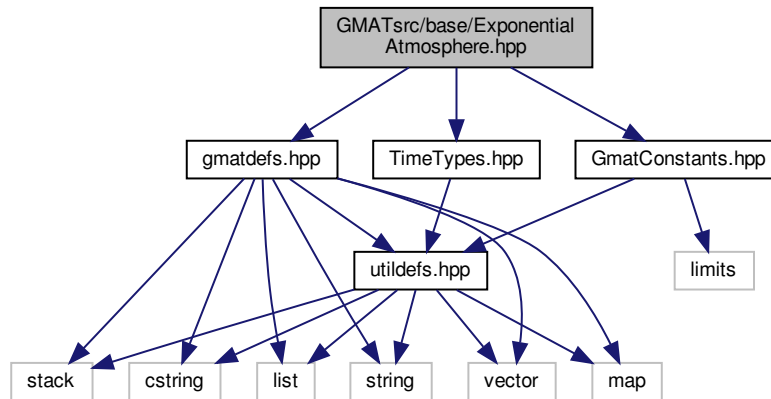


### 9.2 GMATsrc/base/ExponentialAtmosphere.hpp File Reference

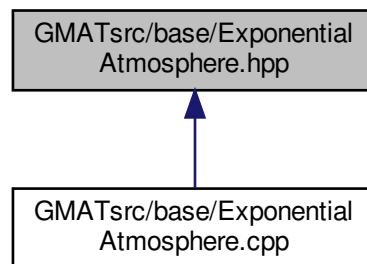
```
#include "gmdefs.hpp"
#include "TimeTypes.hpp"
```

```
#include "GmatConstants.hpp"
```

Include dependency graph for ExponentialAtmosphere.hpp:



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



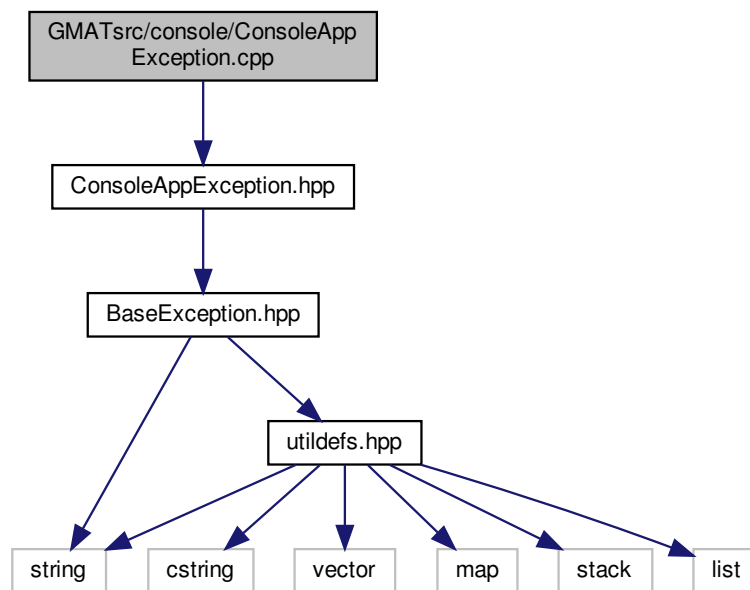
## Classes

- class [ExponentialAtmosphere](#)

## 9.3 GMATsrc/console/ConsoleAppException.cpp File Reference

```
#include "ConsoleAppException.hpp"
```

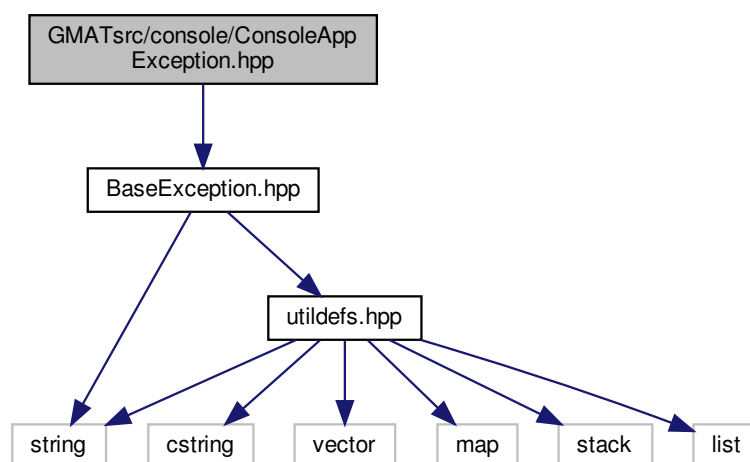
Include dependency graph for ConsoleAppException.cpp:



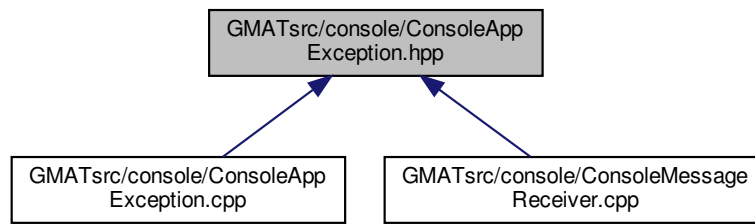
## 9.4 GMATsrc/console/ConsoleAppException.hpp File Reference

```
#include "BaseException.hpp"
```

Include dependency graph for ConsoleAppException.hpp:



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



## Classes

- class [ConsoleAppException](#)

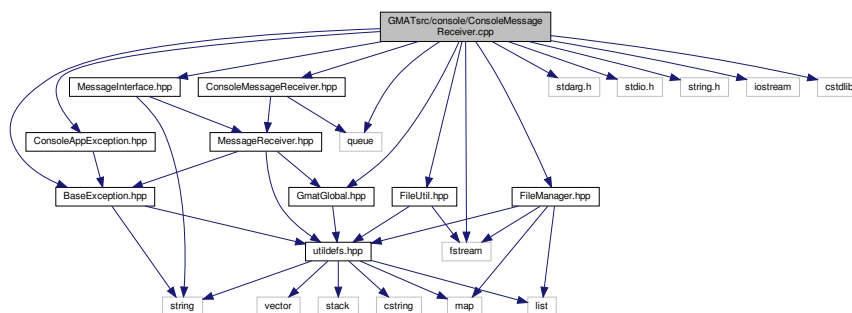
## 9.5 GMATsrc/console/ConsoleMessageReceiver.cpp File Reference

```

#include "ConsoleMessageReceiver.hpp"
#include <stdarg.h>
#include <stdio.h>
#include <string.h>
#include <iostream>
#include <fstream>
#include <queue>
#include "MessageInterface.hpp"
#include "BaseException.hpp"
#include "ConsoleAppException.hpp"
#include "FileManager.hpp"
#include "FileUtil.hpp"
#include "GmatGlobal.hpp"
#include <cstdlib>

```

Include dependency graph for ConsoleMessageReceiver.cpp:

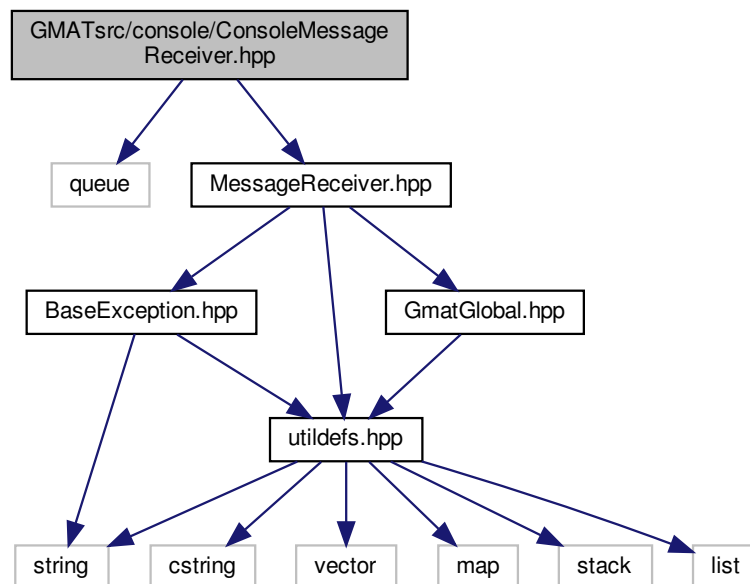


## 9.6 GMATsrc/console/ConsoleMessageReceiver.hpp File Reference

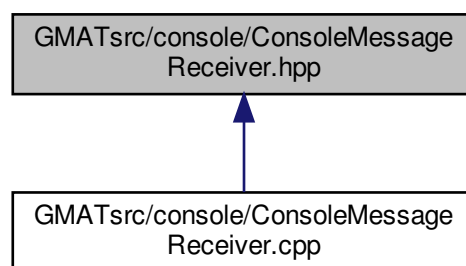
```
#include <queue>
```

```
#include "MessageReceiver.hpp"
```

Include dependency graph for ConsoleMessageReceiver.hpp:



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



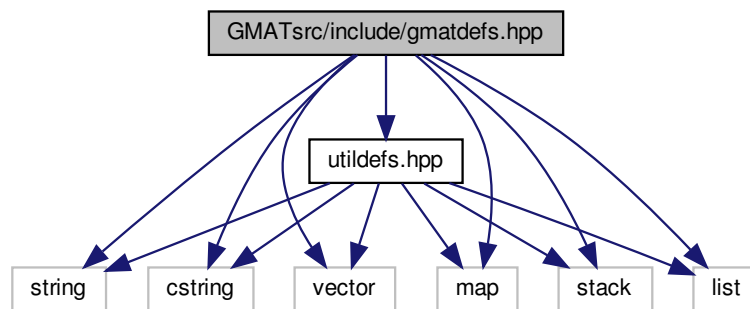
### Classes

- class [ConsoleMessageReceiver](#)

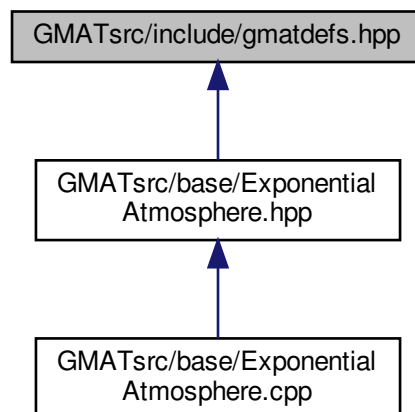
## 9.7 GMATsrc/include/gmatdefs.hpp File Reference

```
#include <string>
#include <cstring>
#include <vector>
#include <map>
#include <stack>
#include <list>
#include "utildefs.hpp"
```

Include dependency graph for gmatdefs.hpp:



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



### Classes

- struct [geoparms](#)
- struct [Gmat::PluginResource](#)

## Namespaces

- [Gmat](#)

## Macros

- `#define` [GMAT\\_API](#)
- `#define` [DEFAULT\\_TO\\_NO\\_CLONES](#) virtual bool HasLocalClones() { return false; }
- `#define` [DEFAULT\\_TO\\_NO\\_REFOBJECTS](#)

## Typedefs

- typedef double [Real](#)
- typedef int [Integer](#)
- typedef unsigned char [Byte](#)
- typedef unsigned int [UnsignedInt](#)
- typedef std::vector< [Real](#) > [RealArray](#)
- typedef std::vector< [Integer](#) > [IntegerArray](#)
- typedef std::vector< [UnsignedInt](#) > [UnsignedIntArray](#)
- typedef std::vector< std::string > [StringArray](#)
- typedef std::vector< bool > [BooleanArray](#)
- typedef std::vector< GmatBase \* > [ObjectArray](#)
- typedef std::vector< ElementWrapper \* > [WrapperArray](#)
- typedef std::vector< [Rvector6](#) \* > [StateArray](#)
- typedef std::vector< [A1Mjd](#) \* > [EpochArray](#)
- typedef std::map< std::string, [Integer](#) > [IntegerMap](#)
- typedef std::map< std::string, [UnsignedInt](#) > [ColorMap](#)
- typedef std::map< std::string, GmatBase \* > [ObjectMap](#)
- typedef std::map< std::string, ElementWrapper \* > [WrapperMap](#)
- typedef std::stack< [ObjectMap](#) \* > [ObjectMapStack](#)
- typedef struct [geoparms](#) [GEOPARMS](#)
- typedef [Real](#) [GmatEpoch](#)  
*GMAT's epoch representation; eventually a struct holding MJ day & sec of day.*
- typedef [Real](#) [Radians](#)
- typedef struct [Gmat::PluginResource](#) [Gmat::PLUGIN\\_RESOURCE](#)

## Enumerations

- enum [Gmat::ObjectType](#) {  
[Gmat::SPACECRAFT](#) = 101, [Gmat::FORMATION](#), [Gmat::SPACEOBJECT](#), [Gmat::GROUND\\_STATION](#),  
[Gmat::BURN](#), [Gmat::IMPULSIVE\\_BURN](#), [Gmat::FINITE\\_BURN](#), [Gmat::COMMAND](#),  
[Gmat::PROPAGATOR](#), [Gmat::ODE\\_MODEL](#), [Gmat::PHYSICAL\\_MODEL](#), [Gmat::TRANSIENT\\_FORCE](#),  
[Gmat::INTERPOLATOR](#), [Gmat::SOLAR\\_SYSTEM](#), [Gmat::SPACE\\_POINT](#), [Gmat::CELESTIAL\\_BODY](#),  
[Gmat::CALCULATED\\_POINT](#), [Gmat::LIBRATION\\_POINT](#), [Gmat::BARYCENTER](#), [Gmat::ATMOSPHERE](#),  
[Gmat::PARAMETER](#), [Gmat::VARIABLE](#), [Gmat::ARRAY](#), [Gmat::STRING](#),  
[Gmat::STOP\\_CONDITION](#), [Gmat::SOLVER](#), [Gmat::SUBSCRIBER](#), [Gmat::REPORT\\_FILE](#),  
[Gmat::XY\\_PLOT](#), [Gmat::ORBIT\\_VIEW](#), [Gmat::DYNAMIC\\_DATA\\_DISPLAY](#), [Gmat::EPHEMERIS\\_FILE](#),  
[Gmat::PROP\\_SETUP](#), [Gmat::FUNCTION](#), [Gmat::FUEL\\_TANK](#), [Gmat::THRUSTER](#),  
[Gmat::CHEMICAL\\_THRUSTER](#), [Gmat::ELECTRIC\\_THRUSTER](#), [Gmat::CHEMICAL\\_FUEL\\_TANK](#), [Gmat::ELECTRIC\\_FUEL\\_TANK](#),  
[Gmat::POWER\\_SYSTEM](#), [Gmat::SOLAR\\_POWER\\_SYSTEM](#), [Gmat::NUCLEAR\\_POWER\\_SYSTEM](#),  
[Gmat::HARDWARE](#),

```

Gmat::COORDINATE_SYSTEM, Gmat::AXIS_SYSTEM, Gmat::ATTITUDE, Gmat::MATH_NODE,
Gmat::MATH_TREE, Gmat::BODY_FIXED_POINT, Gmat::EVENT, Gmat::EVENT_LOCATOR,
Gmat::DATAINTERFACE_SOURCE, Gmat::MEASUREMENT_MODEL, Gmat::ERROR_MODEL, Gmat::←
DATASTREAM,
Gmat::DATA_FILE, Gmat::OBTTYPE, Gmat::DATA_FILTER, Gmat::INTERFACE,
Gmat::MEDIA_CORRECTION, Gmat::SENSOR, Gmat::RF_HARDWARE, Gmat::ANTENNA,
Gmat::USER_DEFINED_OBJECT, Gmat::USER_OBJECT_ID_NEEDED = USER_DEFINED_OBJECT +
500, Gmat::GENERIC_OBJECT, Gmat::UNKNOWN_OBJECT }
• enum Gmat::WriteMode {
 Gmat::SCRIPTING, Gmat::SHOW_SCRIPT, Gmat::OWNED_OBJECT, Gmat::MATLAB_STRUCT,
 Gmat::EPHEM_HEADER, Gmat::NO_COMMENTS, Gmat::GUI_EDITOR, Gmat::OBJECT_EXPORT }
• enum Gmat::StateElementId {
 Gmat::UNKNOWN_STATE = -1, Gmat::CARTESIAN_STATE = 3700, Gmat::EQUINOCTIAL_STATE,
 Gmat::ORBIT_STATE_TRANSITION_MATRIX,
 Gmat::ORBIT_A_MATRIX, Gmat::MASS_FLOW, Gmat::PREDEFINED_STATE_MAX, Gmat::USER_DEF←
INED_BEGIN = 3800,
 Gmat::USER_DEFINED_END = 3999 }

```

## 9.7.1 Macro Definition Documentation

### 9.7.1.1 DEFAULT\_TO\_NO\_CLONES

```
#define DEFAULT_TO_NO_CLONES virtual bool HasLocalClones() { return false; }
```

### 9.7.1.2 DEFAULT\_TO\_NO\_REFOBJECTS

```
#define DEFAULT_TO_NO_REFOBJECTS
```

#### Value:

```

virtual bool RenameRefObject(\
 const UnsignedInt type, const std::string &oldName, \
 const std::string &newName) { return true; }

```

### 9.7.1.3 GMAT\_API

```
#define GMAT_API
```

## 9.7.2 Typedef Documentation



### 9.7.2.1 BooleanArray

```
typedef std::vector<bool> BooleanArray
```

### 9.7.2.2 Byte

```
typedef unsigned char Byte
```

### 9.7.2.3 ColorMap

```
typedef std::map<std::string, UnsignedInt> ColorMap
```

### 9.7.2.4 EpochArray

```
typedef std::vector<AlMjd*> EpochArray
```

### 9.7.2.5 GEOPARMS

```
typedef struct geoparms GEOPARMS
```

### 9.7.2.6 GmatEpoch

```
typedef Real GmatEpoch
```

GMAT's epoch representation; eventually a struct holding MJ day & sec of day.

### 9.7.2.7 Integer

```
typedef int Integer
```

#### 9.7.2.8 IntegerArray

```
typedef std::vector<Integer> IntegerArray
```

#### 9.7.2.9 IntegerMap

```
typedef std::map<std::string, Integer> IntegerMap
```

#### 9.7.2.10 ObjectArray

```
typedef std::vector<GmatBase*> ObjectArray
```

#### 9.7.2.11 ObjectMap

```
typedef std::map<std::string, GmatBase*> ObjectMap
```

#### 9.7.2.12 ObjectMapStack

```
typedef std::stack<ObjectMap*> ObjectMapStack
```

#### 9.7.2.13 Radians

```
typedef Real Radians
```

#### 9.7.2.14 Real

```
typedef double Real
```

#### 9.7.2.15 RealArray

```
typedef std::vector<Real> RealArray
```

#### 9.7.2.16 StateArray

```
typedef std::vector<Rvector6*> StateArray
```

#### 9.7.2.17 StringArray

```
typedef std::vector<std::string> StringArray
```

#### 9.7.2.18 UnsignedInt

```
typedef unsigned int UnsignedInt
```

#### 9.7.2.19 UnsignedIntArray

```
typedef std::vector<UnsignedInt> UnsignedIntArray
```

#### 9.7.2.20 WrapperArray

```
typedef std::vector<ElementWrapper*> WrapperArray
```

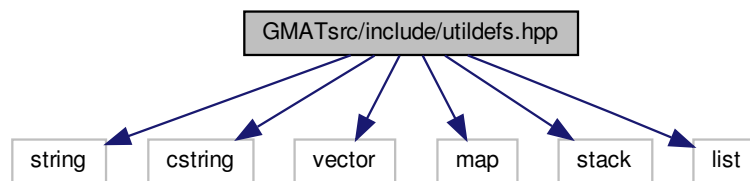
#### 9.7.2.21 WrapperMap

```
typedef std::map<std::string, ElementWrapper*> WrapperMap
```

## 9.8 GMATsrc/include/utldefs.hpp File Reference

```
#include <string>
#include <cstring>
#include <vector>
#include <map>
#include <stack>
#include <list>
```

Include dependency graph for utldefs.hpp:



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



### Namespaces

- [Gmat](#)

### Macros

- `#define` [GMATUTIL\\_API](#)

### Typedefs

- typedef double [Real](#)
- typedef int [Integer](#)
- typedef unsigned char [Byte](#)
- typedef unsigned int [UnsignedInt](#)
- typedef std::vector< [Real](#) > [RealArray](#)
- typedef std::vector< [Integer](#) > [IntegerArray](#)
- typedef std::vector< [UnsignedInt](#) > [UnsignedIntArray](#)
- typedef std::vector< std::string > [StringArray](#)
- typedef std::vector< bool > [BooleanArray](#)
- typedef std::vector< [GmatBase](#) \* > [ObjectArray](#)

- typedef std::vector< ElementWrapper \* > WrapperArray
- typedef std::vector< Rvector6 \* > StateArray
- typedef std::vector< A1Mjd \* > EpochArray
- typedef std::map< std::string, Integer > IntegerMap
- typedef std::map< std::string, UnsignedInt > ColorMap
- typedef std::map< std::string, GmatBase \* > ObjectMap
- typedef std::map< std::string, ElementWrapper \* > WrapperMap
- typedef std::stack< ObjectMap \* > ObjectMapStack
- typedef Real GmatEpoch  
*GMAT's epoch representation; eventually a struct holding MJ day & sec of day.*
- typedef Real Radians  
*GMAT's Radians representation.*
- typedef std::vector< UnsignedInt > ObjectTypeArray
- typedef std::vector< Gmat::WrapperDataType > WrapperTypeArray
- typedef std::map< std::string, UnsignedInt > ObjectTypeMap
- typedef std::map< UnsignedInt, StringArray > ObjectTypeArrayMap

## Enumerations

- enum Gmat::ParameterType {  
Gmat::INTEGER\_TYPE, Gmat::UNSIGNED\_INT\_TYPE, Gmat::UNSIGNED\_INTARRAY\_TYPE, Gmat::IN←  
TARRAY\_TYPE,  
Gmat::REAL\_TYPE, Gmat::REAL\_ELEMENT\_TYPE, Gmat::STRING\_TYPE, Gmat::STRINGARRAY\_TY←  
PE,  
Gmat::BOOLEAN\_TYPE, Gmat::BOOLEANARRAY\_TYPE, Gmat::RVECTOR\_TYPE, Gmat::RMATRIX\_T←  
YPE,  
Gmat::TIME\_TYPE, Gmat::OBJECT\_TYPE, Gmat::OBJECTARRAY\_TYPE, Gmat::ON\_OFF\_TYPE,  
Gmat::ENUMERATION\_TYPE, Gmat::FILENAME\_TYPE, Gmat::COLOR\_TYPE, Gmat::GMATTIME\_TYPE,  
Gmat::TypeCount, Gmat::UNKNOWN\_PARAMETER\_TYPE = -1, Gmat::PARAMETER\_REMOVED = -3 }
- enum Gmat::MessageType {  
Gmat::ERROR\_ = 10, Gmat::WARNING\_, Gmat::INFO\_, Gmat::DEBUG\_,  
Gmat::GENERAL\_ }
- enum Gmat::RunState {  
Gmat::IDLE = 10000, Gmat::RUNNING, Gmat::PAUSED, Gmat::TARGETING,  
Gmat::OPTIMIZING, Gmat::ESTIMATING, Gmat::SOLVING, Gmat::SOLVEDPASS,  
Gmat::WAITING }
- enum Gmat::WrapperDataType {  
Gmat::NUMBER\_WT, Gmat::MATRIX\_WT, Gmat::STRING\_WT, Gmat::STRING\_OBJECT\_WT,  
Gmat::OBJECT\_PROPERTY\_WT, Gmat::VARIABLE\_WT, Gmat::ARRAY\_WT, Gmat::ARRAY\_ELEMEN←  
T\_WT,  
Gmat::PARAMETER\_WT, Gmat::OBJECT\_WT, Gmat::BOOLEAN\_WT, Gmat::INTEGER\_WT,  
Gmat::ON\_OFF\_WT, Gmat::UNKNOWN\_WRAPPER\_TYPE = -2 }

### 9.8.1 Macro Definition Documentation

#### 9.8.1.1 GMATUTIL\_API

```
#define GMATUTIL_API
```

Types and definitions used in the GmatUtil library

## 9.8.2 Typedef Documentation

### 9.8.2.1 BooleanArray

```
typedef std::vector<bool> BooleanArray
```

### 9.8.2.2 Byte

```
typedef unsigned char Byte
```

### 9.8.2.3 ColorMap

```
typedef std::map<std::string, UnsignedInt> ColorMap
```

### 9.8.2.4 EpochArray

```
typedef std::vector<AlMjd*> EpochArray
```

### 9.8.2.5 GmatEpoch

```
typedef Real GmatEpoch
```

GMAT's epoch representation; eventually a struct holding MJ day & sec of day.

### 9.8.2.6 Integer

```
typedef int Integer
```

### 9.8.2.7 IntegerArray

```
typedef std::vector<Integer> IntegerArray
```

#### 9.8.2.8 IntegerMap

```
typedef std::map<std::string, Integer> IntegerMap
```

#### 9.8.2.9 ObjectArray

```
typedef std::vector<GmatBase*> ObjectArray
```

#### 9.8.2.10 ObjectMap

```
typedef std::map<std::string, GmatBase*> ObjectMap
```

#### 9.8.2.11 ObjectMapStack

```
typedef std::stack<ObjectMap*> ObjectMapStack
```

#### 9.8.2.12 ObjectTypeArray

```
typedef std::vector<UnsignedInt> ObjectTypeArray
```

#### 9.8.2.13 ObjectTypeArrayMap

```
typedef std::map<UnsignedInt, StringArray> ObjectTypeArrayMap
```

#### 9.8.2.14 ObjectTypeMap

```
typedef std::map<std::string, UnsignedInt> ObjectTypeMap
```

#### 9.8.2.15 Radians

```
typedef Real Radians
```

GMAT's Radians representation.

#### 9.8.2.16 Real

```
typedef double Real
```

#### 9.8.2.17 RealArray

```
typedef std::vector<Real> RealArray
```

#### 9.8.2.18 StateArray

```
typedef std::vector<Rvector6*> StateArray
```

#### 9.8.2.19 StringArray

```
typedef std::vector<std::string> StringArray
```

#### 9.8.2.20 UnsignedInt

```
typedef unsigned int UnsignedInt
```

#### 9.8.2.21 UnsignedIntArray

```
typedef std::vector<UnsignedInt> UnsignedIntArray
```



## 9.8.2.22 WrapperArray

```
typedef std::vector<ElementWrapper*> WrapperArray
```

## 9.8.2.23 WrapperMap

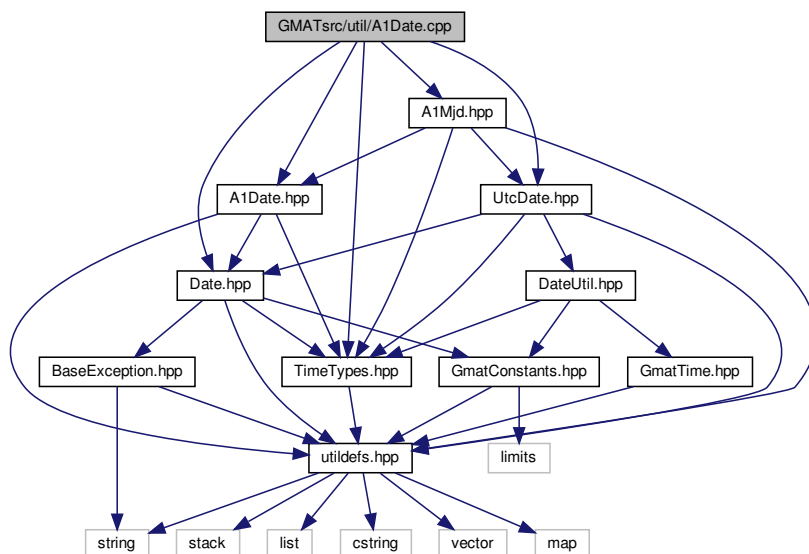
```
typedef std::map<std::string, ElementWrapper*> WrapperMap
```

## 9.8.2.24 WrapperTypeArray

```
typedef std::vector<Gmat::WrapperDataType> WrapperTypeArray
```

## 9.9 GMATsrc/util/A1Date.cpp File Reference

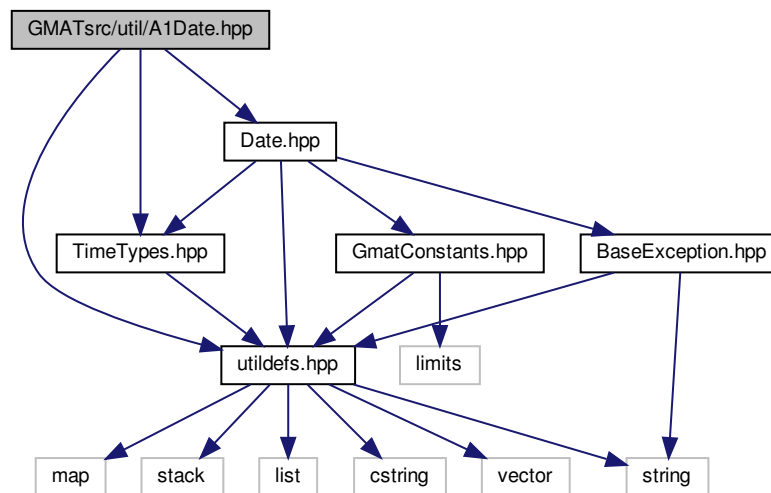
```
#include "A1Date.hpp"
#include "Date.hpp"
#include "TimeTypes.hpp"
#include "UtcDate.hpp"
#include "A1Mjd.hpp"
Include dependency graph for A1Date.cpp:
```



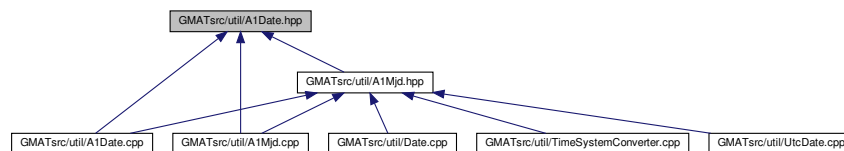
## 9.10 GMATsrc/util/A1Date.hpp File Reference

```
#include "utildefs.hpp"
#include "TimeTypes.hpp"
#include "Date.hpp"
```

Include dependency graph for A1Date.hpp:



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



### Classes

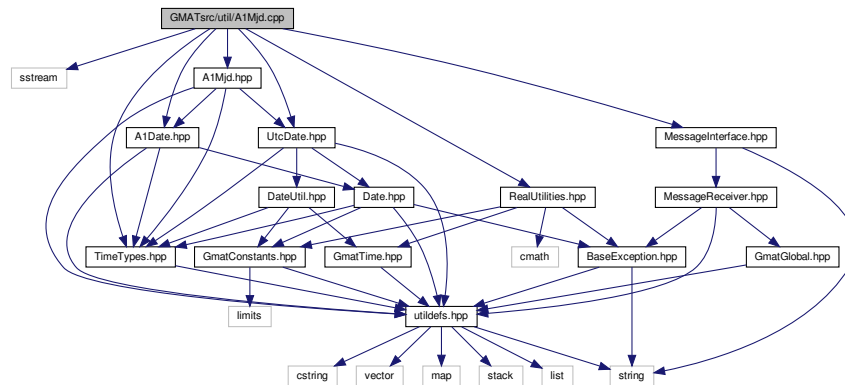
- class [A1Date](#)

## 9.11 GMATsrc/util/A1Mjd.cpp File Reference

```
#include <sstream>
#include "A1Mjd.hpp"
#include "UtcDate.hpp"
#include "A1Date.hpp"
#include "TimeTypes.hpp"
#include "RealUtilities.hpp"
```

```
#include "MessageInterface.hpp"
```

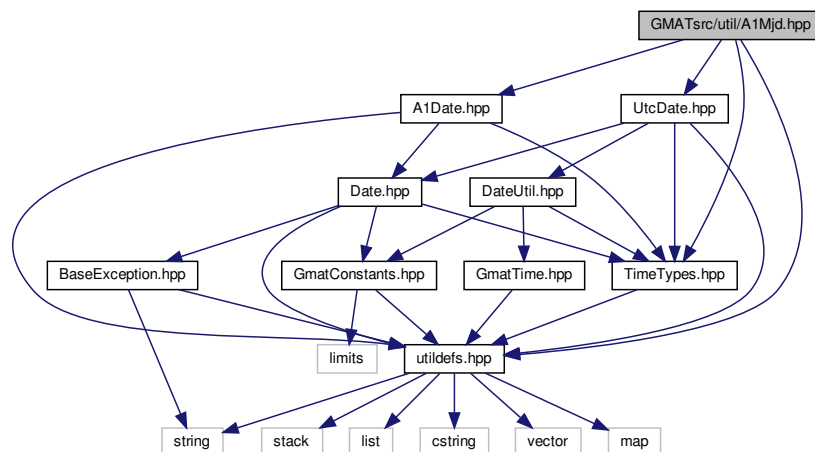
Include dependency graph for A1Mjd.cpp:



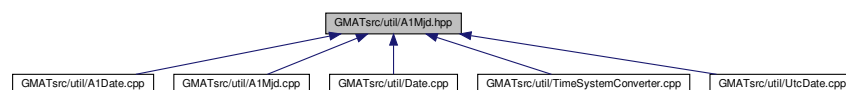
## 9.12 GMATsrc/util/A1Mjd.hpp File Reference

```
#include "utildefs.hpp"
#include "TimeTypes.hpp"
#include "UtcDate.hpp"
#include "A1Date.hpp"
```

Include dependency graph for A1Mjd.hpp:



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



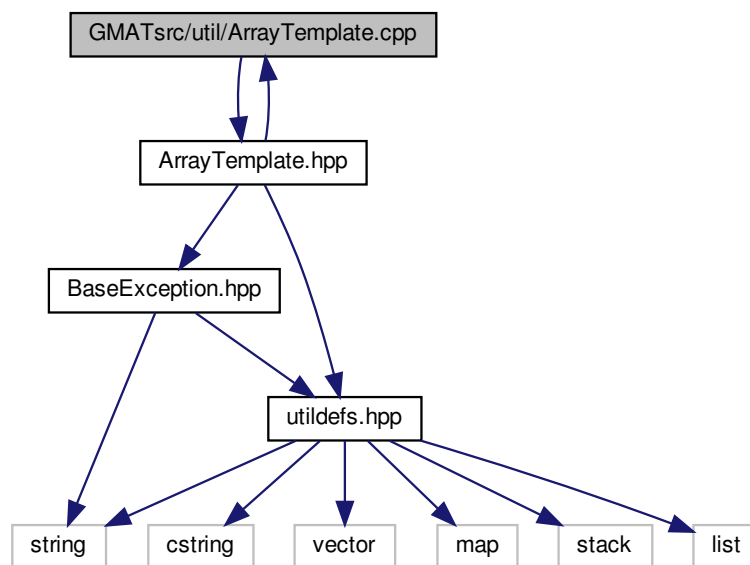
## Classes

- class [A1Mjd](#)

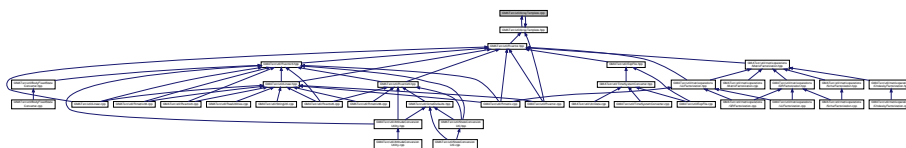
## 9.13 GMATsrc/util/ArrayTemplate.cpp File Reference

```
#include "ArrayTemplate.hpp"
```

Include dependency graph for ArrayTemplate.cpp:



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



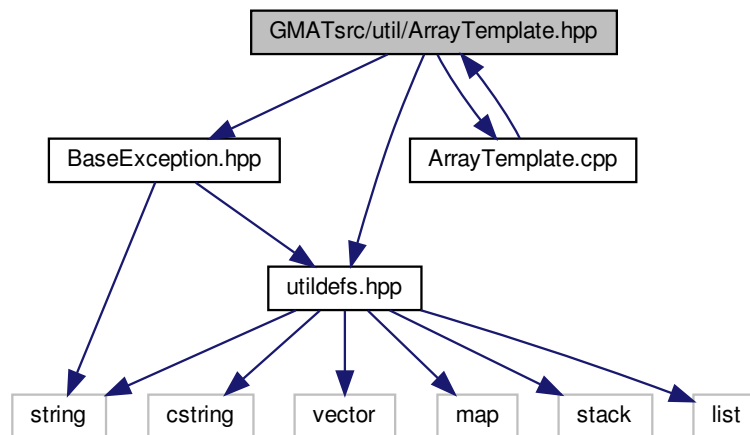
## 9.14 GMATsrc/util/ArrayTemplate.hpp File Reference

```
#include "utildefs.hpp"
```

```
#include "BaseException.hpp"
```

```
#include "ArrayTemplate.cpp"
```

Include dependency graph for ArrayTemplate.hpp:



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



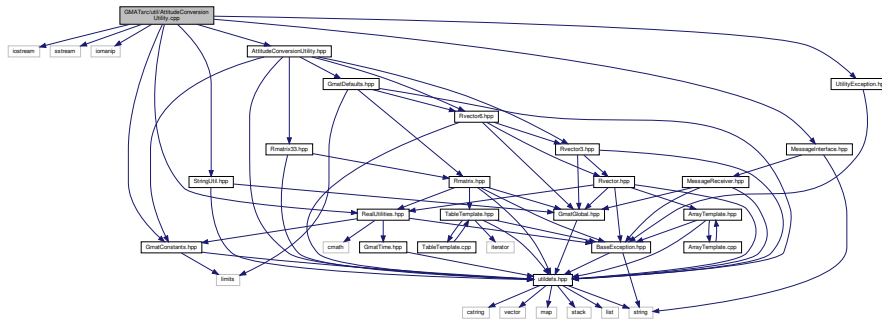
## Classes

- class [ArrayTemplateExceptions](#)
- class [ArrayTemplateExceptions::OutOfBounds](#)
- class [ArrayTemplateExceptions::DimensionError](#)
- class [ArrayTemplateExceptions::UnsizeArray](#)
- class [ArrayTemplateExceptions::ArrayAlreadySized](#)
- class [ArrayTemplateExceptions::IllegalSize](#)
- class [ArrayTemplate< T >](#)

## 9.15 GMATsrc/util/AttitudeConversionUtility.cpp File Reference

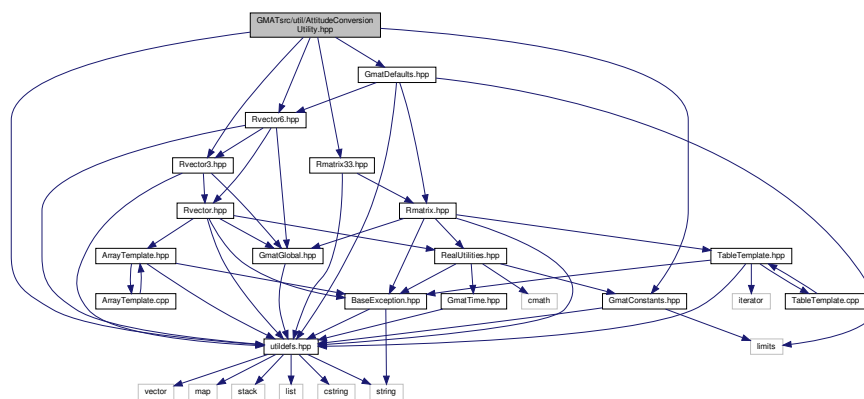
```
#include <iostream>
#include <sstream>
#include <iomanip>
#include "AttitudeConversionUtility.hpp"
#include "RealUtilities.hpp"
#include "MessageInterface.hpp"
#include "StringUtil.hpp"
#include "GmatConstants.hpp"
```

Include dependency graph for AttitudeConversionUtility.cpp:

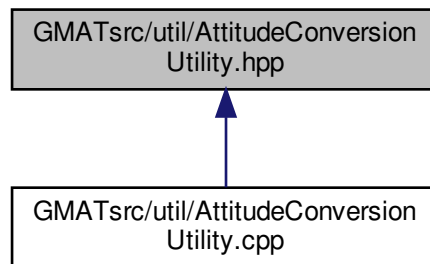


## 9.16 GMATsrc/util/AttitudeConversionUtility.hpp File Reference

Include dependency graph for AttitudeConversionUtility.hpp:



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



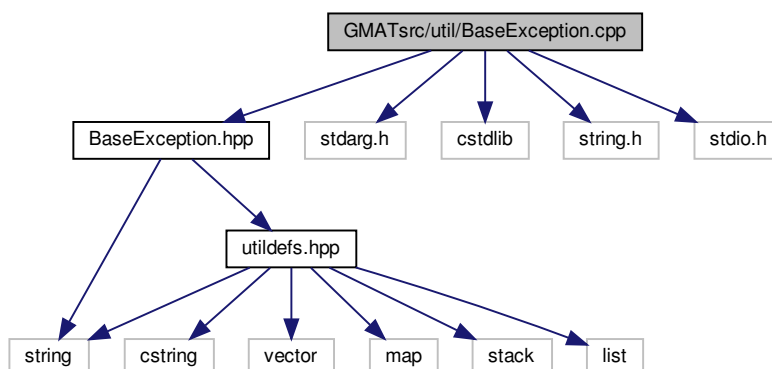
## Classes

- class [AttitudeConversionUtility](#)

## 9.17 GMATsrc/util/BaseException.cpp File Reference

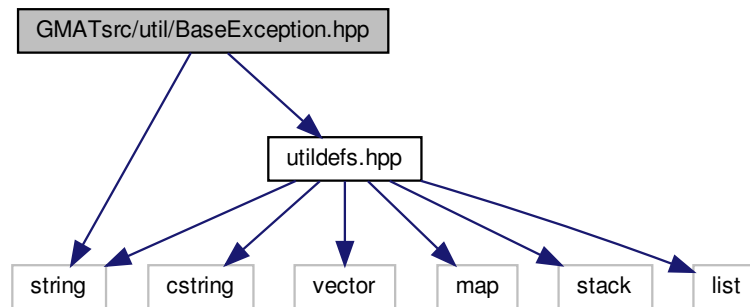
```
#include "BaseException.hpp"
#include <stdarg.h>
#include <cstdlib>
#include <string.h>
#include <stdio.h>
```

Include dependency graph for `BaseException.cpp`:



## 9.18 GMATsrc/util/BaseException.hpp File Reference

```
#include "utildefs.hpp"
#include <string>
Include dependency graph for BaseException.hpp:
```



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



### Classes

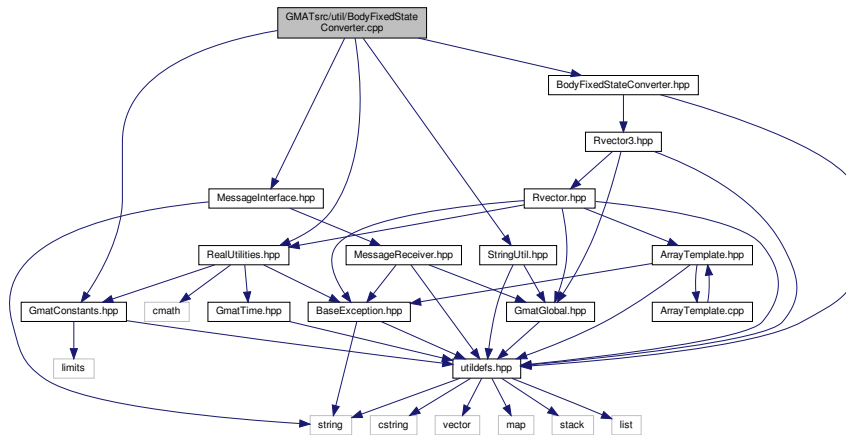
- class [BaseException](#)

## 9.19 GMATsrc/util/BodyFixedStateConverter.cpp File Reference

```
#include "BodyFixedStateConverter.hpp"
#include "StringUtil.hpp"
#include "GmatConstants.hpp"
#include "MessageInterface.hpp"
#include "RealUtilities.hpp"
```



Include dependency graph for BodyFixedStateConverter.cpp:

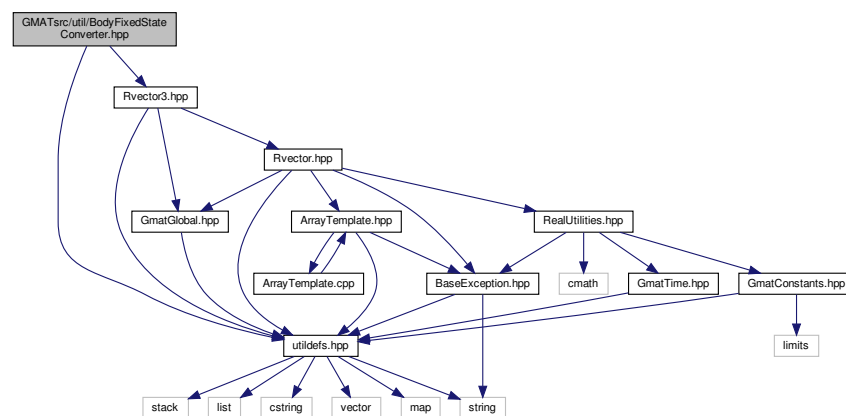


## 9.20 GMATsrc/util/BodyFixedStateConverter.hpp File Reference

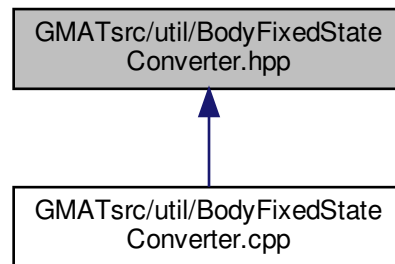
```
#include "utildefs.hpp"
```

```
#include "Rvector3.hpp"
```

Include dependency graph for BodyFixedStateConverter.hpp:



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



## Classes

- class [InvalidStateRepresentationException](#)

## Namespaces

- [BodyFixedStateConverterUtil](#)

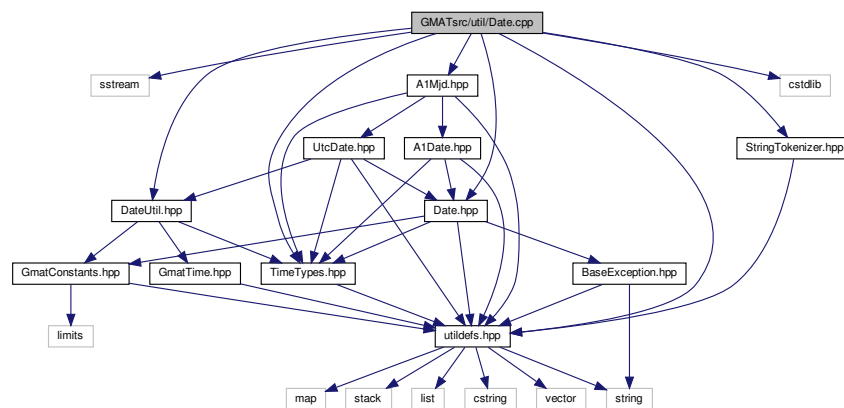
## Functions

- [Rvector3 GMATUTIL\\_API BodyFixedStateConverterUtil::Convert](#) (const [Rvector3](#) &origValue, const std::string &fromType, const std::string &toType, const [Real](#) flattening, const [Real](#) meanRadius)
- [Rvector3 GMATUTIL\\_API BodyFixedStateConverterUtil::Convert](#) (const [Rvector3](#) &origValue, const std::string &fromType, const std::string &fromHorizon, const std::string &toType, const std::string &toHorizon, const [Real](#) flattening, const [Real](#) meanRadius)
- [Rvector3 GMATUTIL\\_API BodyFixedStateConverterUtil::CartesianToSpherical](#) (const [Rvector3](#) &cart, const [Real](#) flattening, const [Real](#) meanRadius)
- [Rvector3 GMATUTIL\\_API BodyFixedStateConverterUtil::SphericalToCartesian](#) (const [Rvector3](#) &spherical, const [Real](#) flattening, const [Real](#) meanRadius)
- [Rvector3 GMATUTIL\\_API BodyFixedStateConverterUtil::SphericalEllipsoidToCartesian](#) (const [Rvector3](#) &sphEll, const [Real](#) flattening, const [Real](#) meanRadius)
- [Rvector3 GMATUTIL\\_API BodyFixedStateConverterUtil::CartesianToSphericalEllipsoid](#) (const [Rvector3](#) &cart, const [Real](#) flattening, const [Real](#) meanRadius)
- [Rvector3 GMATUTIL\\_API BodyFixedStateConverterUtil::SphericalToSphericalEllipsoid](#) (const [Rvector3](#) &spherical, const [Real](#) flattening, const [Real](#) meanRadius)
- [Rvector3 GMATUTIL\\_API BodyFixedStateConverterUtil::SphericalEllipsoidToSpherical](#) (const [Rvector3](#) &sphEll, const [Real](#) flattening, const [Real](#) meanRadius)
- [bool GMATUTIL\\_API BodyFixedStateConverterUtil::IsValidStateRepresentation](#) (const std::string &rep)
- [StringArray GMATUTIL\\_API BodyFixedStateConverterUtil::GetValidRepresentations](#) ()

## 9.21 GMATsrc/util/Date.cpp File Reference

```
#include <sstream>
#include "utildefs.hpp"
#include "TimeTypes.hpp"
#include "Date.hpp"
#include "A1Mjd.hpp"
#include "DateUtil.hpp"
#include "StringTokenizer.hpp"
#include <cstdlib>
```

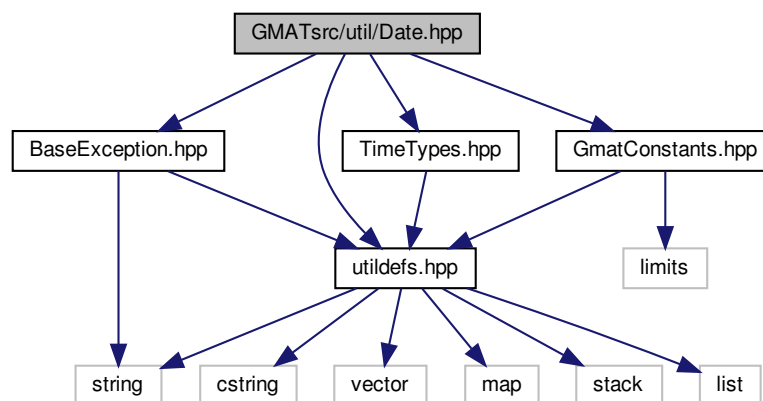
Include dependency graph for Date.cpp:



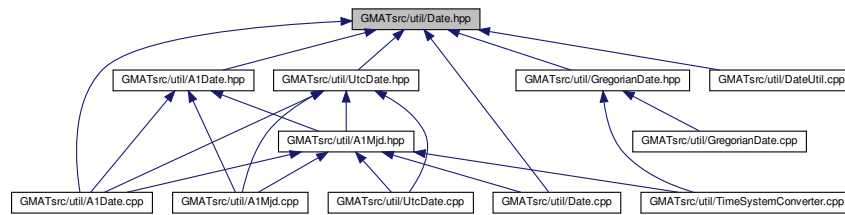
## 9.22 GMATsrc/util/Date.hpp File Reference

```
#include "utildefs.hpp"
#include "BaseException.hpp"
#include "GmatConstants.hpp"
#include "TimeTypes.hpp"
```

Include dependency graph for Date.hpp:



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



## Classes

- class [Date](#)
- class [Date::TimeRangeError](#)
- class [Date::LeapYearError](#)

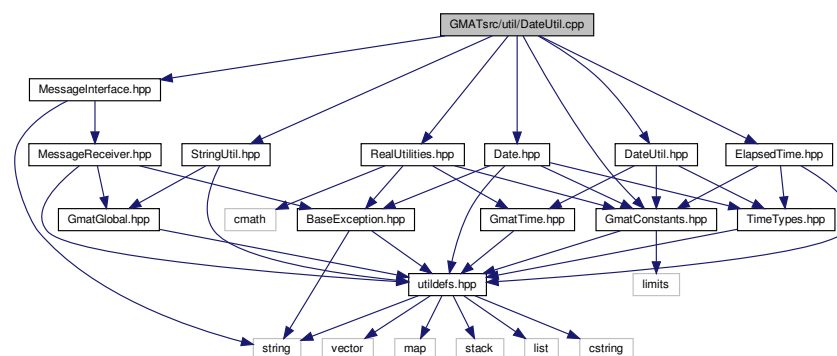
## 9.23 GMATsrc/util/DateUtil.cpp File Reference

```

#include "GmatConstants.hpp"
#include "Date.hpp"
#include "DateUtil.hpp"
#include "RealUtilities.hpp"
#include "StringUtil.hpp"
#include "ElapsedTime.hpp"
#include "MessageInterface.hpp"

```

Include dependency graph for DateUtil.cpp:



## Functions

- [Real JulianDate](#) ([YearNumber](#) year, [MonthOfYear](#) month, [DayOfMonth](#) day, [Integer](#) hour, [Integer](#) minute, [Real](#) second)
- [Real ModifiedJulianDate](#) ([YearNumber](#) year, [MonthOfYear](#) month, [DayOfMonth](#) day, [Integer](#) hour, [Integer](#) minute, [Real](#) second, [Real](#) refEpochJD)

- `GmatTime ModifiedJulianDateGT` (`YearNumber` year, `MonthOfYear` month, `DayOfMonth` day, `Integer` hour, `Integer` minute, `Real` second, `Real` refEpochJD)
- void `UnpackDate` (`Real` packedDate, `Integer` &year, `Integer` &month, `Integer` &day)
- void `UnpackDateWithDOY` (`Real` packedDate, `Integer` &year, `Integer` &day)
- void `UnpackTime` (`Real` packedTime, `Integer` &hour, `Integer` &minute, `Real` &second)
- void `ToMonthDayFromYearDOY` (`Integer` year, `Integer` dayOfYear, `Integer` &month, `Integer` &day)
- `Integer` `ToDOYFromYearMonthDay` (`Integer` year, `Integer` month, `Integer` day)
- `Real` `ToSecondsOfDayFromHMS` (`Integer` hour, `Integer` minute, `Real` second)
- void `ToHMSFromSecondsOfDay` (`Real` secsOfDay, `Integer` &hour, `Integer` &minute, `Real` &second)
- bool `IsValidTime` (`Integer` year, `Integer` month, `Integer` day, `Integer` hour, `Integer` minute, `Real` second)
- bool `IsLeapYear` (`Integer` year)

### 9.23.1 Function Documentation

#### 9.23.1.1 IsLeapYear()

```
bool IsLeapYear (
 Integer year)
```

Friend function.

#### 9.23.1.2 IsValidTime()

```
bool IsValidTime (
 Integer year,
 Integer month,
 Integer day,
 Integer hour,
 Integer minute,
 Real second)
```

Friend function.

#### 9.23.1.3 JulianDate()

```
Real JulianDate (
 YearNumber year,
 MonthOfYear month,
 DayOfMonth day,
 Integer hour,
 Integer minute,
 Real second)
```

Friend function. Converted from calendar date to Julian [Date](#).

#### Parameters

|          |                                   |
|----------|-----------------------------------|
| <year>   | - year of calendar                |
| <month>  | - month in calendar format        |
| <day>    | - day of month in calendar format |
| <hour>   | - hour of day                     |
| <minute> | - minute of hour                  |
| <second> | - seconds including millisecond   |

**Returns**

Julian date

**Note**

: The algorithm is used in the Vallado book.

**9.23.1.4 ModifiedJulianDate()**

```
Real ModifiedJulianDate (
 YearNumber year,
 MonthOfYear month,
 DayOfMonth day,
 Integer hour,
 Integer minute,
 Real second,
 Real refEpochJD)
```

Friend function. Converted from calendar date to Modified Julian [Date](#).

**Parameters**

|              |                                               |
|--------------|-----------------------------------------------|
| <year>       | - year of calendar                            |
| <month>      | - month in calendar format                    |
| <day>        | - day of month in calendar format             |
| <hour>       | - hour of day                                 |
| <minute>     | - minute of hour                              |
| <second>     | - seconds including millisecond               |
| <refEpochJD> | - reference epoch Julian <a href="#">Date</a> |

**Returns**

Modified Julian date

**9.23.1.5 ModifiedJulianDateGT()**

```
GmatTime ModifiedJulianDateGT (
 YearNumber year,
 MonthOfYear month,
 DayOfMonth day,
 Integer hour,
 Integer minute,
 Real second,
 Real refEpochJD)
```

Friend function. Converted from calendar date to Modified Julian [Date](#).

## Parameters

|              |                                               |
|--------------|-----------------------------------------------|
| <year>       | - year of calendar                            |
| <month>      | - month in calendar format                    |
| <day>        | - day of month in calendar format             |
| <hour>       | - hour of day                                 |
| <minute>     | - minute of hour                              |
| <second>     | - seconds including millisecond               |
| <refEpochJD> | - reference epoch Julian <a href="#">Date</a> |

## Returns

Modified Julian date

## 9.23.1.6 ToDOYFromYearMonthDay()

```
Integer ToDOYFromYearMonthDay (
 Integer year,
 Integer month,
 Integer day)
```

Friend function.

## Note

Year is needed to determine if it is a leap year

## 9.23.1.7 ToHMSFromSecondsOfDay()

```
void ToHMSFromSecondsOfDay (
 Real secsOfDay,
 Integer & hour,
 Integer & minute,
 Real & second)
```

Friend function.

Notes: Seconds are Real to permit fractions; seconds of day constrained to 0.0 .. 86401.0; the last second of a leap second day will be 23:59:60; assumes only 1 leap second per day maximum.

### 9.23.1.8 ToMonthDayFromYearDOY()

```
void ToMonthDayFromYearDOY (
 Integer year,
 Integer dayOfYear,
 Integer & month,
 Integer & day)
```

Friend function.

#### Note

Year is needed to determine if it is a leap year.

### 9.23.1.9 ToSecondsOfDayFromHMS()

```
Real ToSecondsOfDayFromHMS (
 Integer hour,
 Integer minute,
 Real second)
```

Friend function.

#### Note

Seconds of day constrained to 0.0..86401.0; assumes only 1 leap second per day maximum.

### 9.23.1.10 UnpackDate()

```
void UnpackDate (
 Real packedDate,
 Integer & year,
 Integer & month,
 Integer & day)
```

Friend function.

#### Note

Input date in YYYYMMDD



## 9.23.1.11 UnpackDateWithDOY()

```
void UnpackDateWithDOY (
 Real packedDate,
 Integer & year,
 Integer & day)
```

Friend function.

**Note**

Input date is in YYYYDDD; Day is rounded because of potential floating point representation problem. Do not pre-correct the input by adding 0.5.

## 9.23.1.12 UnpackTime()

```
void UnpackTime (
 Real packedTime,
 Integer & hour,
 Integer & minute,
 Real & second)
```

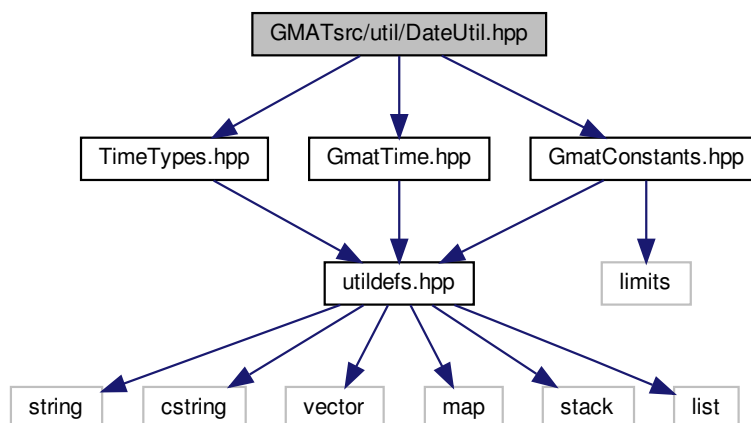
Friend function.

**Note**

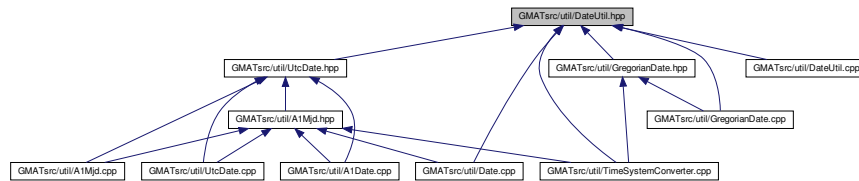
Input time is in hhmmssnnn; added 20 seconds to the input time, to be subtracted later, to avoid gross errors around minute boundaries.

## 9.24 GMATsrc/util/DateUtil.hpp File Reference

```
#include "TimeTypes.hpp"
#include "GmatConstants.hpp"
#include "GmatTime.hpp"
Include dependency graph for DateUtil.hpp:
```



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



## Classes

- class [DateUtil](#)

## Functions

- [Real](#) [GMATUTIL\\_API](#) [JulianDate](#) ([YearNumber](#) year, [MonthOfYear](#) month, [DayOfMonth](#) day, [Integer](#) hour, [Integer](#) minute, [Real](#) second)
- [Real](#) [GMATUTIL\\_API](#) [ModifiedJulianDate](#) ([YearNumber](#) year, [MonthOfYear](#) month, [DayOfMonth](#) day, [Integer](#) hour, [Integer](#) minute, [Real](#) second, [Real](#) refEpochJD=[GmatTimeConstants::JULIAN\\_DATE\\_OF\\_010541](#))
- [GmatTime](#) [GMATUTIL\\_API](#) [ModifiedJulianDateGT](#) ([YearNumber](#) year, [MonthOfYear](#) month, [DayOfMonth](#) day, [Integer](#) hour, [Integer](#) minute, [Real](#) second, [Real](#) refEpochJD=[GmatTimeConstants::JULIAN\\_DATE\\_OF\\_010541](#))
- void [GMATUTIL\\_API](#) [UnpackDate](#) ([Real](#) packedDate, [Integer](#) &year, [Integer](#) &month, [Integer](#) &day)
- void [GMATUTIL\\_API](#) [UnpackDateWithDOY](#) ([Real](#) packedDate, [Integer](#) &year, [Integer](#) &day)
- void [GMATUTIL\\_API](#) [UnpackTime](#) ([Real](#) packedTime, [Integer](#) &hour, [Integer](#) &minute, [Real](#) &second)
- void [GMATUTIL\\_API](#) [ToMonthDayFromYearDOY](#) ([Integer](#) year, [Integer](#) dayOfYear, [Integer](#) &month, [Integer](#) &day)
- [Integer](#) [GMATUTIL\\_API](#) [ToDOYFromYearMonthDay](#) ([Integer](#) year, [Integer](#) month, [Integer](#) day)
- [Real](#) [GMATUTIL\\_API](#) [ToSecondsOfDayFromHMS](#) ([Integer](#) hour, [Integer](#) minute, [Real](#) second)
- void [GMATUTIL\\_API](#) [ToHMSFromSecondsOfDay](#) ([Real](#) secsOfDay, [Integer](#) &hour, [Integer](#) &minute, [Real](#) &second)
- bool [GMATUTIL\\_API](#) [IsValidTime](#) ([Integer](#) year, [Integer](#) month, [Integer](#) day, [Integer](#) hour, [Integer](#) minute, [Real](#) second)
- bool [GMATUTIL\\_API](#) [IsLeapYear](#) ([Integer](#) year)

### 9.24.1 Function Documentation

#### 9.24.1.1 IsLeapYear()

```
bool GMATUTIL_API IsLeapYear (
 Integer year)
```

Friend function.

## 9.24.1.2 IsValidTime()

```
bool GMATUTIL_API IsValidTime (
 Integer year,
 Integer month,
 Integer day,
 Integer hour,
 Integer minute,
 Real second)
```

Friend function.

## 9.24.1.3 JulianDate()

```
Real GMATUTIL_API JulianDate (
 YearNumber year,
 MonthOfYear month,
 DayOfMonth day,
 Integer hour,
 Integer minute,
 Real second)
```

Friend function. Converted from calendar date to Julian [Date](#).

## Parameters

|          |                                   |
|----------|-----------------------------------|
| <year>   | - year of calendar                |
| <month>  | - month in calendar format        |
| <day>    | - day of month in calendar format |
| <hour>   | - hour of day                     |
| <minute> | - minute of hour                  |
| <second> | - seconds including millisecond   |

## Returns

Julian date

## Note

: The algorithm is used in the Vallado book.

## 9.24.1.4 ModifiedJulianDate()

```
Real GMATUTIL_API ModifiedJulianDate (
 YearNumber year,
 MonthOfYear month,
 DayOfMonth day,
 Integer hour,
 Integer minute,
 Real second,
 Real refEpochJD)
```

Friend function. Converted from calendar date to Modified Julian [Date](#).

**Parameters**

|              |                                               |
|--------------|-----------------------------------------------|
| <year>       | - year of calendar                            |
| <month>      | - month in calendar format                    |
| <day>        | - day of month in calendar format             |
| <hour>       | - hour of day                                 |
| <minute>     | - minute of hour                              |
| <second>     | - seconds including millisecond               |
| <refEpochJD> | - reference epoch Julian <a href="#">Date</a> |

**Returns**

Modified Julian date

**9.24.1.5 ModifiedJulianDateGT()**

```
GmatTime GMATUTIL_API ModifiedJulianDateGT (
 YearNumber year,
 MonthOfYear month,
 DayOfMonth day,
 Integer hour,
 Integer minute,
 Real second,
 Real refEpochJD)
```

Friend function. Converted from calendar date to Modified Julian [Date](#).

**Parameters**

|              |                                               |
|--------------|-----------------------------------------------|
| <year>       | - year of calendar                            |
| <month>      | - month in calendar format                    |
| <day>        | - day of month in calendar format             |
| <hour>       | - hour of day                                 |
| <minute>     | - minute of hour                              |
| <second>     | - seconds including millisecond               |
| <refEpochJD> | - reference epoch Julian <a href="#">Date</a> |

**Returns**

Modified Julian date

**9.24.1.6 ToDOYFromYearMonthDay()**

```
Integer GMATUTIL_API ToDOYFromYearMonthDay (
 Integer year,
```

```
Integer month,
Integer day)
```

Friend function.

#### Note

Year is needed to determine if it is a leap year

#### 9.24.1.7 ToHMSFromSecondsOfDay()

```
void GMATUTIL_API ToHMSFromSecondsOfDay (
 Real secsOfDay,
 Integer & hour,
 Integer & minute,
 Real & second)
```

Friend function.

Notes: Seconds are Real to permit fractions; seconds of day constrained to 0.0 .. 86401.0; the last second of a leap second day will be 23:59:60; assumes only 1 leap second per day maximum.

#### 9.24.1.8 ToMonthDayFromYearDOY()

```
void GMATUTIL_API ToMonthDayFromYearDOY (
 Integer year,
 Integer dayOfYear,
 Integer & month,
 Integer & day)
```

Friend function.

#### Note

Year is needed to determine if it is a leap year.

#### 9.24.1.9 ToSecondsOfDayFromHMS()

```
Real GMATUTIL_API ToSecondsOfDayFromHMS (
 Integer hour,
 Integer minute,
 Real second)
```

Friend function.

#### Note

Seconds of day constrained to 0.0..86401.0; assumes only 1 leap second per day maximum.

#### 9.24.1.10 UnpackDate()

```
void GMATUTIL_API UnpackDate (
 Real packedDate,
 Integer & year,
 Integer & month,
 Integer & day)
```

Friend function.

##### Note

Input date in YYYYMMDD

#### 9.24.1.11 UnpackDateWithDOY()

```
void GMATUTIL_API UnpackDateWithDOY (
 Real packedDate,
 Integer & year,
 Integer & day)
```

Friend function.

##### Note

Input date is in YYYYDDD; Day is rounded because of potential floating point representation problem. Do not pre-correct the input by adding 0.5.

#### 9.24.1.12 UnpackTime()

```
void GMATUTIL_API UnpackTime (
 Real packedTime,
 Integer & hour,
 Integer & minute,
 Real & second)
```

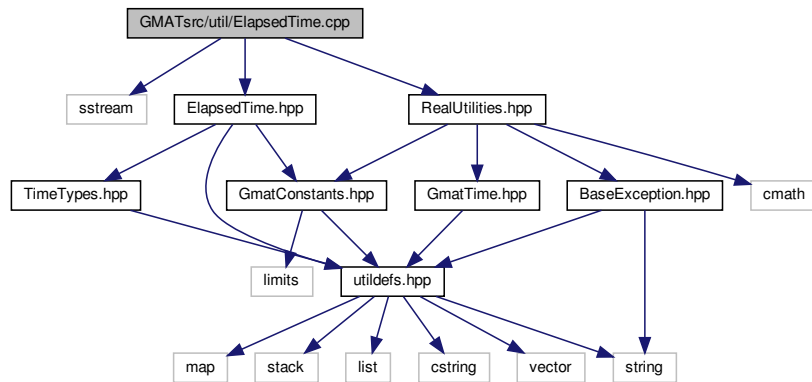
Friend function.

##### Note

Input time is in hhmmssnnn; added 20 seconds to the input time, to be subtracted later, to avoid gross errors around minute boundaries.

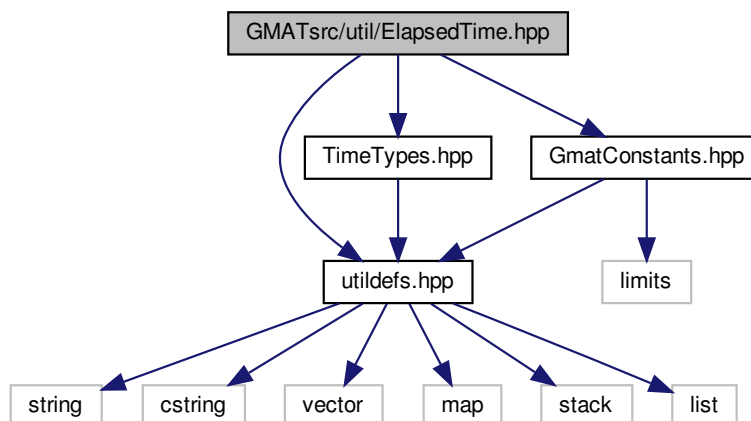
## 9.25 GMATsrc/util/ElapsedTime.cpp File Reference

```
#include <sstream>
#include "ElapsedTime.hpp"
#include "RealUtilities.hpp"
Include dependency graph for ElapsedTime.cpp:
```

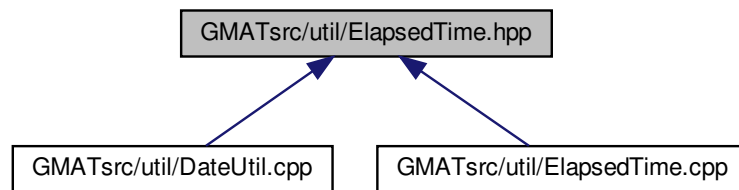


## 9.26 GMATsrc/util/ElapsedTime.hpp File Reference

```
#include "utildefs.hpp"
#include "TimeTypes.hpp"
#include "GmatConstants.hpp"
Include dependency graph for ElapsedTime.hpp:
```



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



## Classes

- class [ElapsedTime](#)

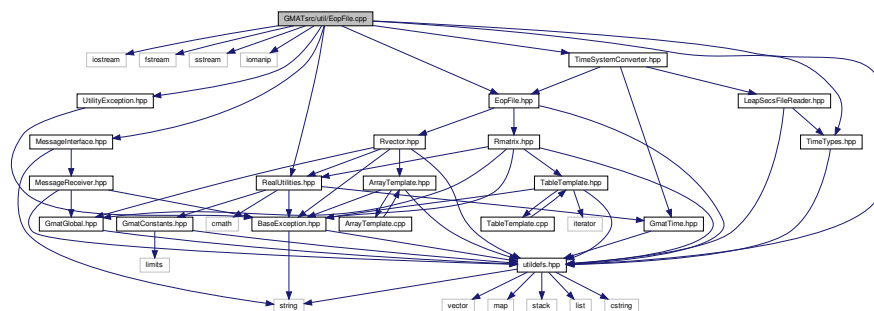
## 9.27 GMATsrc/util/EopFile.cpp File Reference

```

#include <iostream>
#include <fstream>
#include <sstream>
#include <iomanip>
#include "utildefs.hpp"
#include "EopFile.hpp"
#include "TimeTypes.hpp"
#include "UtilityException.hpp"
#include "RealUtilities.hpp"
#include "MessageInterface.hpp"
#include "TimeSystemConverter.hpp"

```

Include dependency graph for EopFile.cpp:



## 9.28 GMATsrc/util/EopFile.hpp File Reference

```

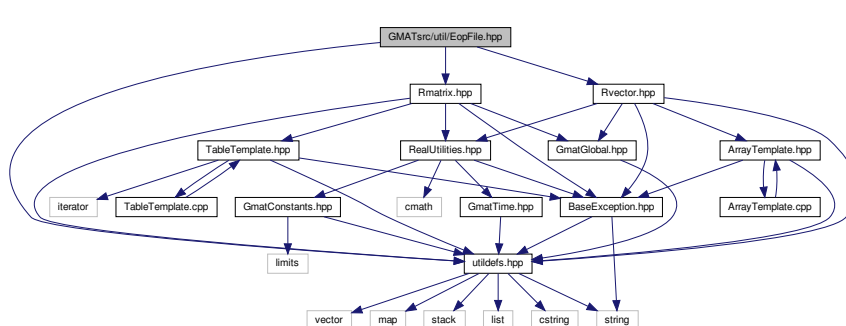
#include "utildefs.hpp"
#include "Rmatrix.hpp"

```

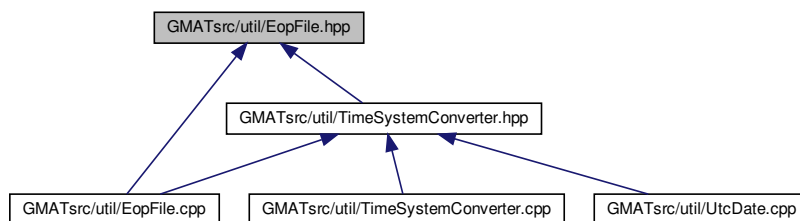


```
#include "Rvector.hpp"
```

Include dependency graph for EopFile.hpp:



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



## Classes

- class EopFile

## Namespaces

- GmatEop

## Enumerations

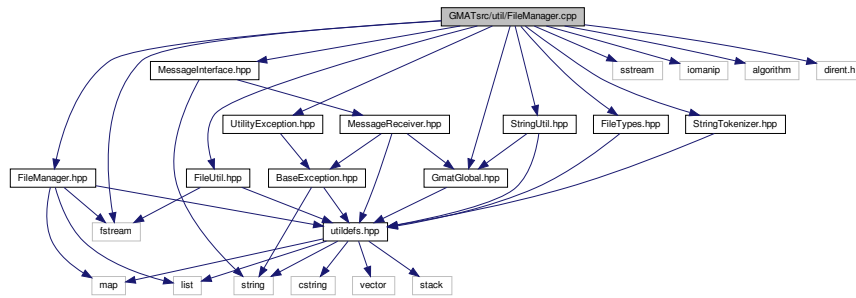
- enum GmatEop::EopFileType { GmatEop::EOP\_C04, GmatEop::FINALS }

## 9.29 GMATsrc/util/FileManager.cpp File Reference

```
#include "FileManager.hpp"
#include "MessageInterface.hpp"
#include "UtilityException.hpp"
#include "StringUtil.hpp"
#include "FileTypes.hpp"
```

```
#include "FileUtil.hpp"
#include "StringTokenizer.hpp"
#include "GmatGlobal.hpp"
#include <fstream>
#include <sstream>
#include <iomanip>
#include <algorithm>
#include <dirent.h>
```

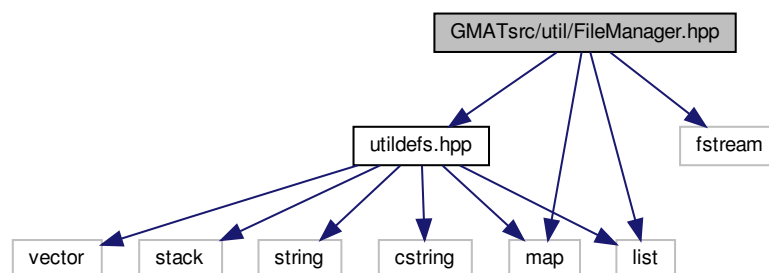
Include dependency graph for FileManager.cpp:



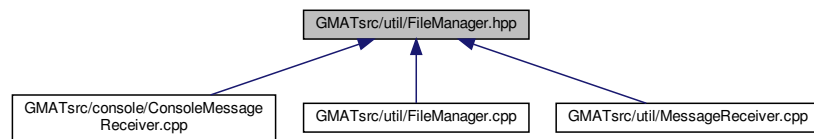
### 9.30 GMATsrc/util/FileManager.hpp File Reference

```
#include "utildefs.hpp"
#include <map>
#include <list>
#include <fstream>
```

Include dependency graph for FileManager.hpp:



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



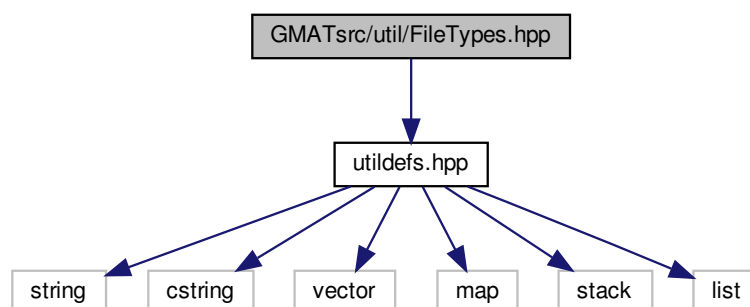
## Classes

- class [FileManager](#)

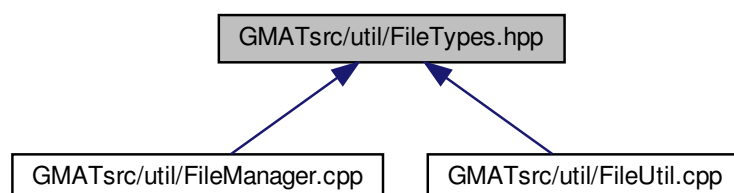
## 9.31 GMATsrc/util/FileTypes.hpp File Reference

```
#include "utildefs.hpp"
```

Include dependency graph for `FileTypes.hpp`:



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



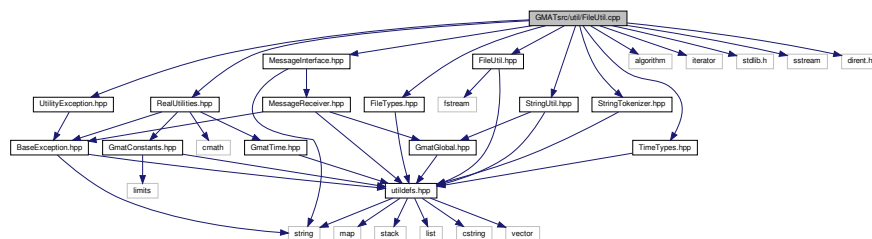
## Namespaces

- [GmatFile](#)

### 9.32 GMATsrc/util/FileUtil.cpp File Reference

```
#include "FileUtil.hpp"
#include "StringTokenizer.hpp"
#include "MessageInterface.hpp"
#include "TimeTypes.hpp"
#include "RealUtilities.hpp"
#include "StringUtil.hpp"
#include "FileTypes.hpp"
#include "UtilityException.hpp"
#include <algorithm>
#include <iterator>
#include <stdlib.h>
#include <sstream>
#include <dirent.h>
```

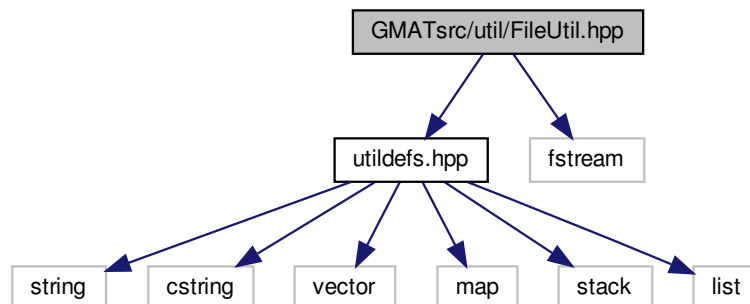
Include dependency graph for FileUtil.cpp:



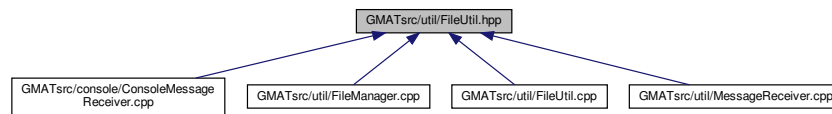
### 9.33 GMATsrc/util/FileUtil.hpp File Reference

```
#include "utildefs.hpp"
#include <fstream>
```

Include dependency graph for FileUtil.hpp:



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



## Namespaces

- [GmatFileUtil](#)

## Functions

- `std::string GMATUTIL_API GmatFileUtil::GetPathSeparator ()`
- `std::string GMATUTIL_API GmatFileUtil::ConvertToOsFileName (const std::string &fileName)`
- `std::string GMATUTIL_API GmatFileUtil::GetCurrentWorkingDirectory ()`
- `bool GMATUTIL_API GmatFileUtil::SetCurrentWorkingDirectory (const std::string &newDir)`
- `std::string GMATUTIL_API GmatFileUtil::GetApplicationPath ()`
- `std::string GMATUTIL_API GmatFileUtil::GetTemporaryDirectory ()`
- `std::string GMATUTIL_API GmatFileUtil::ParseFirstPathName (const std::string &fullPath, bool appendSep=true)`
- `std::string GMATUTIL_API GmatFileUtil::ParsePathName (const char *fullPath, bool appendSep=true)`
- `std::string GMATUTIL_API GmatFileUtil::ParsePathName (const std::string &fullPath, bool appendSep=true)`
- `std::string GMATUTIL_API GmatFileUtil::ParseFileName (const char *fullPath, bool removeExt=false)`
- `std::string GMATUTIL_API GmatFileUtil::ParseFileName (const std::string &fullPath, bool removeExt=false)`
- `std::string GMATUTIL_API GmatFileUtil::ParseFileExtension (const char *fullPath, bool prependDot=false)`
- `std::string GMATUTIL_API GmatFileUtil::ParseFileExtension (const std::string &fullPath, bool prependDot=false)`
- `std::string GMATUTIL_API GmatFileUtil::GetInvalidFileNameMessage (Integer option=1)`
- `bool GMATUTIL_API GmatFileUtil::IsOsWindows ()`
- `bool GMATUTIL_API GmatFileUtil::IsPathRelative (const char *fullPath)`
- `bool GMATUTIL_API GmatFileUtil::IsPathRelative (const std::string &fullPath)`
- `bool GMATUTIL_API GmatFileUtil::IsPathAbsolute (const std::string &fullPath)`
- `bool GMATUTIL_API GmatFileUtil::HasNoPath (const std::string &fullPath)`
- `bool GMATUTIL_API GmatFileUtil::IsValidFileName (const std::string &fname, bool isBlankOk=true)`
- `bool GMATUTIL_API GmatFileUtil::IsSameFileName (const char *fname1, const char *fname2)`
- `bool GMATUTIL_API GmatFileUtil::IsSameFileName (const std::string &fname1, const std::string &fname2)`
- `bool GMATUTIL_API GmatFileUtil::DoesDirectoryExist (const char *dirPath, bool isBlankOk=true)`
- `bool GMATUTIL_API GmatFileUtil::DoesDirectoryExist (const std::string &dirPath, bool isBlankOk=true)`
- `bool GMATUTIL_API GmatFileUtil::DoesFileExist (const char *filename)`
- `bool GMATUTIL_API GmatFileUtil::DoesFileExist (const std::string &filename)`
- `bool GMATUTIL_API GmatFileUtil::GetLine (std::istream *inStream, std::string &line)`
- `bool GMATUTIL_API GmatFileUtil::IsApplInstalled (const std::string &appName, std::string &appLoc)`
- `std::string GmatFileUtil::GetGmatPath ()`
- `WrapperTypeArray GMATUTIL_API GmatFileUtil::GetFunctionOutputTypes (std::istream *is, const StringArray &inputs, const StringArray &outputs, std::string &errMsg, IntegerArray &outputRows, IntegerArray &outputCols)`
- `StringArray GMATUTIL_API GmatFileUtil::GetFileListFromDirectory (const std::string &dirName, bool addPath=false)`
- `StringArray GMATUTIL_API GmatFileUtil::GetTextLines (const std::string &fileName)`

- bool [GmatFileUtil::PrepareCompare](#) (Integer numDirsToCompare, const std::string &basefilename, const std::string &filename1, const std::string &filename2, const std::string &filename3, std::ifstream &baseIn, std::ifstream &in1, std::ifstream &in2, std::ifstream &in3)
- bool [GMATUTIL\\_API GmatFileUtil::CompareLines](#) (const std::string &line1, const std::string &line2, Real &diff, Real tol=COMPARE\_TOLERANCE)
- [StringArray GMATUTIL\\_API & GmatFileUtil::CompareTextLines](#) (Integer numDirsToCompare, const char \*basefilename, const char \*filename1, const char \*filename2, const char \*filename3, int &file1DiffCount, int &file2DiffCount, int &file3DiffCount, bool skipBlankLines=false)
- [StringArray GMATUTIL\\_API & GmatFileUtil::CompareTextLines](#) (Integer numDirsToCompare, const std::string &basefilename, const std::string &filename1, const std::string &filename2, const std::string &filename3, int &file1DiffCount, int &file2DiffCount, int &file3DiffCount, bool skipBlankLines=false)
- [StringArray GMATUTIL\\_API & GmatFileUtil::CompareNumericLines](#) (Integer numDirsToCompare, const char \*basefilename, const char \*filename1, const char \*filename2, const char \*filename3, int &file1DiffCount, int &file2DiffCount, int &file3DiffCount, Real tol=COMPARE\_TOLERANCE)
- [StringArray GMATUTIL\\_API & GmatFileUtil::CompareNumericLines](#) (Integer numDirsToCompare, const std::string &basefilename, const std::string &filename1, const std::string &filename2, const std::string &filename3, int &file1DiffCount, int &file2DiffCount, int &file3DiffCount, Real tol=COMPARE\_TOLERANCE)
- [StringArray GMATUTIL\\_API & GmatFileUtil::CompareNumericColumns](#) (Integer numDirsToCompare, const char \*basefilename, const char \*filename1, const char \*filename2, const char \*filename3, Real tol=COMPARE\_TOLERANCE)
- [StringArray GMATUTIL\\_API & GmatFileUtil::CompareNumericColumns](#) (Integer numDirsToCompare, const std::string &basefilename, const std::string &filename1, const std::string &filename2, const std::string &filename3, Real tol=COMPARE\_TOLERANCE)
- bool [GMATUTIL\\_API GmatFileUtil::SkipHeaderLines](#) (std::ifstream &in, RealArray &realArray, const std::string &filename)
- bool [GMATUTIL\\_API GmatFileUtil::IsAsciiFile](#) (std::ifstream &file, const std::string &filename)
- bool [GMATUTIL\\_API GmatFileUtil::GetRealColumns](#) (const std::string &line, RealArray &cols)

## Variables

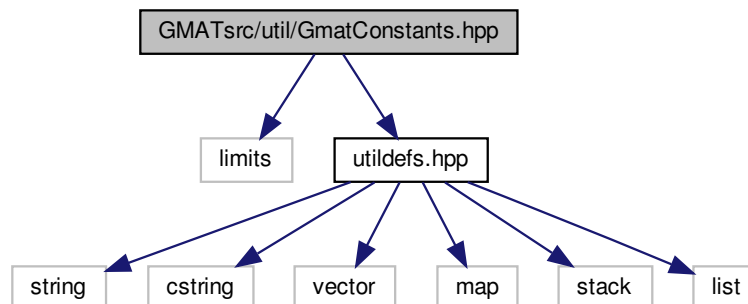
- const Integer [GmatFileUtil::BUFFER\\_SIZE](#) = 4096

## 9.34 GMATsrc/util/GmatConstants.hpp File Reference

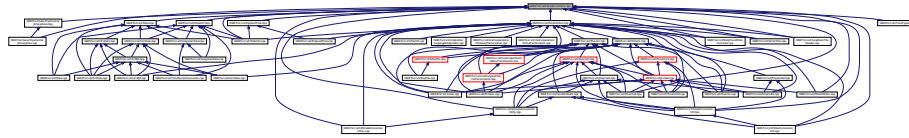
```
#include <limits>
```

```
#include "utildefs.hpp"
```

Include dependency graph for GmatConstants.hpp:



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



## Namespaces

- [GmatRealConstants](#)
- [GmatPhysicalConstants](#)
- [GmatIntegerConstants](#)
- [GmatTimeConstants](#)
- [GmatMathConstants](#)
- [GmatOrbitConstants](#)
- [GmatAttitudeConstants](#)

## Enumerations

- `enum GmatTimeConstants::DayName {  
GmatTimeConstants::SUNDAY, GmatTimeConstants::MONDAY, GmatTimeConstants::TUESDAY, GmatTimeConstants::WEDNESDAY,  
GmatTimeConstants::THURSDAY, GmatTimeConstants::FRIDAY, GmatTimeConstants::SATURDAY }`
- `enum GmatTimeConstants::MonthName {  
GmatTimeConstants::JANUARY = 1, GmatTimeConstants::FEBRUARY, GmatTimeConstants::MARCH,  
GmatTimeConstants::APRIL,  
GmatTimeConstants::MAY, GmatTimeConstants::JUNE, GmatTimeConstants::JULY, GmatTimeConstants::AUGUST,  
GmatTimeConstants::SEPTEMBER, GmatTimeConstants::OCTOBER, GmatTimeConstants::NOVEMBER,  
GmatTimeConstants::DECEMBER }`
- `enum GmatMathConstants::SIGN { GmatMathConstants::PLUS = 1, GmatMathConstants::MINUS = -1 }`

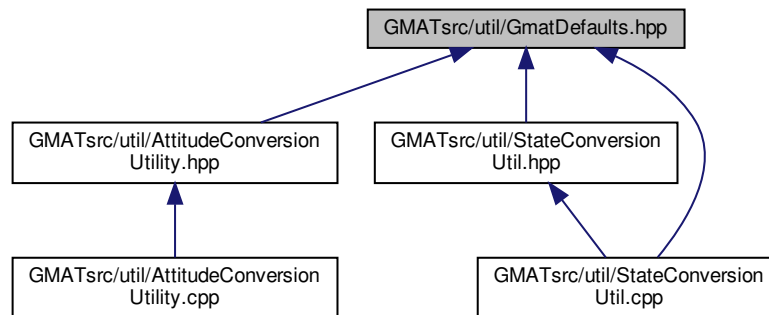
## Variables

- `const Real GmatRealConstants::REAL_TOL = 1.0e-15`
- `const Real GmatRealConstants::REAL_UNDEFINED = -987654321.0123e-45`
- `const Real GmatRealConstants::REAL_UNDEFINED_LARGE = -9876543210.1234`
- `const Integer GmatRealConstants::INTEGER_UNDEFINED = -987654321`
- `const Integer GmatRealConstants::INTEGER_MAX = std::numeric_limits<Integer>::max()`
- `const Integer GmatRealConstants::SHORT_REAL_RADIX = 2`
- `const Real GmatRealConstants::REAL_EPSILON = 2.2204460492503131e-16`
- `const Integer GmatRealConstants::REAL_DIG = 53`
- `const Integer GmatRealConstants::REAL_MIN_EXP = -1021`
- `const Real GmatRealConstants::REAL_MIN = 2.2250738585072014e-308`
- `const Integer GmatRealConstants::REAL_MIN_10_EXP = -307`
- `const Integer GmatRealConstants::REAL_MAX_EXP = 1024`
- `const Real GmatRealConstants::REAL_MAX = 1.7976931348623157e+308`
- `const Integer GmatRealConstants::REAL_MAX_10_EXP = 308`
- `const Real GmatPhysicalConstants::SPEED_OF_LIGHT_VACUUM = 299792458.0`
- `const Real GmatPhysicalConstants::c = 299792458.0`
- `const Real GmatPhysicalConstants::UNIVERSAL_GRAVITATIONAL_CONSTANT = 6.673e-20`





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



## Namespaces

- [GmatSolarSystemDefaults](#)

## Enumerations

- enum [GmatSolarSystemDefaults::DefaultPlanets](#) { [GmatSolarSystemDefaults::MERCURY](#) = 0, [GmatSolarSystemDefaults::VENUS](#), [GmatSolarSystemDefaults::EARTH](#), [GmatSolarSystemDefaults::MARS](#), [GmatSolarSystemDefaults::JUPITER](#), [GmatSolarSystemDefaults::SATURN](#), [GmatSolarSystemDefaults::URANUS](#), [GmatSolarSystemDefaults::NEPTUNE](#), [GmatSolarSystemDefaults::PLUTO](#), [GmatSolarSystemDefaults::NumberOfDefaultPlanets](#) }  
*Default planet data ----- planets -----.*
- enum [GmatSolarSystemDefaults::DefaultMoons](#) { [GmatSolarSystemDefaults::LUNA](#) = 0, [GmatSolarSystemDefaults::NumberOfDefaultMoons](#) }  
*Default planet data ----- moons -----.*

## Variables

- const std::string [GmatSolarSystemDefaults::SOLAR\\_SYSTEM\\_BARYCENTER\\_NAME](#) = "SolarSystemBarycenter"  
*default names for each of the possible celestial bodies in the solar system*
- const std::string [GmatSolarSystemDefaults::SUN\\_NAME](#) = "Sun"
- const std::string [GmatSolarSystemDefaults::MERCURY\\_NAME](#) = "Mercury"
- const std::string [GmatSolarSystemDefaults::VENUS\\_NAME](#) = "Venus"
- const std::string [GmatSolarSystemDefaults::EARTH\\_NAME](#) = "Earth"
- const std::string [GmatSolarSystemDefaults::MOON\\_NAME](#) = "Luna"
- const std::string [GmatSolarSystemDefaults::MARS\\_NAME](#) = "Mars"
- const std::string [GmatSolarSystemDefaults::PHOBOS\\_NAME](#) = "Phobos"
- const std::string [GmatSolarSystemDefaults::DEIMOS\\_NAME](#) = "Deimos"
- const std::string [GmatSolarSystemDefaults::JUPITER\\_NAME](#) = "Jupiter"
- const std::string [GmatSolarSystemDefaults::METIS\\_NAME](#) = "Metis"
- const std::string [GmatSolarSystemDefaults::ADRASTEIA\\_NAME](#) = "Adrastea"
- const std::string [GmatSolarSystemDefaults::AMALTHEA\\_NAME](#) = "Amalthea"

- const std::string GmatSolarSystemDefaults::THEBE\_NAME = "Thebe"
- const std::string GmatSolarSystemDefaults::IO\_NAME = "Io"
- const std::string GmatSolarSystemDefaults::EUROPA\_NAME = "Europa"
- const std::string GmatSolarSystemDefaults::GANYMEDE\_NAME = "Ganymede"
- const std::string GmatSolarSystemDefaults::CALLISTO\_NAME = "Callisto"
- const std::string GmatSolarSystemDefaults::SATURN\_NAME = "Saturn"
- const std::string GmatSolarSystemDefaults::PAN\_NAME = "Pan"
- const std::string GmatSolarSystemDefaults::ATLAS\_NAME = "Atlas"
- const std::string GmatSolarSystemDefaults::PROMETHEUS\_NAME = "Prometheus"
- const std::string GmatSolarSystemDefaults::PANDORA\_NAME = "Pandora"
- const std::string GmatSolarSystemDefaults::EPIMETHEUS\_NAME = "Epimetheus"
- const std::string GmatSolarSystemDefaults::JANUS\_NAME = "Janus"
- const std::string GmatSolarSystemDefaults::MIMAS\_NAME = "Mimas"
- const std::string GmatSolarSystemDefaults::ENCELADUS\_NAME = "Enceladus"
- const std::string GmatSolarSystemDefaults::TETHYS\_NAME = "Tethys"
- const std::string GmatSolarSystemDefaults::TELESTO\_NAME = "Telesto"
- const std::string GmatSolarSystemDefaults::CALYPSO\_NAME = "Calypso"
- const std::string GmatSolarSystemDefaults::DIONE\_NAME = "Dione"
- const std::string GmatSolarSystemDefaults::HELENE\_NAME = "Helene"
- const std::string GmatSolarSystemDefaults::RHEA\_NAME = "Rhea"
- const std::string GmatSolarSystemDefaults::TITAN\_NAME = "Titan"
- const std::string GmatSolarSystemDefaults::IAPETUS\_NAME = "Iapetus"
- const std::string GmatSolarSystemDefaults::PHOEBE\_NAME = "Phoebe"
- const std::string GmatSolarSystemDefaults::URANUS\_NAME = "Uranus"
- const std::string GmatSolarSystemDefaults::CORDELIA\_NAME = "Cordelia"
- const std::string GmatSolarSystemDefaults::OPHELIA\_NAME = "Ophelia"
- const std::string GmatSolarSystemDefaults::BIANCA\_NAME = "Bianca"
- const std::string GmatSolarSystemDefaults::CRESSIDA\_NAME = "Cressida"
- const std::string GmatSolarSystemDefaults::DESDEMONA\_NAME = "Desdemona"
- const std::string GmatSolarSystemDefaults::JULIET\_NAME = "Juliet"
- const std::string GmatSolarSystemDefaults::PORTIA\_NAME = "Portia"
- const std::string GmatSolarSystemDefaults::ROSALIND\_NAME = "Rosalind"
- const std::string GmatSolarSystemDefaults::BELINDA\_NAME = "Belinda"
- const std::string GmatSolarSystemDefaults::PUCK\_NAME = "Puck"
- const std::string GmatSolarSystemDefaults::MIRANDA\_NAME = "Miranda"
- const std::string GmatSolarSystemDefaults::ARIEL\_NAME = "Ariel"
- const std::string GmatSolarSystemDefaults::UMBRIEL\_NAME = "Umbriel"
- const std::string GmatSolarSystemDefaults::TITANIA\_NAME = "Titania"
- const std::string GmatSolarSystemDefaults::OBERON\_NAME = "Oberon"
- const std::string GmatSolarSystemDefaults::NEPTUNE\_NAME = "Neptune"
- const std::string GmatSolarSystemDefaults::NAIAD\_NAME = "Naiad"
- const std::string GmatSolarSystemDefaults::THALASSA\_NAME = "Thalassa"
- const std::string GmatSolarSystemDefaults::DESPINA\_NAME = "Despina"
- const std::string GmatSolarSystemDefaults::GALATEA\_NAME = "Galatea"
- const std::string GmatSolarSystemDefaults::LARISSA\_NAME = "Larissa"
- const std::string GmatSolarSystemDefaults::PROTEUS\_NAME = "Proteus"
- const std::string GmatSolarSystemDefaults::TRITON\_NAME = "Triton"
- const std::string GmatSolarSystemDefaults::PLUTO\_NAME = "Pluto"
- const std::string GmatSolarSystemDefaults::CHARON\_NAME = "Charon"
- const Integer GmatSolarSystemDefaults::SSB\_NAIF\_ID = 0

*Default barycenter data.*

- const Real GmatSolarSystemDefaults::SSB\_MU = 0.0
- const std::string GmatSolarSystemDefaults::PLANET\_NAMES [NumberOfDefaultPlanets]

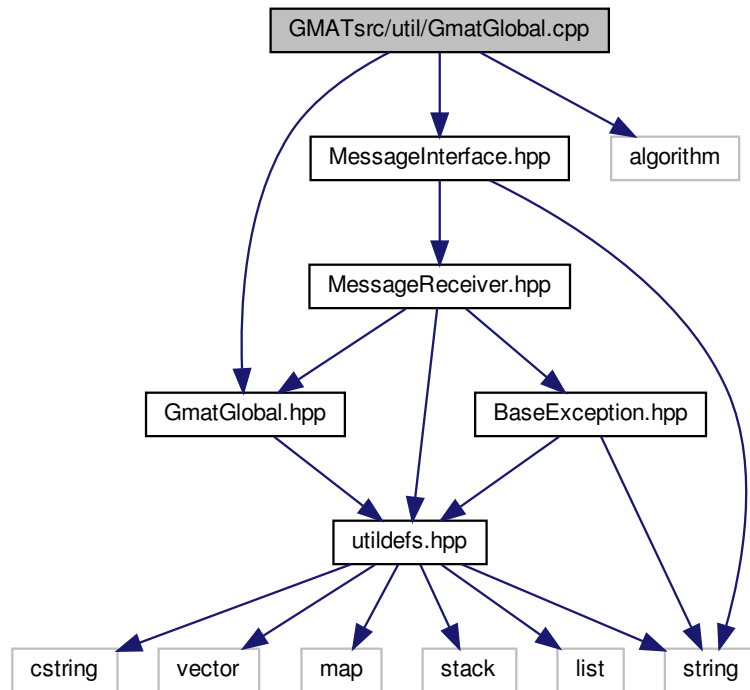
*Default planet data.*

- const Real GmatSolarSystemDefaults::PLANET\_EQUATORIAL\_RADIUS [NumberOfDefaultPlanets]
  - const Real GmatSolarSystemDefaults::PLANET\_FLATTENING [NumberOfDefaultPlanets]
  - const Real GmatSolarSystemDefaults::PLANET\_MU [NumberOfDefaultPlanets]
  - const Real GmatSolarSystemDefaults::PLANET\_TWO\_BODY\_EPOCH [NumberOfDefaultPlanets]
  - const Rvector6 GmatSolarSystemDefaults::PLANET\_TWO\_BODY\_ELEMENTS [NumberOfDefaultPlanets]
  - const Rvector6 GmatSolarSystemDefaults::PLANET\_ORIENTATION\_PARAMETERS [NumberOfDefaultPlanets]
  - const Integer GmatSolarSystemDefaults::PLANET\_NAIF\_IDS [NumberOfDefaultPlanets]
  - const std::string GmatSolarSystemDefaults::PLANET\_SPICE\_FRAME\_ID [NumberOfDefaultPlanets]
  - const std::string GmatSolarSystemDefaults::MOON\_NAMES [NumberOfDefaultMoons]
  - const std::string GmatSolarSystemDefaults::MOON\_CENTRAL\_BODIES [NumberOfDefaultMoons]
  - const Real GmatSolarSystemDefaults::MOON\_EQUATORIAL\_RADIUS [NumberOfDefaultMoons]
  - const Real GmatSolarSystemDefaults::MOON\_FLATTENING [NumberOfDefaultMoons]
  - const Real GmatSolarSystemDefaults::MOON\_MU [NumberOfDefaultMoons]
  - const Real GmatSolarSystemDefaults::MOON\_TWO\_BODY\_EPOCH [NumberOfDefaultMoons]
  - const Rvector6 GmatSolarSystemDefaults::MOON\_TWO\_BODY\_ELEMENTS [NumberOfDefaultMoons]
  - const Rvector6 GmatSolarSystemDefaults::MOON\_ORIENTATION\_PARAMETERS [NumberOfDefaultMoons]
  - const Integer GmatSolarSystemDefaults::MOON\_NAIF\_IDS [NumberOfDefaultMoons]
  - const std::string GmatSolarSystemDefaults::MOON\_SPICE\_FRAME\_ID [NumberOfDefaultMoons]
  - const Real GmatSolarSystemDefaults::STAR\_EQUATORIAL\_RADIUS = 695990.0000
- Default star data ----- the Sun -----.*
- const Real GmatSolarSystemDefaults::STAR\_FLATTENING = 0.0
  - const Real GmatSolarSystemDefaults::STAR\_MU = 132712440017.99
  - const Real GmatSolarSystemDefaults::STAR\_TWO\_BODY\_EPOCH = 21544.500370768266
  - const Rvector6 GmatSolarSystemDefaults::STAR\_TWO\_BODY\_ELEMENTS
  - const Rvector6 GmatSolarSystemDefaults::STAR\_ORIENTATION\_PARAMETERS
  - const Integer GmatSolarSystemDefaults::STAR\_NAIF\_IDS = 10
  - const std::string GmatSolarSystemDefaults::STAR\_SPICE\_FRAME\_ID = "IAU\_SUN"
  - const Real GmatSolarSystemDefaults::STAR\_RADIANT\_POWER = 1358.0
  - const Real GmatSolarSystemDefaults::STAR\_REFERENCE\_DISTANCE = GmatPhysicalConstants::ASTRONOMICAL\_UNIT
  - const Real GmatSolarSystemDefaults::STAR\_PHOTOSPHERE\_RADIUS = 695990000.0

## 9.36 GMATsrc/util/GmatGlobal.cpp File Reference

```
#include "GmatGlobal.hpp"
#include "MessageInterface.hpp"
#include <algorithm>
```

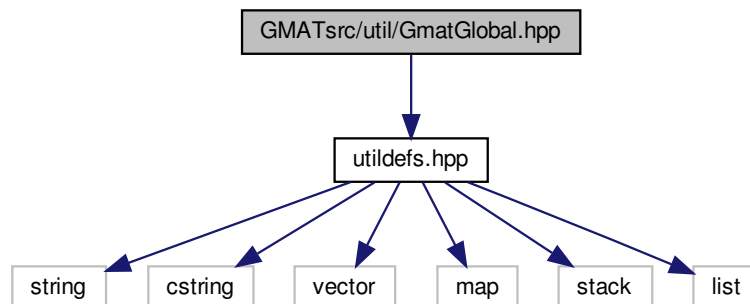
Include dependency graph for GmatGlobal.cpp:



### 9.37 GMATsrc/util/GmatGlobal.hpp File Reference

```
#include "utildefs.hpp"
```

Include dependency graph for GmatGlobal.hpp:



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



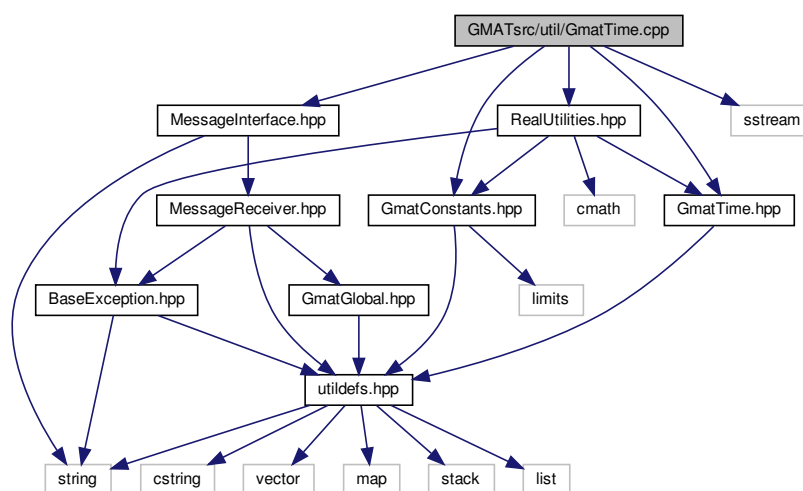
## Classes

- class [GmatGlobal](#)

## 9.38 GMATsrc/util/GmatTime.cpp File Reference

```
#include "GmatTime.hpp"
#include "GmatConstants.hpp"
#include "RealUtilities.hpp"
#include "MessageInterface.hpp"
#include <sstream>
```

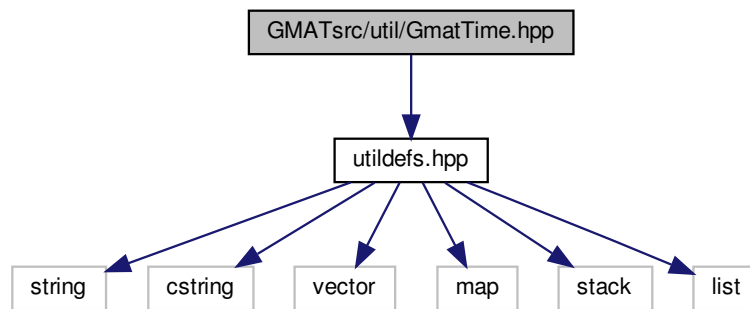
Include dependency graph for GmatTime.cpp:



## 9.39 GMATsrc/util/GmatTime.hpp File Reference

```
#include "utildefs.hpp"
```

Include dependency graph for GmatTime.hpp:



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



## Classes

- class [GmatTime](#)

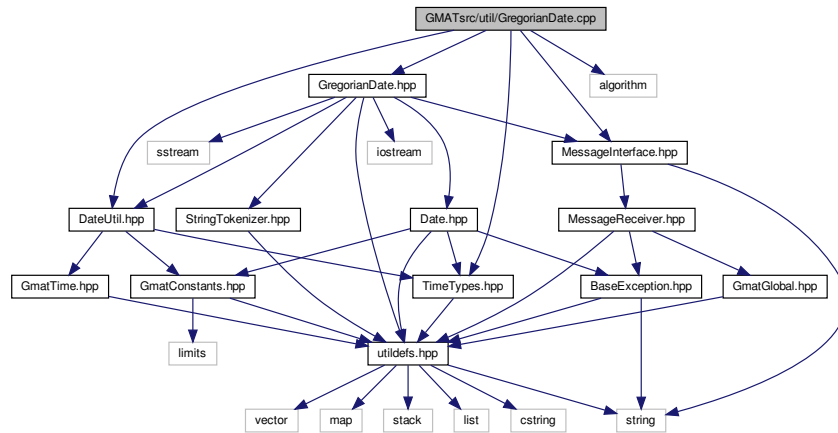
## 9.40 GMATsrc/util/GregorianCalendar.cpp File Reference

```

#include "GregorianCalendar.hpp"
#include "TimeTypes.hpp"
#include "DateUtil.hpp"
#include "MessageInterface.hpp"
#include <algorithm>

```

Include dependency graph for GregorianCalendarDate.cpp:



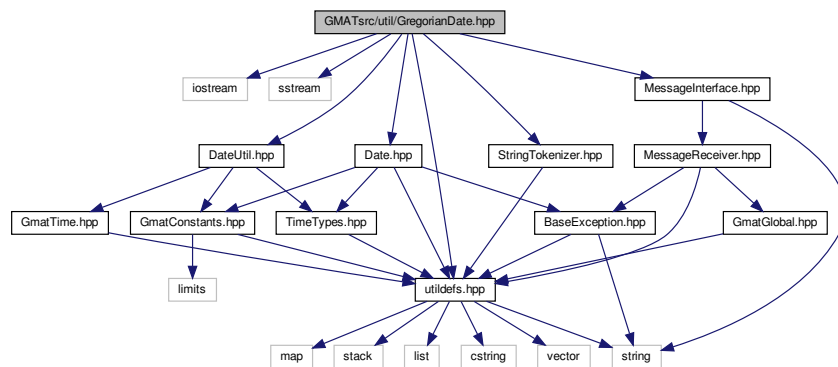
## 9.41 GMATsrc/util/GregorianCalendarDate.hpp File Reference

```

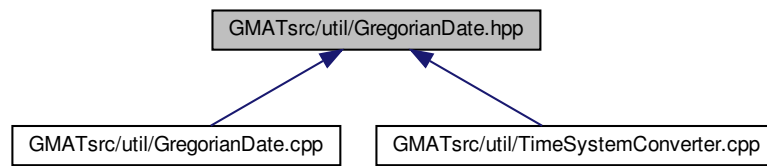
#include <iostream>
#include <sstream>
#include "utildefs.hpp"
#include "Date.hpp"
#include "DateUtil.hpp"
#include "StringTokenizer.hpp"
#include "MessageInterface.hpp"

```

Include dependency graph for GregorianCalendarDate.hpp:



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

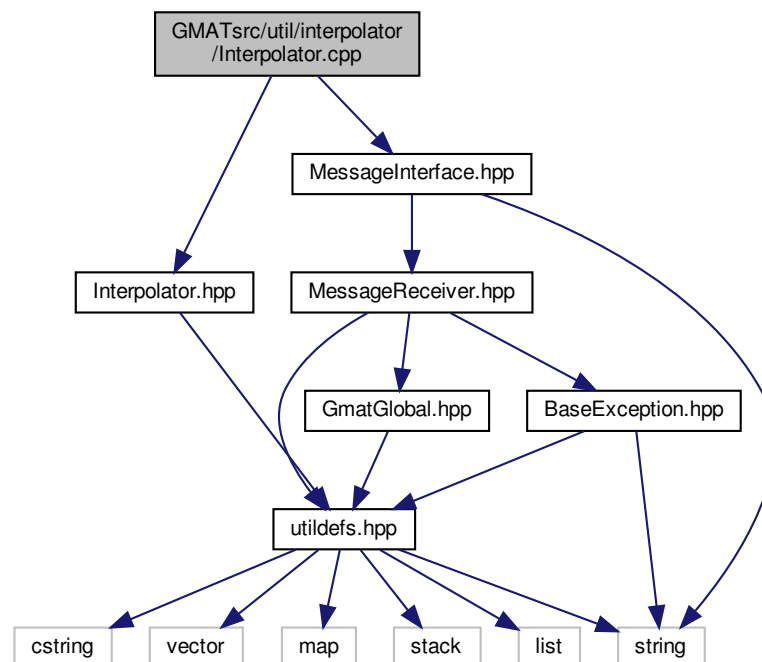


## Classes

- class [GregorianDate](#)
- class [GregorianDate::GregorianDateException](#)

## 9.42 GMATsrc/util/interpolator/Interpolator.cpp File Reference

```
#include "Interpolator.hpp"
#include "MessageInterface.hpp"
Include dependency graph for Interpolator.cpp:
```

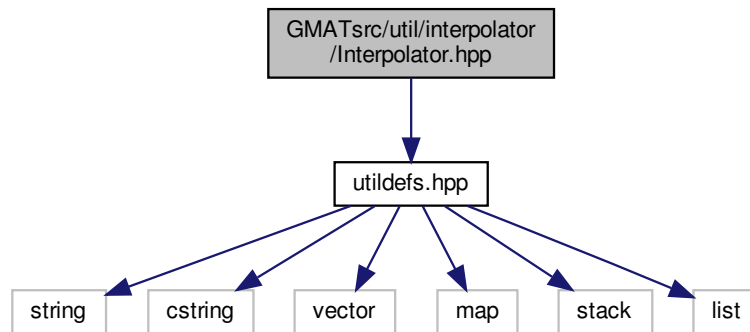




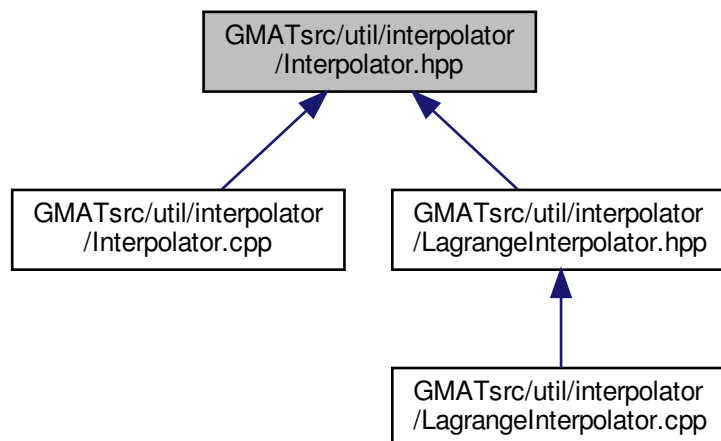
## 9.43 GMATsrc/util/interpolator/Interpolator.hpp File Reference

```
#include "utildefs.hpp"
```

Include dependency graph for Interpolator.hpp:



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



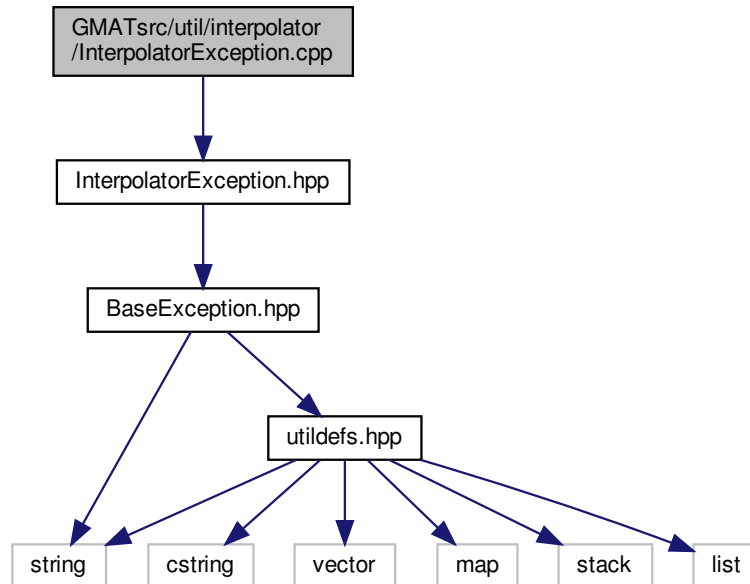
### Classes

- class [Interpolator](#)

## 9.44 GMATsrc/util/interpolator/InterpolatorException.cpp File Reference

```
#include "InterpolatorException.hpp"
```

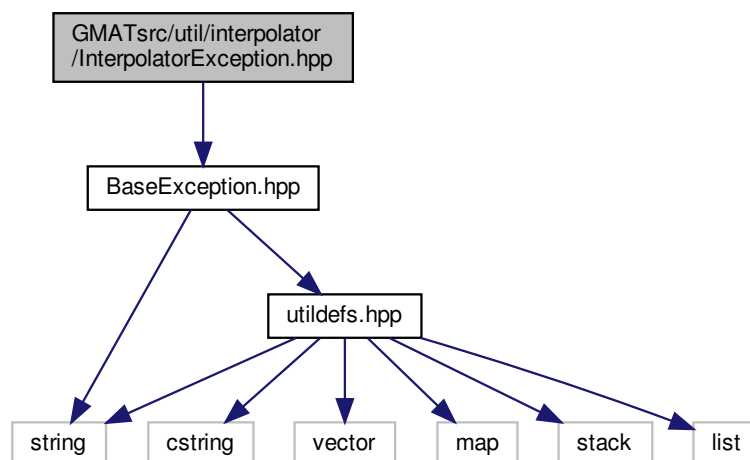
Include dependency graph for InterpolatorException.cpp:



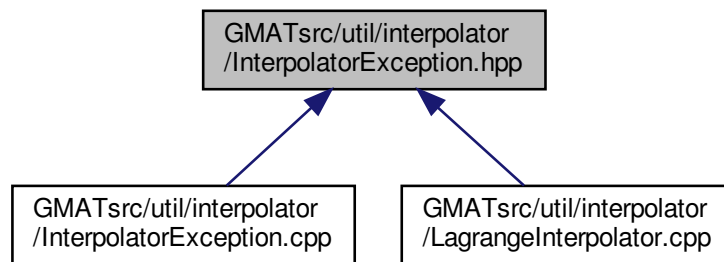
## 9.45 GMATsrc/util/interpolator/InterpolatorException.hpp File Reference

```
#include "BaseException.hpp"
```

Include dependency graph for InterpolatorException.hpp:



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



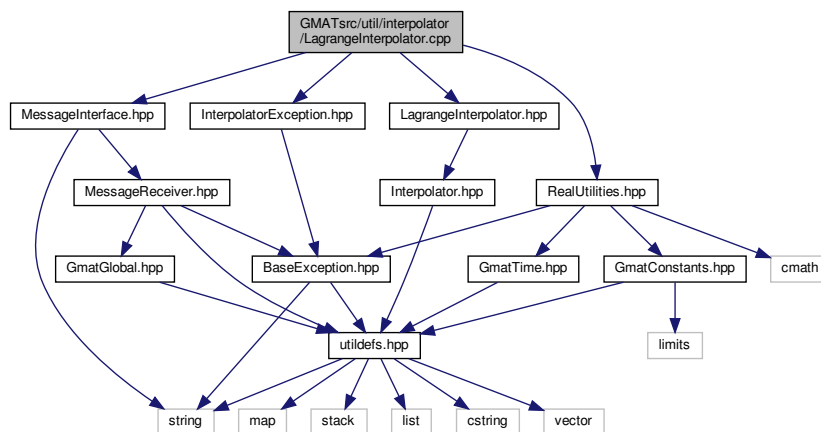
## Classes

- class [InterpolatorException](#)

## 9.46 GMATsrc/util/interpolator/LagrangeInterpolator.cpp File Reference

```
#include "LagrangeInterpolator.hpp"
#include "InterpolatorException.hpp"
#include "RealUtilities.hpp"
#include "MessageInterface.hpp"
```

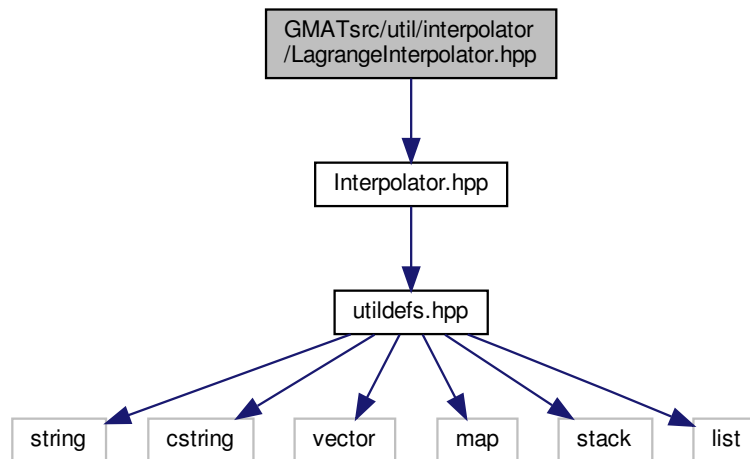
Include dependency graph for LagrangeInterpolator.cpp:



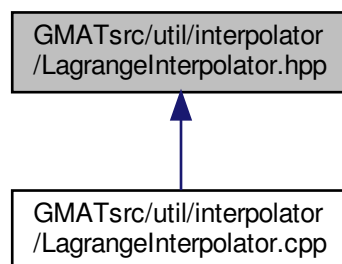
## 9.47 GMATsrc/util/interpolator/LagrangeInterpolator.hpp File Reference

```
#include "Interpolator.hpp"
```

Include dependency graph for LagrangeInterpolator.hpp:



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



### Classes

- class [LagrangeInterpolator](#)

## 9.48 GMATsrc/util/LeapSecsFileReader.cpp File Reference

```
#include "LeapSecsFileReader.hpp"
```

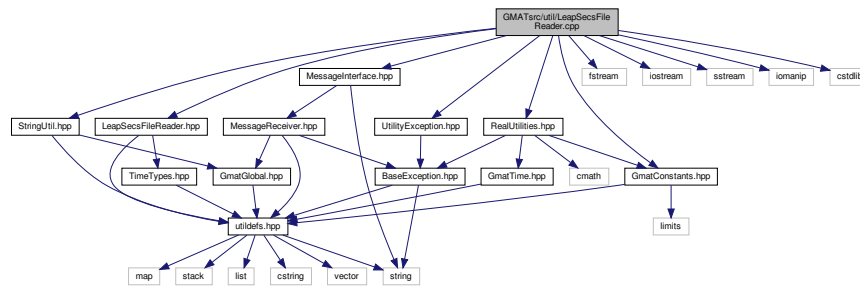
```
#include "MessageInterface.hpp"
```

```

#include "GmatConstants.hpp"
#include "RealUtilities.hpp"
#include "StringUtil.hpp"
#include "UtilityException.hpp"
#include <fstream>
#include <iostream>
#include <sstream>
#include <iomanip>
#include <cstdlib>

```

Include dependency graph for LeapSecsFileReader.cpp:



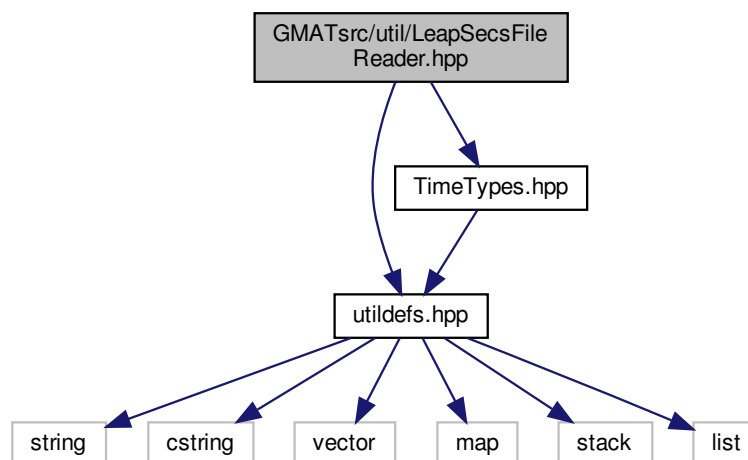
## 9.49 GMATsrc/util/LeapSecsFileReader.hpp File Reference

```

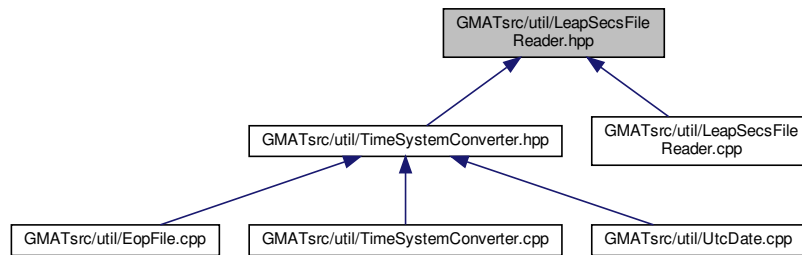
#include "utildefs.hpp"
#include "TimeTypes.hpp"

```

Include dependency graph for LeapSecsFileReader.hpp:



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



## Classes

- struct [LeapSecondInformation](#)
- class [LeapSecsFileReader](#)

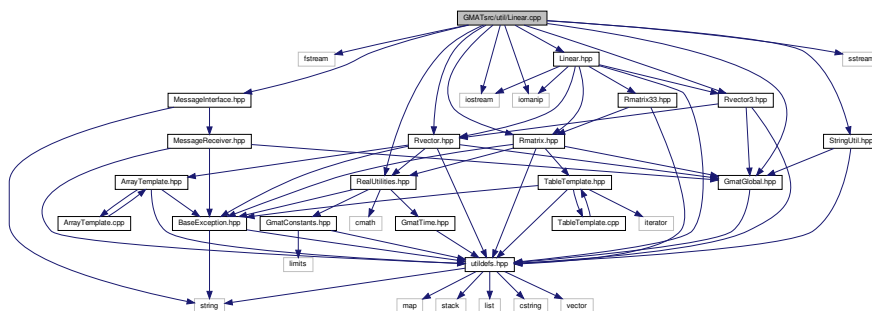
## 9.50 GMATsrc/util/Linear.cpp File Reference

```

#include <fstream>
#include <iostream>
#include <iomanip>
#include <sstream>
#include "RealUtilities.hpp"
#include "Rvector.hpp"
#include "Rvector3.hpp"
#include "Rmatrix.hpp"
#include "Linear.hpp"
#include "GmatGlobal.hpp"
#include "StringUtil.hpp"
#include "MessageInterface.hpp"

```

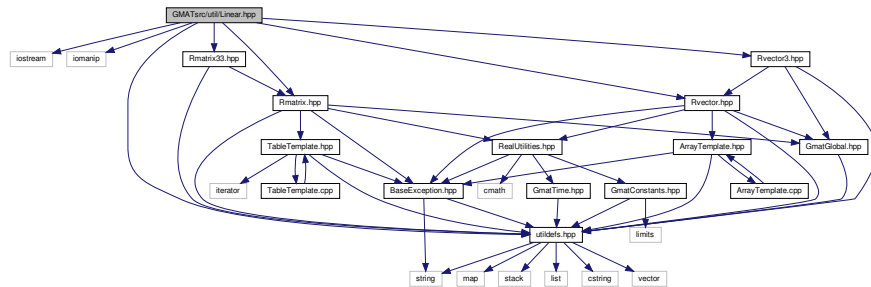
Include dependency graph for Linear.cpp:



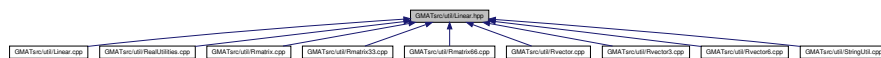
## 9.51 GMATsrc/util/Linear.hpp File Reference

```
#include <iostream>
#include <iomanip>
#include "utildefs.hpp"
#include "Rvector.hpp"
#include "Rvector3.hpp"
#include "Rmatrix.hpp"
#include "Rmatrix33.hpp"
```

Include dependency graph for Linear.hpp:



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



### Classes

- struct [GmatRealUtil::RaCodec](#)
- struct [GmatRealUtil::RaDec](#)

### Namespaces

- [GmatRealUtil](#)

### Functions

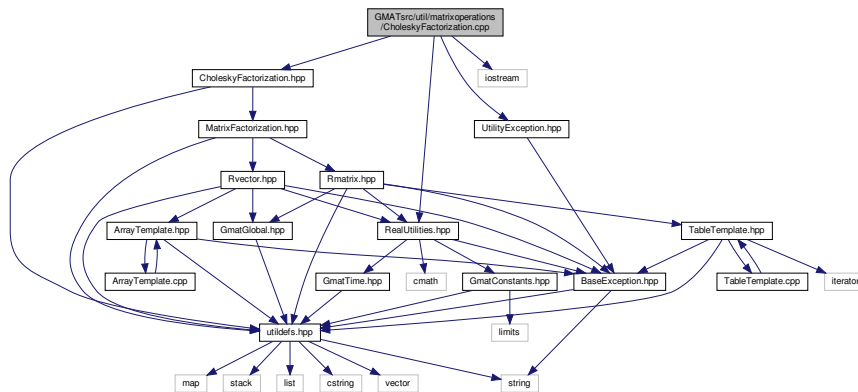
- [GMATUTIL\\_API RaCodec GmatRealUtil::CartesianToRaCodec](#) (const [Rvector3](#) &r)
- [GMATUTIL\\_API RaDec GmatRealUtil::CartesianToRaDec](#) (const [Rvector3](#) &r)
- [GMATUTIL\\_API Rvector3 GmatRealUtil::RaCodecToCartesian](#) (const [RaCodec](#) &r)
- [GMATUTIL\\_API RaDec GmatRealUtil::RaCodecToRaDec](#) (const [RaCodec](#) &r)
- [GMATUTIL\\_API Rvector3 GmatRealUtil::RaDecToCartesian](#) (const [RaDec](#) &r)
- [GMATUTIL\\_API RaCodec GmatRealUtil::RaDecToRaCodec](#) (const [RaDec](#) &r)
- [GMATUTIL\\_API Real GmatRealUtil::Min](#) (const [Rvector](#) &numbers)
- [GMATUTIL\\_API Real GmatRealUtil::Max](#) (const [Rvector](#) &numbers)
- [GMATUTIL\\_API std::istream & GmatRealUtil::operator>>](#) (std::istream &input, [Rvector](#) &a)
- [GMATUTIL\\_API std::ostream & GmatRealUtil::operator<<](#) (std::ostream &output, const [Rvector](#) &a)

- **GMATUTIL\_API** `std::istream & GmatRealUtil::operator>> (std::istream &input, Rmatrix &a)`
- **GMATUTIL\_API** `std::ostream & GmatRealUtil::operator<< (std::ostream &output, const Rmatrix &a)`
- **GMATUTIL\_API** `std::string GmatRealUtil::RealToString (const Real &rval, bool useCurrentFormat=true, bool scientific=false, bool showPoint=false, Integer precision=GmatGlobal::DATA_PRECISION, Integer width=GmatGlobal::DATA_WIDTH)`
- **GMATUTIL\_API** `std::string GmatRealUtil::ToString (const Real &rval, bool useCurrentFormat=true, bool scientific=false, bool showPoint=false, Integer precision=GmatGlobal::DATA_PRECISION, Integer width=GmatGlobal::DATA_WIDTH)`
- **GMATUTIL\_API** `std::string GmatRealUtil::ToString (const Integer &ival, bool useCurrentFormat=true, Integer width=GmatGlobal::INTEGER_WIDTH)`

## 9.52 GMATsrc/util/matrixoperations/CholeskyFactorization.cpp File Reference

```
#include "CholeskyFactorization.hpp"
#include "RealUtilities.hpp"
#include "UtilityException.hpp"
#include <iostream>
```

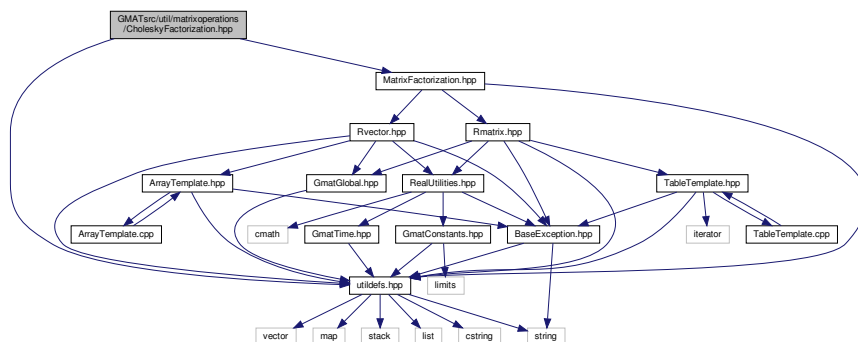
Include dependency graph for CholeskyFactorization.cpp:



## 9.53 GMATsrc/util/matrixoperations/CholeskyFactorization.hpp File Reference

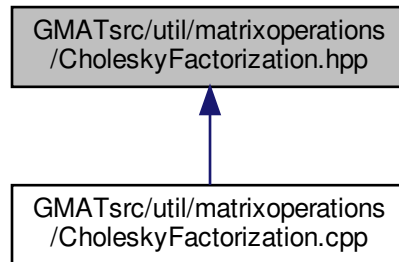
```
#include "utildefs.hpp"
#include "MatrixFactorization.hpp"
```

Include dependency graph for CholeskyFactorization.hpp:





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



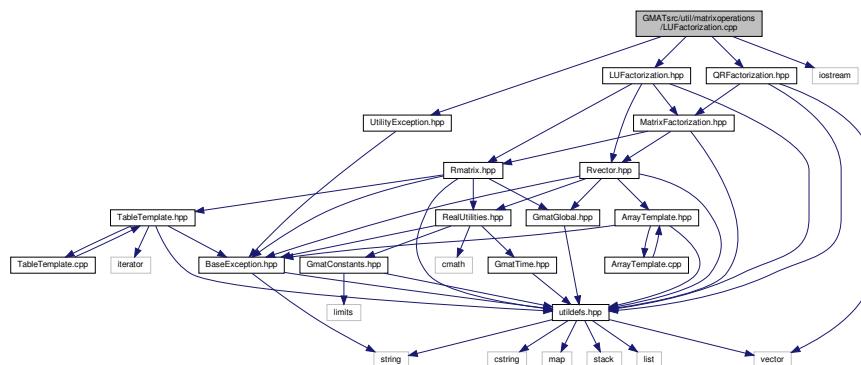
## Classes

- class [CholeskyFactorization](#)

## 9.54 GMATsrc/util/matrixoperations/LUFactorization.cpp File Reference

```
#include "LUFactorization.hpp"
#include "QRFactorization.hpp"
#include "UtilityException.hpp"
#include <iostream>
```

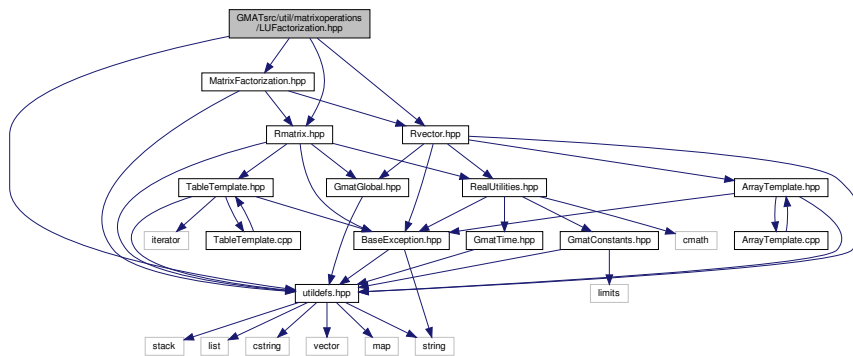
Include dependency graph for LUFactorization.cpp:



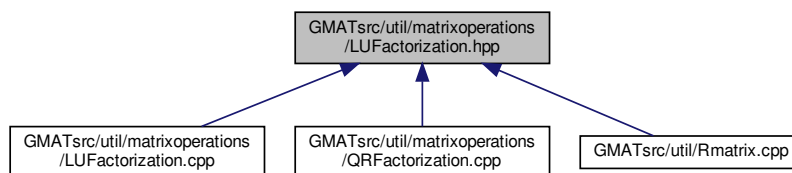
## 9.55 GMATsrc/util/matrixoperations/LUFactorization.hpp File Reference

```
#include "Rmatrix.hpp"
#include "Rvector.hpp"
#include "utildefs.hpp"
```

```
#include "MatrixFactorization.hpp"
Include dependency graph for LUFactorization.hpp:
```



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

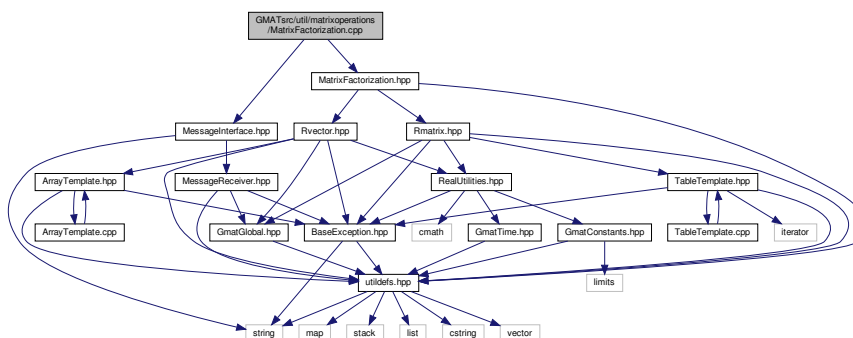


## Classes

- class [LUFactorization](#)

## 9.56 GMATsrc/util/matrixoperations/MatrixFactorization.cpp File Reference

```
#include "MatrixFactorization.hpp"
#include "MessageInterface.hpp"
Include dependency graph for MatrixFactorization.cpp:
```



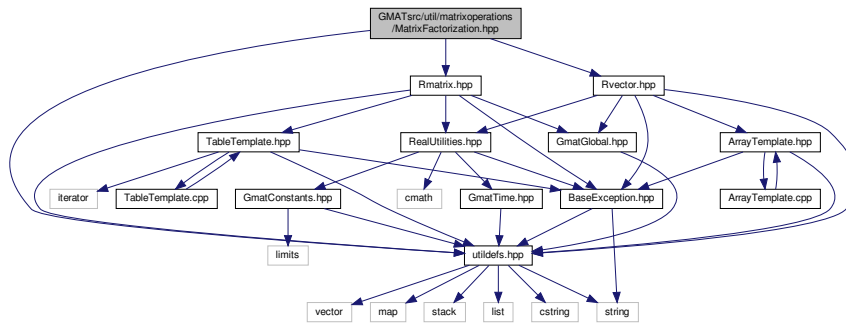
## 9.57 GMATsrc/util/matrixoperations/MatrixFactorization.hpp File Reference

```
#include "utildefs.hpp"
```

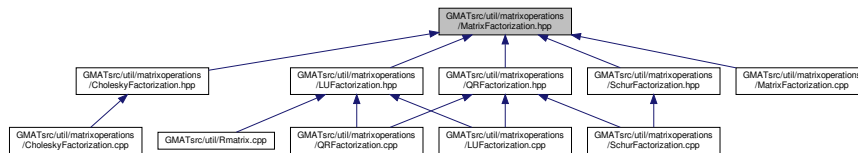
```
#include "Rmatrix.hpp"
```

```
#include "Rvector.hpp"
```

Include dependency graph for MatrixFactorization.hpp:



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



### Classes

- class [MatrixFactorization](#)

## 9.58 GMATsrc/util/matrixoperations/QRFactorization.cpp File Reference

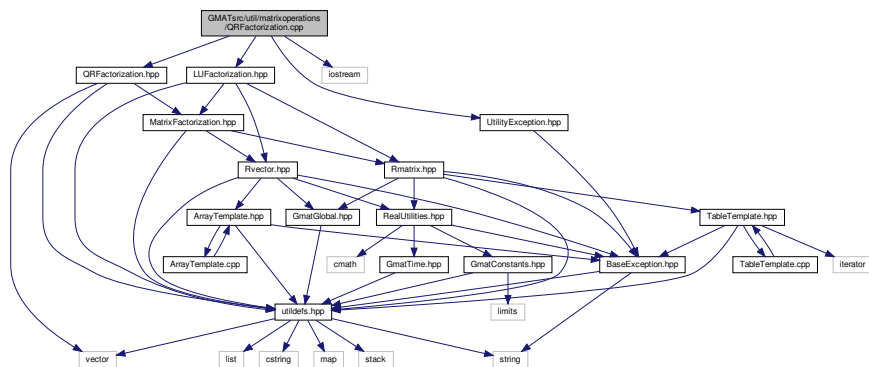
```
#include "QRFactorization.hpp"
```

```
#include "UtilityException.hpp"
```

```
#include "LUFactorization.hpp"
```

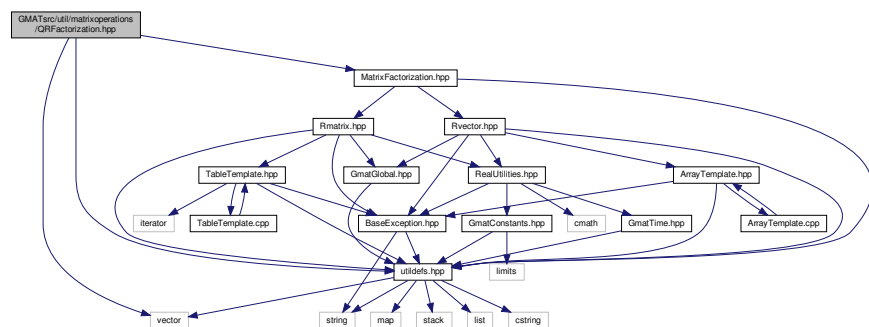
```
#include <iostream>
```

Include dependency graph for QRFactorization.cpp:

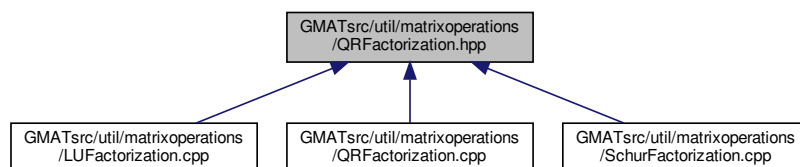


## 9.59 GMATsrc/util/matrixoperations/QRFactorization.hpp File Reference

```
#include <vector>
#include "MatrixFactorization.hpp"
#include "utildefs.hpp"
Include dependency graph for QRFactorization.hpp:
```



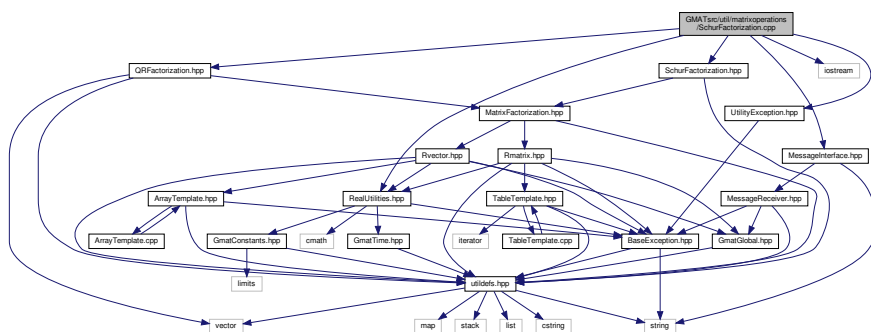
This graph shows which files directly or indirectly include this file:



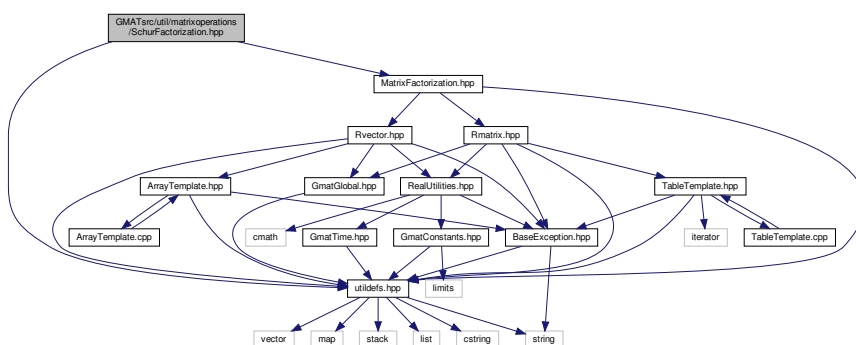
## Classes

- class [QRFactorization](#)

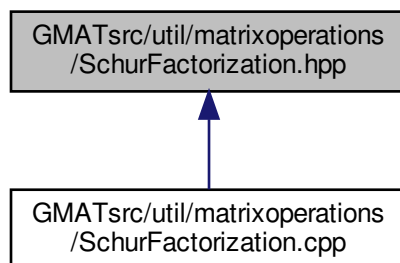
```
#include "SchurFactorization.hpp"
#include "QRFactorization.hpp"
#include "RealUtilities.hpp"
#include "UtilityException.hpp"
#include <iostream>
#include "MessageInterface.hpp"
Include dependency graph for SchurFactorization.cpp:
```



```
#include "utildefs.hpp"
#include "MatrixFactorization.hpp"
Include dependency graph for SchurFactorization.hpp:
```



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



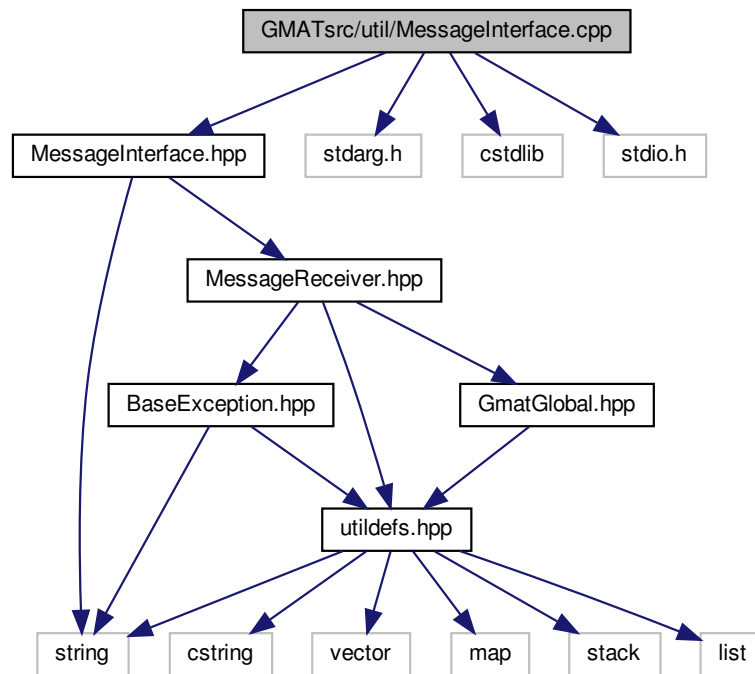
## Classes

- class [SchurFactorization](#)

## 9.62 GMATsrc/util/MessageInterface.cpp File Reference

```
#include "MessageInterface.hpp"
#include <stdarg.h>
#include <cstdlib>
#include <stdio.h>
```

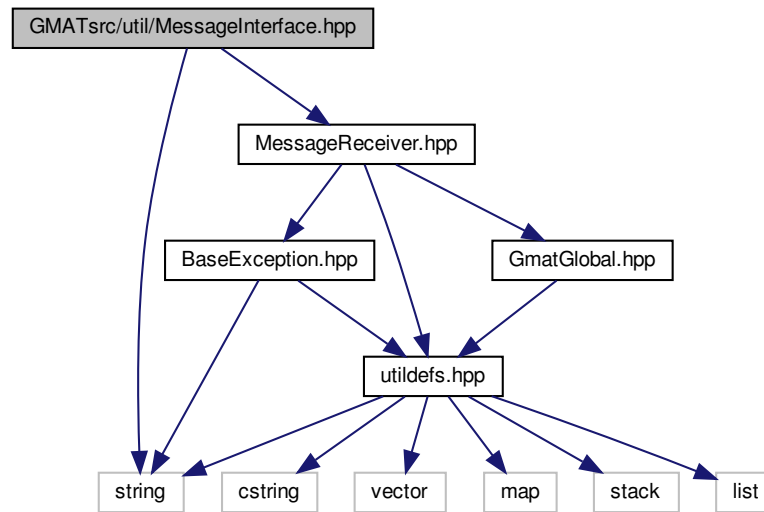
Include dependency graph for MessageInterface.cpp:



## 9.63 GMATsrc/util/MessageInterface.hpp File Reference

```
#include <string>
#include "MessageReceiver.hpp"
```

Include dependency graph for MessageInterface.hpp:



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



## Classes

- class [MessageInterface](#)

## 9.64 GMATsrc/util/MessageReceiver.cpp File Reference

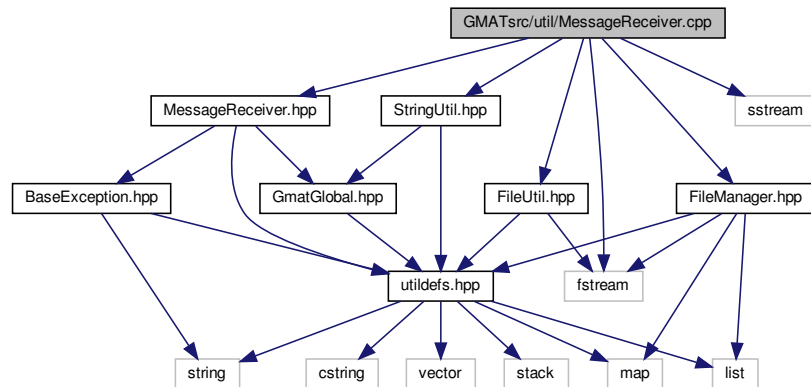
```

#include "MessageReceiver.hpp"
#include "FileManager.hpp"
#include "FileUtil.hpp"
#include "StringUtil.hpp"
#include <fstream>
#include <sstream>

```



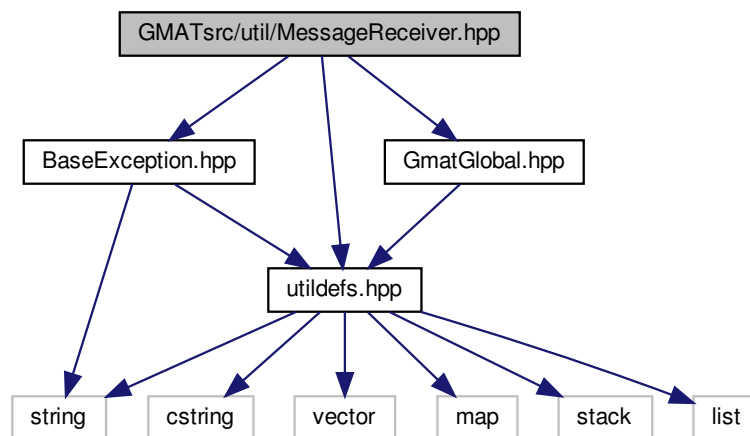
Include dependency graph for MessageReceiver.cpp:



## 9.65 GMATsrc/util/MessageReceiver.hpp File Reference

```
#include "BaseException.hpp"
#include "GmatGlobal.hpp"
#include "utildefs.hpp"
```

Include dependency graph for MessageReceiver.hpp:



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

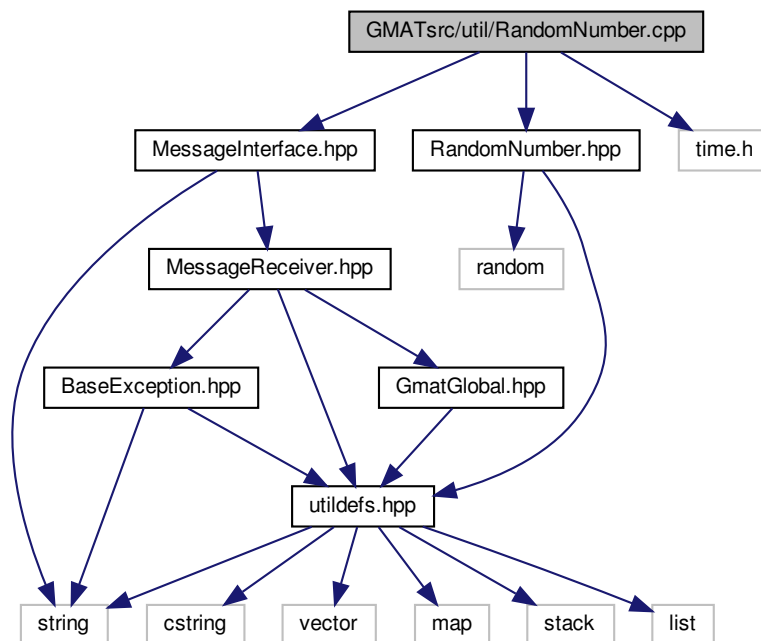


## Classes

- class [MessageReceiver](#)

## 9.66 GMATsrc/util/RandomNumber.cpp File Reference

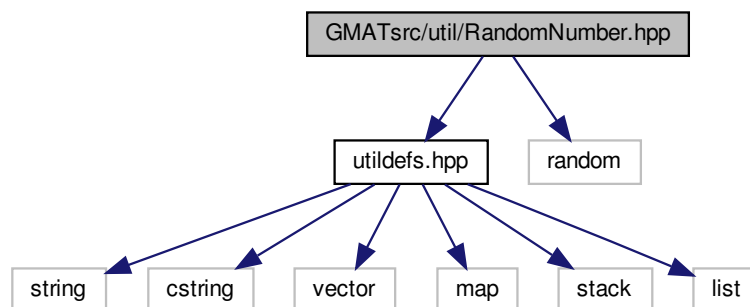
```
#include "RandomNumber.hpp"
#include "MessageInterface.hpp"
#include <time.h>
Include dependency graph for RandomNumber.cpp:
```



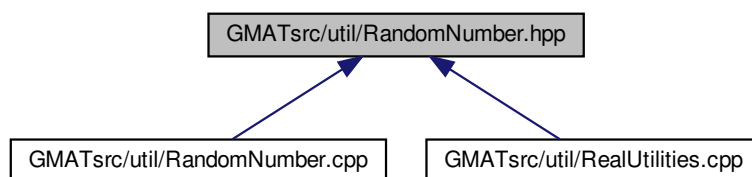
## 9.67 GMATsrc/util/RandomNumber.hpp File Reference

```
#include "utildefs.hpp"
#include <random>
```

Include dependency graph for RandomNumber.hpp:



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



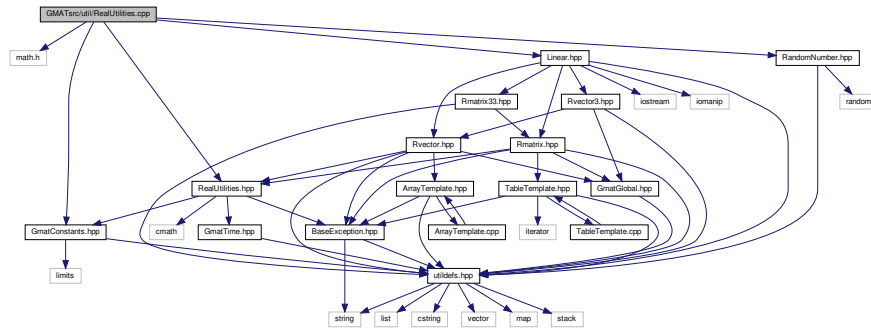
## Classes

- class [RandomNumber](#)

## 9.68 GMATsrc/util/RealUtilities.cpp File Reference

```
#include <math.h>
#include "RealUtilities.hpp"
#include "GmatConstants.hpp"
#include "Linear.hpp"
#include "RandomNumber.hpp"
```

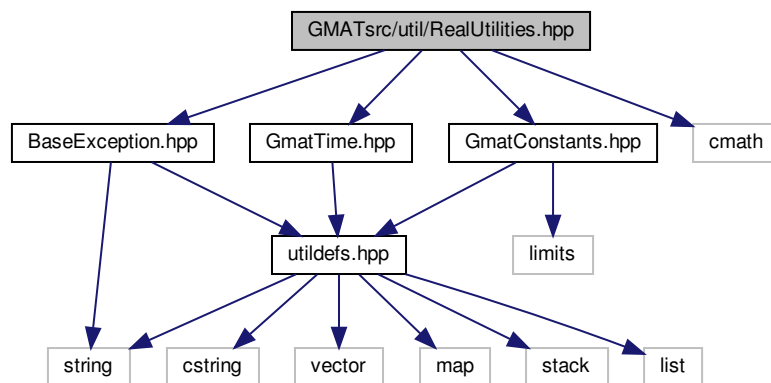
Include dependency graph for RealUtilities.cpp:



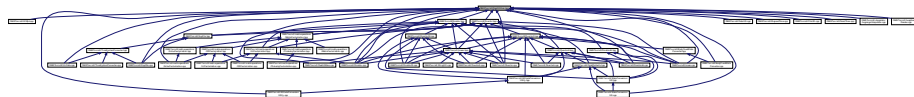
## 9.69 GMATsrc/util/RealUtilities.hpp File Reference

```
#include "BaseException.hpp"
#include "GmatConstants.hpp"
#include "GmatTime.hpp"
#include <cmath>
```

Include dependency graph for RealUtilities.hpp:



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



## Classes

- struct [RealUtilitiesExceptions](#)
- class [RealUtilitiesExceptions::ArgumentError](#)
- class [RealUtilitiesExceptions::IllegalTime](#)

## Namespaces

- [GmatMathUtil](#)

## Functions

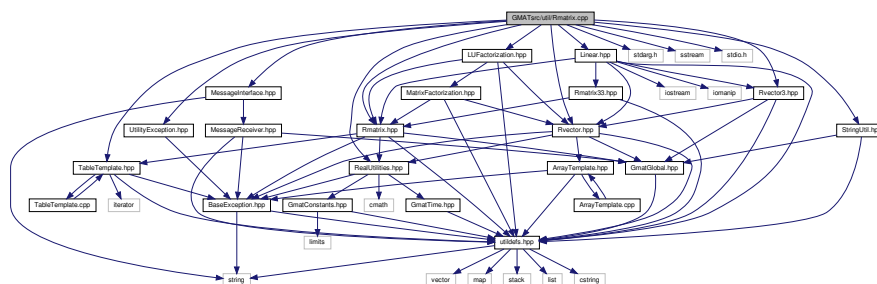
- [Integer GMATUTIL\\_API GmatMathUtil::Abs \(Integer theNumber\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::Abs \(Real theNumber\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::NearestInt \(Real theNumber\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::Round \(Real theNumber\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::Floor \(Real theNumber\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::Fix \(Real theNumber\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::Ceiling \(Real theNumber\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::Mod \(Real left, Real right\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::Rem \(Real left, Real right\)](#)
- [void GMATUTIL\\_API GmatMathUtil::Quotient \(Real top, Real bottom, Integer &result\)](#)
- [void GMATUTIL\\_API GmatMathUtil::Quotient \(Real top, Real bottom, Real &result\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::Min \(Real left, Real right\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::Max \(Real left, Real right\)](#)
- [bool GMATUTIL\\_API GmatMathUtil::IsPositive \(Real theNumber\)](#)
- [bool GMATUTIL\\_API GmatMathUtil::IsNegative \(Real theNumber\)](#)
- [bool GMATUTIL\\_API GmatMathUtil::IsNonNegative \(Real theNumber\)](#)
- [bool GMATUTIL\\_API GmatMathUtil::IsZero \(Real theNumber, Real accuracy=GmatRealConstants::REAL\\_EPSILON\)](#)
- [bool GMATUTIL\\_API GmatMathUtil::IsEqual \(Real left, Real right, Real accuracy=GmatRealConstants::REAL\\_EPSILON\)](#)
- [bool GMATUTIL\\_API GmatMathUtil::IsEqual \(GmatTime left, GmatTime right, Real accuracy=GmatRealConstants::REAL\\_EPSILON\)](#)
- [Integer GMATUTIL\\_API GmatMathUtil::SignOf \(Real theNumber\)](#)
- [bool GMATUTIL\\_API GmatMathUtil::IsOdd \(Integer theNumber\)](#)
- [bool GMATUTIL\\_API GmatMathUtil::IsEven \(Integer theNumber\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::Rad \(Real angleInDeg, bool modBy2Pi=false\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::Deg \(Real angleInRad, bool modBy360=false\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::DegToRad \(Real deg, bool modBy2Pi=false\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::RadToDeg \(Real rad, bool modBy360=false\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::ArcsecToDeg \(Real asec, bool modBy360=false\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::ArcsecToRad \(Real asec, bool modBy2Pi=false\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::Sin \(Real angleInRad, Real cycleInRad=GmatMathConstants::TWO\\_PI\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::SinXOverX \(Real angleInRad, Real cycleInRad=GmatMathConstants::TWO\\_PI\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::Cos \(Real angleInRad, Real cycleInRad=GmatMathConstants::TWO\\_PI\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::Tan \(Real angleInRad, Real cycleInRad=GmatMathConstants::TWO\\_PI\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::Cosh \(Real angleInRad, Real cycleInRad=GmatMathConstants::TWO\\_PI\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::Sinh \(Real angleInRad, Real cycleInRad=GmatMathConstants::TWO\\_PI\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::Tanh \(Real angleInRad, Real cycleInRad=GmatMathConstants::TWO\\_PI\)](#)
- [Real GMATUTIL\\_API GmatMathUtil::ASin \(Real x, Real tol=GmatRealConstants::REAL\\_TOL, Real cycleInRad=GmatMathConstants::TWO\\_PI\)](#)

- `Real GMATUTIL_API GmatMathUtil::ACos (Real x, Real tol=GmatRealConstants::REAL_TOL, Real cycleInRad=GmatMathConstants::TWO_PI)`
- `Real GMATUTIL_API GmatMathUtil::ATan (Real y, Real x=1.0, Real cycleInRad=GmatMathConstants::TWO_PI)`
- `Real GMATUTIL_API GmatMathUtil::ATan2 (Real y, Real x=1.0, Real cycleInRad=GmatMathConstants::TWO_PI)`
- `Real GMATUTIL_API GmatMathUtil::ASinh (Real x, Real cycleInRad=GmatMathConstants::TWO_PI)`
- `Real GMATUTIL_API GmatMathUtil::ACosh (Real x, Real cycleInRad=GmatMathConstants::TWO_PI)`
- `Real GMATUTIL_API GmatMathUtil::ATanh (Real x, Real cycleInRad=GmatMathConstants::TWO_PI)`
- `Real GMATUTIL_API GmatMathUtil::Ln (Real x)`
- `Real GMATUTIL_API GmatMathUtil::Log (Real x)`
- `Real GMATUTIL_API GmatMathUtil::Log10 (Real x)`
- `Real GMATUTIL_API GmatMathUtil::Log (Real x, Real base)`
- `Real GMATUTIL_API GmatMathUtil::Log (Real x, Integer base)`
- `Real GMATUTIL_API GmatMathUtil::Rand (Real lowerBound=0.0, Real upperBound=1.0)`
- `Real GMATUTIL_API GmatMathUtil::Randn (Real mean=0.0, Real stdev=1.0)`
- `void GMATUTIL_API GmatMathUtil::SetSeed (UnsignedInt seed)`
- `Real GMATUTIL_API GmatMathUtil::Cbrt (Real x)`
- `Real GMATUTIL_API GmatMathUtil::Sqrt (Real x)`
- `Real GMATUTIL_API GmatMathUtil::Exp (Real x)`
- `Real GMATUTIL_API GmatMathUtil::Exp10 (Real x)`
- `Real GMATUTIL_API GmatMathUtil::Pow (Real x, Real y)`
- `Real GMATUTIL_API GmatMathUtil::Pow (Real x, Integer y)`
- `bool GMATUTIL_API GmatMathUtil::IsNaN (Real x)`
- `bool GMATUTIL_API GmatMathUtil::IsInf (Real x)`

## 9.70 GMATsrc/util/Rmatrix.cpp File Reference

```
#include "TableTemplate.hpp"
#include "Rmatrix.hpp"
#include "Rvector.hpp"
#include "Rvector3.hpp"
#include "RealUtilities.hpp"
#include "UtilityException.hpp"
#include "Linear.hpp"
#include "StringUtil.hpp"
#include <stdarg.h>
#include <sstream>
#include <stdio.h>
#include "MessageInterface.hpp"
#include "LUFactorization.hpp"
```

Include dependency graph for Rmatrix.cpp:



## Functions

- [Rmatrix operator+](#) ([Real](#) scalar, const [Rmatrix](#) &m)
- [Rmatrix operator-](#) ([Real](#) scalar, const [Rmatrix](#) &m)
- [Rmatrix operator\\*](#) ([Real](#) scalar, const [Rmatrix](#) &m)
- [Rmatrix operator/](#) ([Real](#) scalar, const [Rmatrix](#) &m)
- [Rmatrix SkewSymmetric4by4](#) (const [Rvector3](#) &v)
- [Rmatrix TransposeTimesMatrix](#) (const [Rmatrix](#) &m1, const [Rmatrix](#) &m2)
- [Rmatrix MatrixTimesTranspose](#) (const [Rmatrix](#) &m1, const [Rmatrix](#) &m2)
- [Rmatrix TransposeTimesTranspose](#) (const [Rmatrix](#) &m1, const [Rmatrix](#) &m2)
- `std::istream & operator>>` (`std::istream &input`, [Rmatrix](#) &a)
- `std::ostream & operator<<` (`std::ostream &output`, const [Rmatrix](#) &a)

### 9.70.1 Function Documentation

#### 9.70.1.1 [MatrixTimesTranspose\(\)](#)

```
Rmatrix MatrixTimesTranspose (
 const Rmatrix & m1,
 const Rmatrix & m2)
```

#### 9.70.1.2 [operator\\*\(\)](#)

```
Rmatrix operator* (
 Real scalar,
 const Rmatrix & m)
```

#### 9.70.1.3 [operator+\(\)](#)

```
Rmatrix operator+ (
 Real scalar,
 const Rmatrix & m)
```

#### 9.70.1.4 [operator-\(\)](#)

```
Rmatrix operator- (
 Real scalar,
 const Rmatrix & m)
```

#### 9.70.1.5 operator/()

```
Rmatrix operator/ (
 Real scalar,
 const Rmatrix & m)
```

#### 9.70.1.6 operator<<()

```
std::ostream& operator<< (
 std::ostream & output,
 const Rmatrix & a)
```

Formats [Rmatrix](#) value using global format and sends to output stream. Once global format is set, it remains the same format until it is reset by global->SetActualFormat().

##### Parameters

|               |                                      |
|---------------|--------------------------------------|
| <i>output</i> | Output stream                        |
| <i>a</i>      | <a href="#">Rmatrix</a> to write out |

return Output stream

#### 9.70.1.7 operator>>()

```
std::istream& operator>> (
 std::istream & input,
 Rmatrix & a)
```

#### 9.70.1.8 SkewSymmetric4by4()

```
Rmatrix SkewSymmetric4by4 (
 const Rvector3 & v)
```

#### 9.70.1.9 TransposeTimesMatrix()

```
Rmatrix TransposeTimesMatrix (
 const Rmatrix & m1,
 const Rmatrix & m2)
```

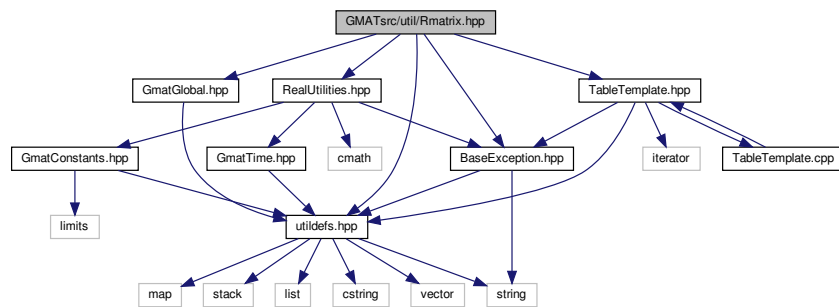


## 9.70.1.10 TransposeTimesTranspose()

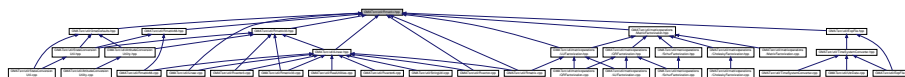
```
Rmatrix TransposeTimesTranspose (
 const Rmatrix & m1,
 const Rmatrix & m2)
```

## 9.71 GMATsrc/util/Rmatrix.hpp File Reference

```
#include "RealUtilities.hpp"
#include "TableTemplate.hpp"
#include "BaseException.hpp"
#include "utildefs.hpp"
#include "GmatGlobal.hpp"
Include dependency graph for Rmatrix.hpp:
```



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



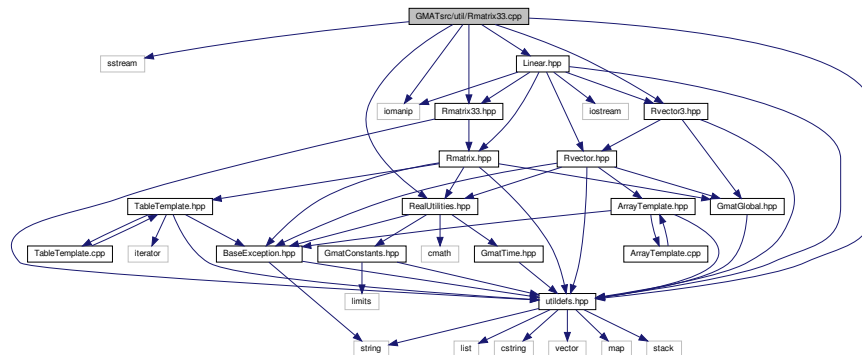
## Classes

- class [Rmatrix](#)
- class [Rmatrix::NotSquare](#)
- class [Rmatrix::IsSingular](#)
- class [Rmatrix::DivideByZero](#)

## 9.72 GMATsrc/util/Rmatrix33.cpp File Reference

```
#include <sstream>
#include <iomanip>
#include "utildefs.hpp"
#include "Rvector3.hpp"
```

```
#include "Rmatrix33.hpp"
#include "RealUtilities.hpp"
#include "Linear.hpp"
Include dependency graph for Rmatrix33.cpp:
```



## Functions

- [Rmatrix33 operator\\*](#) ([Real](#) scalar, const [Rmatrix33](#) &m)
- [Rmatrix33 SkewSymmetric](#) (const [Rvector3](#) &v)
- [Rmatrix33 TransposeTimesRmatrix](#) (const [Rmatrix33](#) &m1, const [Rmatrix33](#) &m2)
- [Rmatrix33 MatrixTimesTranspose](#) (const [Rmatrix33](#) &m1, const [Rmatrix33](#) &m2)
- [Rmatrix33 TransposeTimesTranspose](#) (const [Rmatrix33](#) &m1, const [Rmatrix33](#) &m2)

### 9.72.1 Function Documentation

#### 9.72.1.1 MatrixTimesTranspose()

```
Rmatrix33 MatrixTimesTranspose (
 const Rmatrix33 & m1,
 const Rmatrix33 & m2)
```

#### 9.72.1.2 operator\*()

```
Rmatrix33 operator* (
 Real scalar,
 const Rmatrix33 & m)
```





## 9.74.1.2 operator\*()

```
Rmatrix66 operator* (
 Real scalar,
 const Rmatrix66 & m)
```

## 9.74.1.3 SkewSymmetric()

```
Rmatrix66 SkewSymmetric (
 const Rvector6 & v)
```

## 9.74.1.4 TransposeTimesMatrix()

```
Rmatrix66 TransposeTimesMatrix (
 const Rmatrix66 & m1,
 const Rmatrix66 & m2)
```

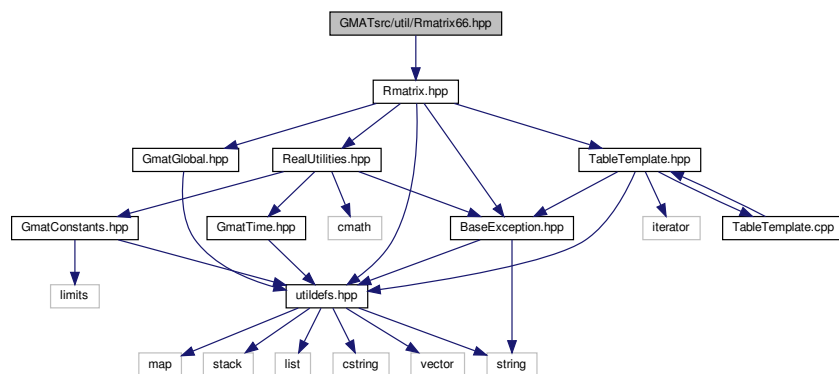
## 9.74.1.5 TransposeTimesTranspose()

```
Rmatrix66 TransposeTimesTranspose (
 const Rmatrix66 & m1,
 const Rmatrix66 & m2)
```

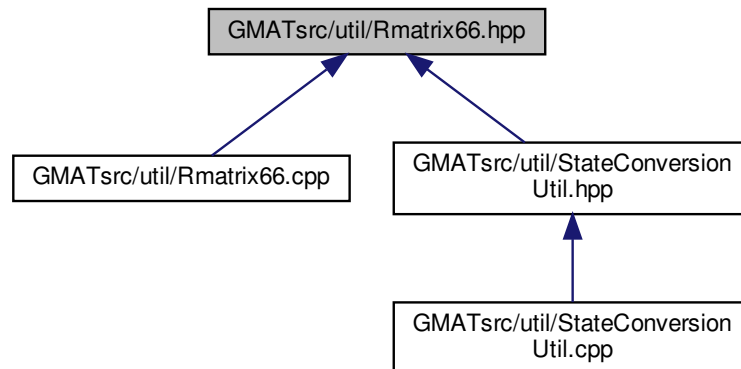
## 9.75 GMATsrc/util/Rmatrix66.hpp File Reference

```
#include "Rmatrix.hpp"
```

Include dependency graph for Rmatrix66.hpp:



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



## Classes

- class [Rmatrix66](#)

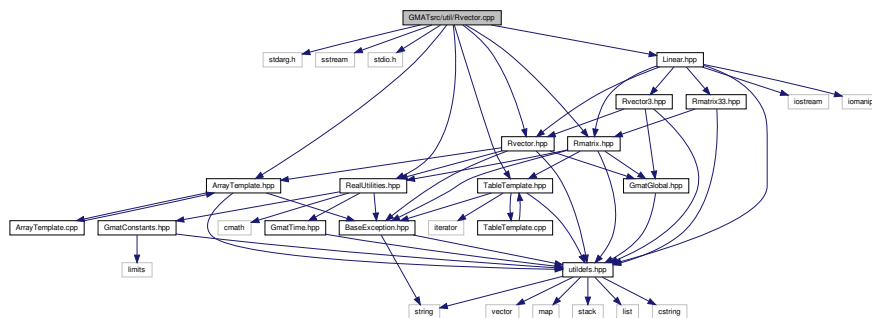
## 9.76 GMATsrc/util/Rvector.cpp File Reference

```

#include <stdarg.h>
#include <sstream>
#include <stdio.h>
#include "ArrayTemplate.hpp"
#include "TableTemplate.hpp"
#include "Rmatrix.hpp"
#include "RealUtilities.hpp"
#include "Rvector.hpp"
#include "Linear.hpp"

```

Include dependency graph for `Rvector.cpp`:



## Functions

- [Rvector operator\\*](#) ([Real](#) s, const [Rvector](#) &v)
- [Rmatrix Outerproduct](#) (const [Rvector](#) &v1, const [Rvector](#) &v2)
- `std::istream & operator>>` (`std::istream &input`, [Rvector](#) &a)
- `std::ostream & operator<<` (`std::ostream &output`, const [Rvector](#) &a)
- `bool operator<` (const [Element](#) &e1, const [Element](#) &e2)

### 9.76.1 Function Documentation

#### 9.76.1.1 operator\*()

```
Rvector operator* (
 Real s,
 const Rvector & v)
```

#### 9.76.1.2 operator<()

```
bool operator< (
 const Element & e1,
 const Element & e2)
```

provides ordering function for struct [Element](#), which is used by STL list sort function when sorting list of values and original indices

e1 left hand [Element](#) in test

#### Parameters

|           |                                            |
|-----------|--------------------------------------------|
| <i>e2</i> | right hand <a href="#">Element</a> in test |
|-----------|--------------------------------------------|

#### Returns

value is true if `e1.value < e2.value`

#### 9.76.1.3 operator<<()

```
std::ostream& operator<< (
 std::ostream & output,
 const Rvector & a)
```

Formats [Rvector](#) value using global format and sends to output stream. Once global format is set, it remains the same format until it is reset by `global->SetActualFormat()`.

## Parameters

|               |                                      |
|---------------|--------------------------------------|
| <i>output</i> | Output stream                        |
| <i>a</i>      | <a href="#">Rvector</a> to write out |

return Output stream

9.76.1.4 `operator>>()`

```
std::istream& operator>> (
 std::istream & input,
 Rvector & a)
```

9.76.1.5 `Outerproduct()`

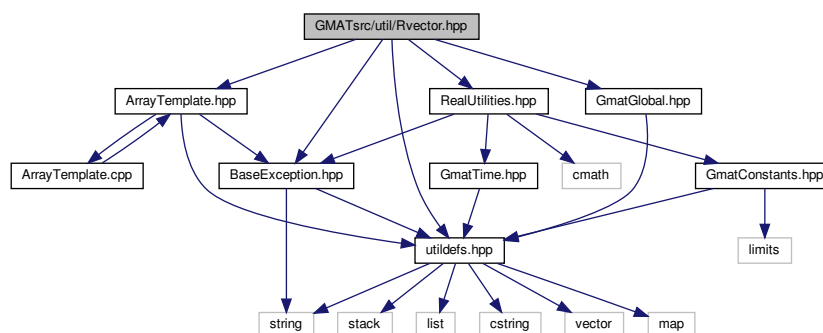
```
Rmatrix Outerproduct (
 const Rvector & v1,
 const Rvector & v2)
```

## Note

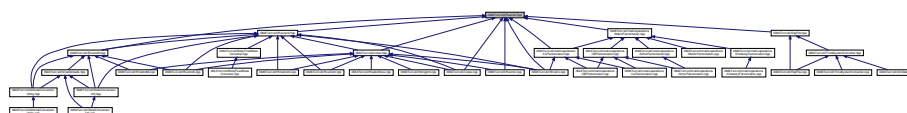
[Rmatrix](#) multiplication, with first vector considered a Nx1 matrix and a second vector considered a 1xM matrix.

9.77 `GMATsrc/util/Rvector.hpp` File Reference

```
#include "utildefs.hpp"
#include "ArrayTemplate.hpp"
#include "BaseException.hpp"
#include "RealUtilities.hpp"
#include "GmatGlobal.hpp"
Include dependency graph for Rvector.hpp:
```



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





## Classes

- class [Rvector](#)
- class [Rvector::ZeroVector](#)
- struct [Element](#)

## Functions

- bool [operator<](#) (const [Element](#) &e1, const [Element](#) &e2)

### 9.77.1 Function Documentation

#### 9.77.1.1 [operator<\(\)](#)

```
bool operator< (
 const Element & e1,
 const Element & e2)
```

provides ordering function for struct [Element](#), which is used by STL list sort function when sorting list of values and original indices

e1 left hand [Element](#) in test

#### Parameters

|    |                                            |
|----|--------------------------------------------|
| e2 | right hand <a href="#">Element</a> in test |
|----|--------------------------------------------|

#### Returns

value is true if e1.value < e2.value

## 9.78 GMATsrc/util/Rvector3.cpp File Reference

```
#include <sstream>
#include "utildefs.hpp"
#include "Rvector3.hpp"
#include "Rmatrix33.hpp"
#include "Linear.hpp"
#include "MessageInterface.hpp"
```



## 9.78.1.4 operator&gt;&gt;()

```
std::istream& operator>> (
 std::istream & input,
 Rvector3 & a)
```

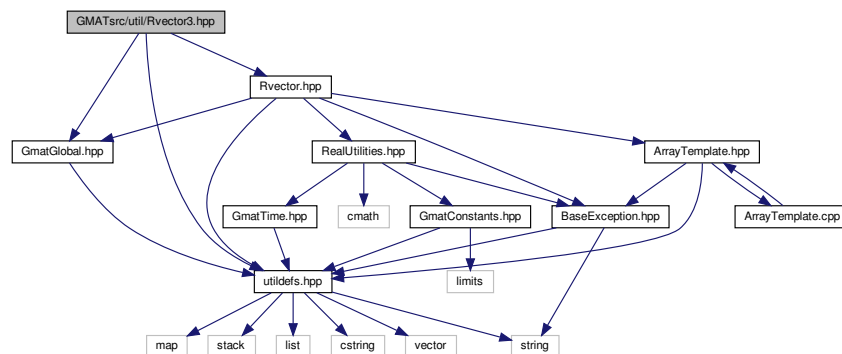
## 9.78.1.5 Outerproduct()

```
Rmatrix33 Outerproduct (
 const Rvector3 & v1,
 const Rvector3 & v2)
```

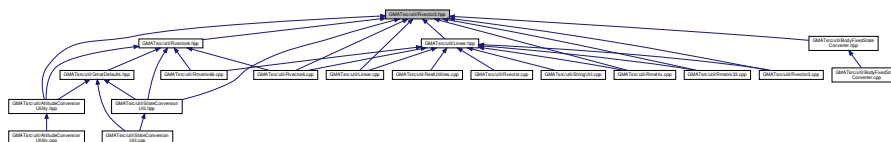
## 9.79 GMATsrc/util/Rvector3.hpp File Reference

```
#include "utildefs.hpp"
#include "Rvector.hpp"
#include "GmatGlobal.hpp"
```

Include dependency graph for Rvector3.hpp:



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



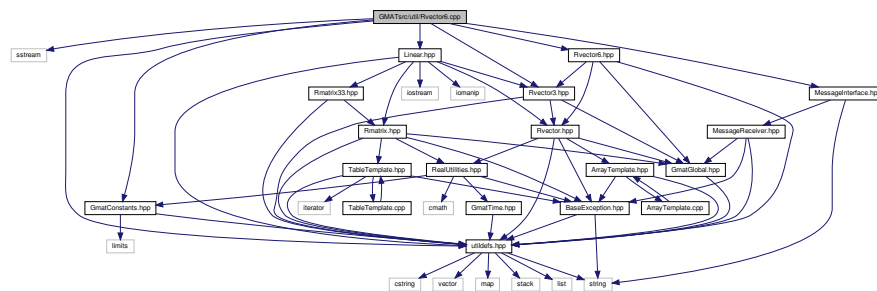
## Classes

- class [Rvector3](#)

## 9.80 GMATsrc/util/Rvector6.cpp File Reference

```
#include <sstream>
#include "utildefs.hpp"
#include "Rvector3.hpp"
#include "Rvector6.hpp"
#include "Linear.hpp"
#include "GmatConstants.hpp"
#include "MessageInterface.hpp"
```

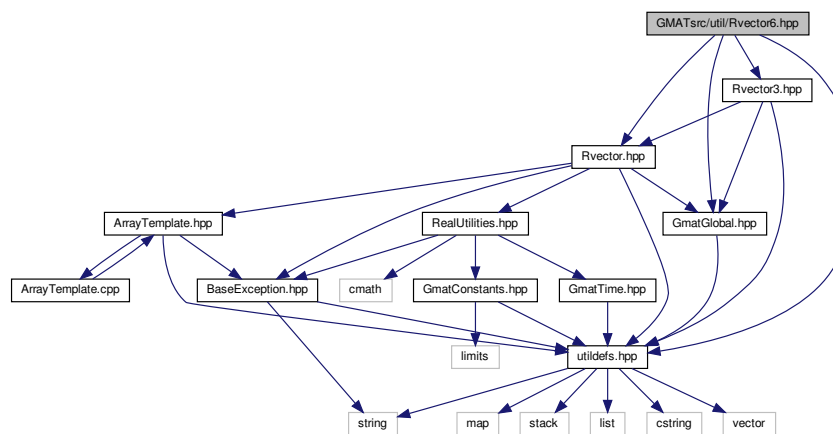
Include dependency graph for Rvector6.cpp:



## 9.81 GMATsrc/util/Rvector6.hpp File Reference

```
#include "utildefs.hpp"
#include "Rvector.hpp"
#include "Rvector3.hpp"
#include "GmatGlobal.hpp"
```

Include dependency graph for Rvector6.hpp:



```

graph TD
 Rvector6_hpp[GMATsrc/Util/Rvector6.hpp]
 GmatDefaults_hpp[GMATsrc/Util/GmatDefaults.hpp]
 Rmatrix66_cpp[GMATsrc/Util/Rmatrix66.cpp]
 Rvector6_cpp[GMATsrc/Util/Rvector6.cpp]
 AttitudeConversionUtility_hpp[GMATsrc/Util/AttitudeConversionUtility.hpp]
 StateConversionUtil_hpp[GMATsrc/Util/StateConversionUtil.hpp]
 AttitudeConversionUtility_cpp[GMATsrc/Util/AttitudeConversionUtility.cpp]
 StateConversionUtil_cpp[GMATsrc/Util/StateConversionUtil.cpp]

 Rvector6_hpp --> GmatDefaults_hpp
 Rvector6_hpp --> Rmatrix66_cpp
 Rvector6_hpp --> Rvector6_cpp
 Rvector6_hpp --> AttitudeConversionUtility_hpp
 Rvector6_hpp --> StateConversionUtil_hpp
 GmatDefaults_hpp --> Rvector6_hpp
 Rmatrix66_cpp --> Rvector6_hpp
 Rvector6_cpp --> Rvector6_hpp
 AttitudeConversionUtility_hpp --> AttitudeConversionUtility_cpp
 StateConversionUtil_hpp --> StateConversionUtil_cpp

```

- class Rvector6

- #define NUM DATA INIT 6

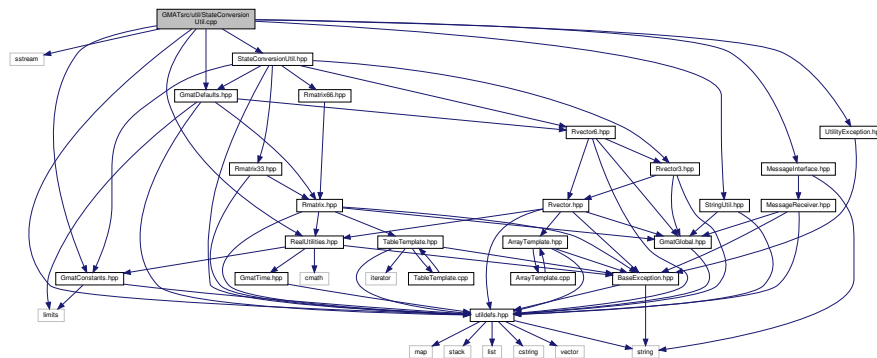
#### 9.81.1.1 NUM DATA INIT

## 9.82 GMATsrc/util/StateConversionUtil.cpp File Reference

Generated by Doxygen

```
#include "StringUtil.hpp"
```

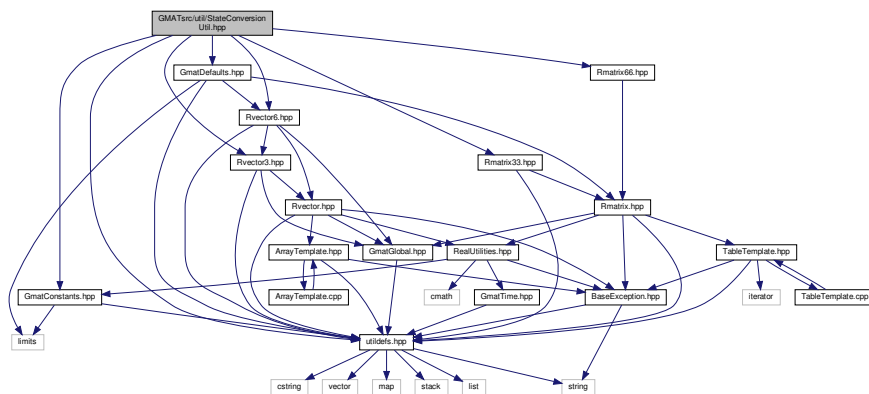
Include dependency graph for StateConversionUtil.cpp:



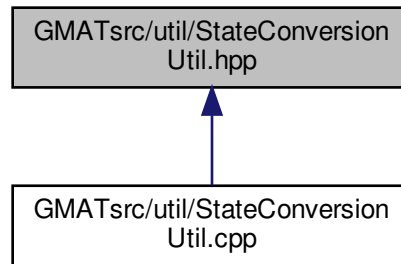
## 9.83 GMATsrc/util/StateConversionUtil.hpp File Reference

```
#include "utildefs.hpp"
#include "GmatConstants.hpp"
#include "GmatDefaults.hpp"
#include "Rmatrix33.hpp"
#include "Rmatrix66.hpp"
#include "Rvector3.hpp"
#include "Rvector6.hpp"
```

Include dependency graph for StateConversionUtil.hpp:



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



## Classes

- class [StateConversionUtil](#)

## Macros

- `#define EARTH_MU GmatSolarSystemDefaults::PLANET_MU[GmatSolarSystemDefaults::EARTH]`
- `#define EARTH_FLATTENING GmatSolarSystemDefaults::PLANET_FLATTENING[GmatSolarSystemDefaults::EARTH]`
- `#define EARTH_EQ_RADIUS GmatSolarSystemDefaults::PLANET_EQUATORIAL_RADIUS[GmatSolarSystemDefaults::EARTH]`

### 9.83.1 Macro Definition Documentation

#### 9.83.1.1 EARTH\_EQ\_RADIUS

```
#define EARTH_EQ_RADIUS GmatSolarSystemDefaults::PLANET_EQUATORIAL_RADIUS[GmatSolarSystemDefaults::EARTH]
```

#### 9.83.1.2 EARTH\_FLATTENING

```
#define EARTH_FLATTENING GmatSolarSystemDefaults::PLANET_FLATTENING[GmatSolarSystemDefaults::EARTH]
```

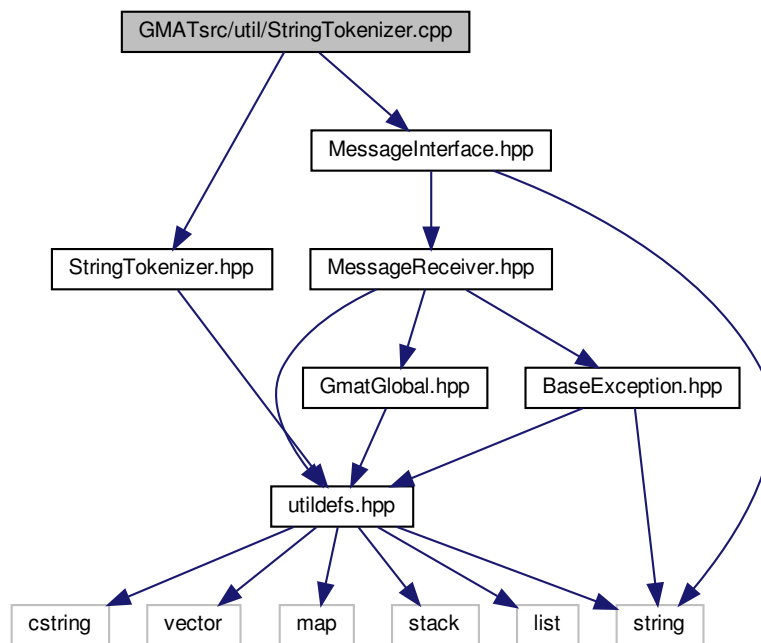
## 9.83.1.3 EARTH\_MU

```
#define EARTH_MU GmatSolarSystemDefaults::PLANET_MU[GmatSolarSystemDefaults::EARTH]
```

## 9.84 GMATsrc/util/StringTokenizer.cpp File Reference

```
#include "StringTokenizer.hpp"
#include "MessageInterface.hpp"
```

Include dependency graph for StringTokenizer.cpp:

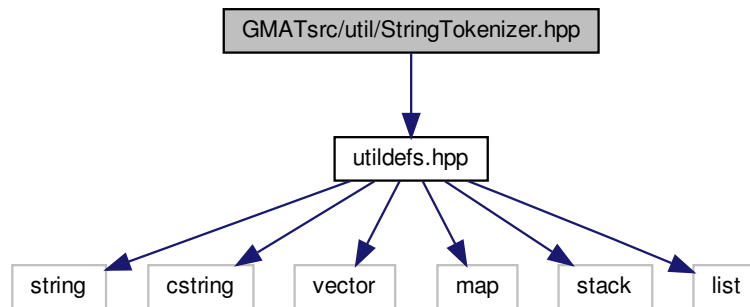


## 9.85 GMATsrc/util/StringTokenizer.hpp File Reference

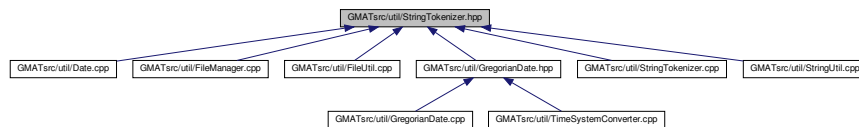
```
#include "utildefs.hpp"
```



Include dependency graph for StringTokenizer.hpp:



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



## Classes

- class [StringTokenizer](#)

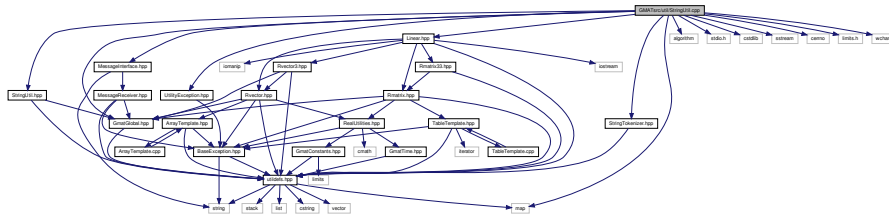
## 9.86 GMATsrc/util/StringUtil.cpp File Reference

```

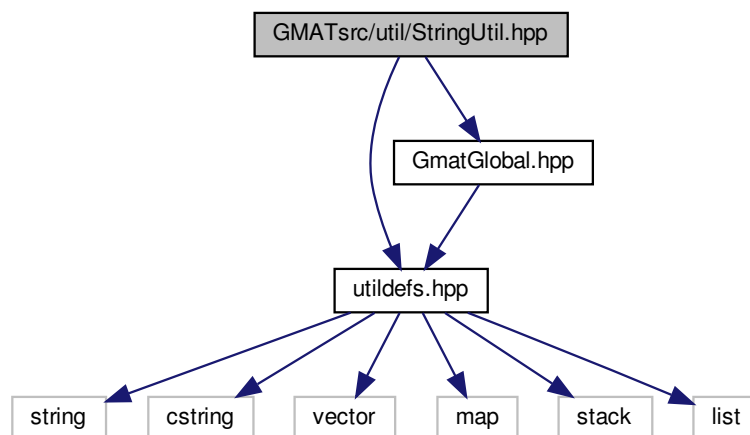
#include "StringUtil.hpp"
#include "MessageInterface.hpp"
#include "UtilityException.hpp"
#include "GmatGlobal.hpp"
#include "Linear.hpp"
#include "StringTokenizer.hpp"
#include <map>
#include <algorithm>
#include <stdio.h>
#include <cstdlib>
#include <sstream>
#include <cerrno>
#include <limits.h>

```

Include dependency graph for StringUtil.cpp:



```
#include "utildefs.hpp"
#include "GmatGlobal.hpp"
Include dependency graph for StringUtil.hpp:
```



- GmatStringUtil

## Enumerations

- enum `GmatStringUtil::StripType` { `GmatStringUtil::LEADING` = 1, `GmatStringUtil::TRAILING` = 2, `GmatStringUtil::BOTH` = 3 }
- enum `GmatStringUtil::AlignmentType` { `GmatStringUtil::LEFT`, `GmatStringUtil::RIGHT`, `GmatStringUtil::CENTER` }

## Functions

- `GMATUTIL_API` `std::string GmatStringUtil::RemoveAll` (const `std::string` &str, char ch, `Integer` start=0)
- `GMATUTIL_API` `std::string GmatStringUtil::RemoveAll` (const `std::string` &str, const `std::string` &removeStr, `Integer` start=0)
- `GMATUTIL_API` `std::string GmatStringUtil::RemoveAllBlanks` (const `std::string` &str, bool ignoreSingleQuotes=false)
- `GMATUTIL_API` `std::string GmatStringUtil::RemoveLastNumber` (const `std::string` &str, `Integer` &lastNumber)
- `GMATUTIL_API` `std::string GmatStringUtil::RemoveLastString` (const `std::string` &str, const `std::string` &lastStr, bool removeAll=false)
- `GMATUTIL_API` `std::string GmatStringUtil::RemoveSpaceInBrackets` (const `std::string` &str, const `std::string` &bracketPair)
- `GMATUTIL_API` `std::string GmatStringUtil::RemoveMultipleSpaces` (const `std::string` &str)
- `GMATUTIL_API` `std::string GmatStringUtil::RemoveTrailingZeros` (`Real` val, const `std::string` &valStr, `Integer` iterCount=0)
- `GMATUTIL_API` `std::string GmatStringUtil::RemoveScientificNotation` (const `std::string` &str)
- `GMATUTIL_API` `std::string GmatStringUtil::RemoveMathSymbols` (const `std::string` &str, bool removeMathOperator=false)
- `GMATUTIL_API` `std::string GmatStringUtil::PadWithBlanks` (const `std::string` &str, `Integer` toSize, `StripType` whichEnd=TRAILING)
- `GMATUTIL_API` `std::string GmatStringUtil::BuildNumber` (`Real` value, bool useExp=false, `Integer` length=17)
- `GMATUTIL_API` `std::string GmatStringUtil::Trim` (const `std::string` &str, `StripType` stype=BOTH, bool removeSemicolon=false, bool removeEol=false)
- `GMATUTIL_API` `std::string GmatStringUtil::Strip` (const `std::string` &str, `StripType` stype=BOTH)
- `GMATUTIL_API` `std::string GmatStringUtil::ToUpper` (const `std::string` &str, bool firstLetterOnly=false)
- `GMATUTIL_API` `std::string GmatStringUtil::ToLower` (const `std::string` &str, bool firstLetterOnly=false)
- `GMATUTIL_API` `std::string GmatStringUtil::Capitalize` (const `std::string` &str)
- `GMATUTIL_API` `std::string GmatStringUtil::ReplaceFirst` (const `std::string` &str, const `std::string` &from, const `std::string` &to, `std::string::size_type` startIdx=0)
- `GMATUTIL_API` `std::string GmatStringUtil::Replace` (const `std::string` &str, const `std::string` &from, const `std::string` &to, `std::string::size_type` startIdx=0)
- `GMATUTIL_API` `std::string GmatStringUtil::ReplaceName` (const `std::string` &str, const `std::string` &from, const `std::string` &to)
- `GMATUTIL_API` `std::string GmatStringUtil::ReplaceNumber` (const `std::string` &str, const `std::string` &from, const `std::string` &to)
- `GMATUTIL_API` `std::string GmatStringUtil::ReplaceChainedUnaryOperators` (const `std::string` &str)
- `GMATUTIL_API` `std::string GmatStringUtil::RealToString` (const `Real` &val, bool useCurrentFormat=true, bool scientific=false, bool showPoint=true, `Integer` precision=`GmatGlobal::DATA_PRECISION`, `Integer` width=`GmatGlobal::DATA_WIDTH`)
- `GMATUTIL_API` `std::string GmatStringUtil::RealToString` (const `Real` &val, `Integer` precision, bool showPoint=false, `Integer` width=1)
- `GMATUTIL_API` `std::string GmatStringUtil::ToString` (const bool &val)
- `GMATUTIL_API` `std::string GmatStringUtil::ToString` (const `Real` &val, `Integer` precision, bool showPoint=false, `Integer` width=1)
- `GMATUTIL_API` `std::string GmatStringUtil::ToString` (const `Integer` &val, `Integer` width)
- `GMATUTIL_API` `std::string GmatStringUtil::ToString` (const `Real` &val, bool useCurrentFormat=true, bool scientific=false, bool showPoint=true, `Integer` precision=`GmatGlobal::DATA_PRECISION`, `Integer` width=`GmatGlobal::DATA_WIDTH`)

- [GMATUTIL\\_API](#) `std::string GmatStringUtil::ToString` (const [Integer](#) &val, bool useCurrentFormat=true, [Integer](#) width=[GmatGlobal::INTEGER\\_WIDTH](#))
- [GMATUTIL\\_API](#) `std::string GmatStringUtil::ToStringNoZeros` (const [Real](#) &val)
- [GMATUTIL\\_API](#) `std::string GmatStringUtil::ToOrdinal` ([Integer](#) i, bool textOnly=false)
- [GMATUTIL\\_API](#) `std::string GmatStringUtil::RemoveExtraParen` (const `std::string` &str, bool ignore←Comma=false, bool ignoreSingleQuotes=false)
- [GMATUTIL\\_API](#) `std::string GmatStringUtil::RemoveOuterParen` (const `std::string` &str)
- [GMATUTIL\\_API](#) `std::string GmatStringUtil::RemoveOuterString` (const `std::string` &str, const `std::string` &start, const `std::string` &end)
- [GMATUTIL\\_API](#) `std::string GmatStringUtil::RemoveEnclosingString` (const `std::string` &str, const `std::string` &enStr)
- [GMATUTIL\\_API](#) `std::string GmatStringUtil::RemoveInlineComment` (const `std::string` &str, const `std::string` &cmStr)
- [GMATUTIL\\_API](#) `std::string GmatStringUtil::MakeCommentLines` (const `std::string` &str, bool breakAtCr=false)
- [GMATUTIL\\_API](#) `std::string GmatStringUtil::ParseFunctionName` (const `std::string` &str, `std::string` &argStr)
- [GMATUTIL\\_API](#) `StringArray GmatStringUtil::ParseFunctionCall` (const `std::string` &str)
- [GMATUTIL\\_API](#) `std::string GmatStringUtil::AddEnclosingString` (const `std::string` &str, const `std::string` &en←Str)
- [GMATUTIL\\_API](#) `std::string GmatStringUtil::GetInvalidNameMessageFormat` ()
- [GMATUTIL\\_API](#) `char GmatStringUtil::GetClosingBracket` (const `char` &openBracket)
- [GMATUTIL\\_API](#) `StringArray GmatStringUtil::SeparateBrackets` (const `std::string` &chunk, const `std::string` &bracketPair, const `std::string` &delim, bool checkOuterBracket=true)
- [GMATUTIL\\_API](#) `StringArray GmatStringUtil::SeparateBy` (const `std::string` &str, const `std::string` &delim, bool putBracketsTogether=false, bool insertDelim=false, bool insertComma=true)
- [GMATUTIL\\_API](#) `StringArray GmatStringUtil::SeparateByComma` (const `std::string` &str, bool checkSingle←Quote=true)
- [GMATUTIL\\_API](#) `StringArray GmatStringUtil::SeparateDots` (const `std::string` &str)
- [GMATUTIL\\_API](#) `StringArray GmatStringUtil::DecomposeBy` (const `std::string` &str, const `std::string` &delim)
- [GMATUTIL\\_API](#) `bool GmatStringUtil::IsNumber` (const `std::string` &str)
- [GMATUTIL\\_API](#) `bool GmatStringUtil::IsValidReal` (const `std::string` &str, [Real](#) &value, [Integer](#) &errorCode, bool trimParens=false, bool allowOverflow=true)
- [GMATUTIL\\_API](#) `bool GmatStringUtil::ToReal` (const `char` \*str, [Real](#) \*value, bool trimParens=false, bool allowOverflow=true)
- [GMATUTIL\\_API](#) `bool GmatStringUtil::ToReal` (const `std::string` &str, [Real](#) \*value, bool trimParens=false, bool allowOverflow=true)
- [GMATUTIL\\_API](#) `bool GmatStringUtil::ToReal` (const `char` \*str, [Real](#) &value, bool trimParens=false, bool allowOverflow=true)
- [GMATUTIL\\_API](#) `bool GmatStringUtil::ToReal` (const `std::string` &str, [Real](#) &value, bool trimParens=false, bool allowOverflow=true)
- [GMATUTIL\\_API](#) `bool GmatStringUtil::ToInteger` (const `std::string` &str, [Integer](#) \*value, bool trimParens=false, bool allowOverflow=true)
- [GMATUTIL\\_API](#) `bool GmatStringUtil::ToInteger` (const `char` \*str, [Integer](#) &value, bool trimParens=false, bool allowOverflow=true)
- [GMATUTIL\\_API](#) `bool GmatStringUtil::ToInteger` (const `std::string` &str, [Integer](#) &value, bool trimParens=false, bool allowOverflow=true)
- [GMATUTIL\\_API](#) `bool GmatStringUtil::ToUnsignedInt` (const `std::string` &str, [UnsignedInt](#) \*value, bool trim←Parens=false, bool allowOverflow=true)
- [GMATUTIL\\_API](#) `bool GmatStringUtil::ToUnsignedInt` (const `std::string` &str, [UnsignedInt](#) &value, bool trim←Parens=false, bool allowOverflow=true)
- [GMATUTIL\\_API](#) `bool GmatStringUtil::ToBoolean` (const `std::string` &str, bool \*value, bool trimParens=false)
- [GMATUTIL\\_API](#) `bool GmatStringUtil::ToBoolean` (const `std::string` &str, bool &value, bool trimParens=false)
- [GMATUTIL\\_API](#) `bool GmatStringUtil::ToOnOff` (const `std::string` &str, `std::string` &value, bool trim←Parens=false)
- [GMATUTIL\\_API](#) `RealArray GmatStringUtil::ToRealArray` (const `std::string` &str, bool allowOverflow=true, bool allowSemicolon=false)
- [GMATUTIL\\_API](#) `IntegerArray GmatStringUtil::ToIntegerArray` (const `char` \*str, bool allowOverflow=true)

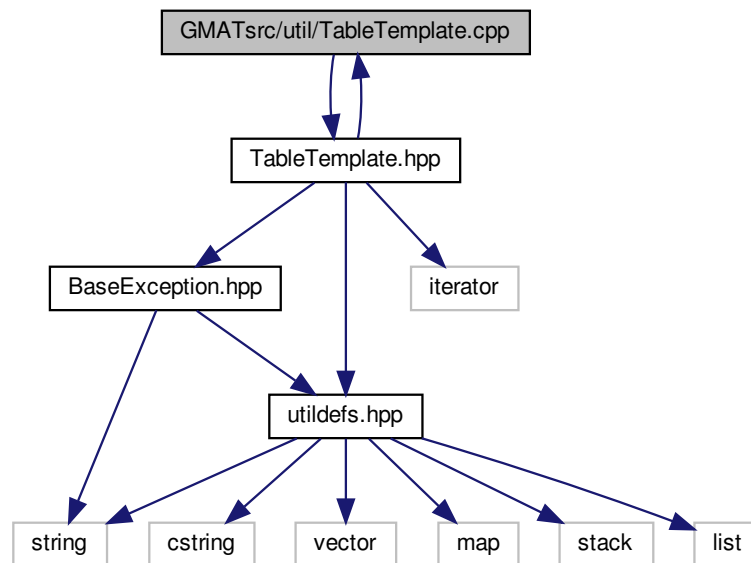
- [GMATUTIL\\_API IntegerArray GmatStringUtil::ToIntegerArray](#) (const std::string &str, bool allowOverflow=true)
- [GMATUTIL\\_API UnsignedIntArray GmatStringUtil::ToUnsignedIntArray](#) (const std::string &str, bool allowOverflow=true)
- [GMATUTIL\\_API StringArray GmatStringUtil::ToStringArray](#) (const std::string &str)
- [GMATUTIL\\_API BooleanArray GmatStringUtil::ToBooleanArray](#) (const std::string &str)
- [GMATUTIL\\_API void GmatStringUtil::ParseParameter](#) (const std::string &str, std::string &type, std::string &owner, std::string &dep)
- [GMATUTIL\\_API void GmatStringUtil::GetArrayCommaIndex](#) (const std::string &str, [Integer](#) &comma, const std::string &bracketPair="()")
- [GMATUTIL\\_API void GmatStringUtil::GetArrayIndexVar](#) (const std::string &str, std::string &rowStr, std::string &colStr, std::string &name, const std::string &bracketPair="()")
- [GMATUTIL\\_API void GmatStringUtil::GetArrayIndex](#) (const std::string &str, [Integer](#) &row, [Integer](#) &col, std::string &name, const std::string &bracketPair="()")
- [GMATUTIL\\_API void GmatStringUtil::GetArrayIndex](#) (const std::string &str, std::string &rowStr, std::string &colStr, [Integer](#) &row, [Integer](#) &col, std::string &name, const std::string &bracketPair="()")
- [GMATUTIL\\_API std::string GmatStringUtil::GetArrayName](#) (const std::string &str, const std::string &bracketPair="()")
- [GMATUTIL\\_API bool GmatStringUtil::IsOneElementArray](#) (const std::string &str)
- [GMATUTIL\\_API bool GmatStringUtil::IsSimpleArrayElement](#) (const std::string &str)
- [GMATUTIL\\_API void GmatStringUtil::FindFirstAndLast](#) (const std::string &str, char ch, [Integer](#) &first, [Integer](#) &last)
- [GMATUTIL\\_API void GmatStringUtil::FindParenMatch](#) (const std::string &str, [Integer](#) &open, [Integer](#) &close, bool &isOuterParen)
- [GMATUTIL\\_API void GmatStringUtil::FindMatchingParen](#) (const std::string &str, [Integer](#) &openParen, [Integer](#) &closeParen, bool &isOuterParen, [Integer](#) start=0)
- [GMATUTIL\\_API void GmatStringUtil::FindMatchingBracket](#) (const std::string &str, [Integer](#) &openBracket, [Integer](#) &closeBracket, bool &isOuterBracket, const std::string &bracket, [Integer](#) start=0)
- [GMATUTIL\\_API void GmatStringUtil::FindLastParenMatch](#) (const std::string &str, [Integer](#) &openParen, [Integer](#) &closeParen, [Integer](#) start=0)
- [GMATUTIL\\_API bool GmatStringUtil::IsEnclosedWith](#) (const std::string &str, const std::string &enclosingStr)
- [GMATUTIL\\_API bool GmatStringUtil::IsEnclosedWithExtraParen](#) (const std::string &str, bool checkOps=true, bool ignoreComma=false)
- [GMATUTIL\\_API bool GmatStringUtil::IsEnclosedWithBraces](#) (const std::string &str)
- [GMATUTIL\\_API bool GmatStringUtil::IsEnclosedWithBrackets](#) (const std::string &str)
- [GMATUTIL\\_API bool GmatStringUtil::IsBracketBalanced](#) (const std::string &str, const std::string &bracketPair)
- [GMATUTIL\\_API bool GmatStringUtil::IsParenBalanced](#) (const std::string &str)
- [GMATUTIL\\_API bool GmatStringUtil::IsParenEmpty](#) (const std::string &str)
- [GMATUTIL\\_API bool GmatStringUtil::AreAllBracketsBalanced](#) (const std::string &str, const std::string &allPairs)
- [GMATUTIL\\_API bool GmatStringUtil::AreAllNamesValid](#) (const std::string &str, bool blankNamesOk=false)
- [GMATUTIL\\_API bool GmatStringUtil::IsValidFunctionCall](#) (const std::string &str)
- [GMATUTIL\\_API bool GmatStringUtil::IsOuterParen](#) (const std::string &str)
- [GMATUTIL\\_API bool GmatStringUtil::IsCommaPartOfArray](#) (const std::string &str, [Integer](#) start=0)
- [GMATUTIL\\_API bool GmatStringUtil::IsBracketPartOfArray](#) (const std::string &str, const std::string &bracketPairs, bool checkOnlyFirst)
- [GMATUTIL\\_API bool GmatStringUtil::IsParenPartOfArray](#) (const std::string &str)
- [GMATUTIL\\_API bool GmatStringUtil::IsThereEqualSign](#) (const std::string &str)
- [GMATUTIL\\_API bool GmatStringUtil::IsThereMathSymbol](#) (const std::string &str)
- [GMATUTIL\\_API bool GmatStringUtil::HasNoBrackets](#) (const std::string &str, bool parensForArraysAllowed=true)
- [GMATUTIL\\_API bool GmatStringUtil::IsSingleItem](#) (const std::string &str)
- [GMATUTIL\\_API bool GmatStringUtil::StartsWith](#) (const std::string &str, const std::string &value)
- [GMATUTIL\\_API bool GmatStringUtil::EndsWith](#) (const std::string &str, const std::string &value)
- [GMATUTIL\\_API bool GmatStringUtil::EndsWithPathSeparator](#) (const std::string &str)
- [GMATUTIL\\_API bool GmatStringUtil::IsValidNumber](#) (const std::string &str, bool allowOverflow=true)

- `GMATUTIL_API` bool `GmatStringUtil::IsValidName` (const char \*str, bool ignoreBracket=false, bool blankNamesOk=false)
- `GMATUTIL_API` bool `GmatStringUtil::IsValidName` (const std::string &str, bool ignoreBracket=false, bool blankNamesOk=false)
- `GMATUTIL_API` bool `GmatStringUtil::IsValidParameterName` (const std::string &str)
- `GMATUTIL_API` bool `GmatStringUtil::IsLastNumberPartOfName` (const std::string &str)
- `GMATUTIL_API` bool `GmatStringUtil::IsBlank` (const std::string &str, bool ignoreEol=false)
- `GMATUTIL_API` bool `GmatStringUtil::HasMissingQuote` (const std::string &str, const std::string &quote, bool ignoreSpaceAfterQuote=true)
- `GMATUTIL_API` bool `GmatStringUtil::IsStringInsideSymbols` (const std::string &str, const std::string &reqStr, const std::string &symbol, std::string::size\_type &reqStrPos)
- `GMATUTIL_API` bool `GmatStringUtil::IsMathEquation` (const std::string &str, bool checkInvalidOpOnly=false, bool blankNamesOk=false)
- `GMATUTIL_API` bool `GmatStringUtil::IsMathOperator` (const char &ch)
- `GMATUTIL_API` Integer `GmatStringUtil::NumberOfOccurrences` (const std::string &str, const char c)
- `GMATUTIL_API` Integer `GmatStringUtil::NumberOfScientificNotation` (const std::string &str)
- `GMATUTIL_API` StringArray `GmatStringUtil::GetVarNames` (const std::string &str)
- `GMATUTIL_API` void `GmatStringUtil::WriteStringArray` (const StringArray &strArray, const std::string &desc="", const std::string &prefix="")
- `GMATUTIL_API` bool `GmatStringUtil::IsValidIdentity` (const std::string &str)
- `GMATUTIL_API` bool `GmatStringUtil::IsValidExtendedIdentity` (const std::string &str)
- `GMATUTIL_API` bool `GmatStringUtil::IsValidFileName` (const std::string &str)
- `GMATUTIL_API` bool `GmatStringUtil::IsValidFullFileName` (const std::string &str, Integer &error)
- `GMATUTIL_API` StringArray `GmatStringUtil::ParseName` (const std::string &extendedName)
- `GMATUTIL_API` std::string `GmatStringUtil::GetAlignmentString` (const std::string inputString, UnsignedInt len, AlignmentType alignment=LEFT)
- `GMATUTIL_API` std::wstring `GmatStringUtil::StringToWideString` (const std::string &str)
- `GMATUTIL_API` std::string `GmatStringUtil::WideStringToString` (const std::wstring &wstr)
- `GMATUTIL_API` std::string `GmatStringUtil::WideStringToString` (const wchar\_t \*wchar)

## 9.88 GMATsrc/util/TableTemplate.cpp File Reference

```
#include "TableTemplate.hpp"
```

Include dependency graph for TableTemplate.cpp:



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



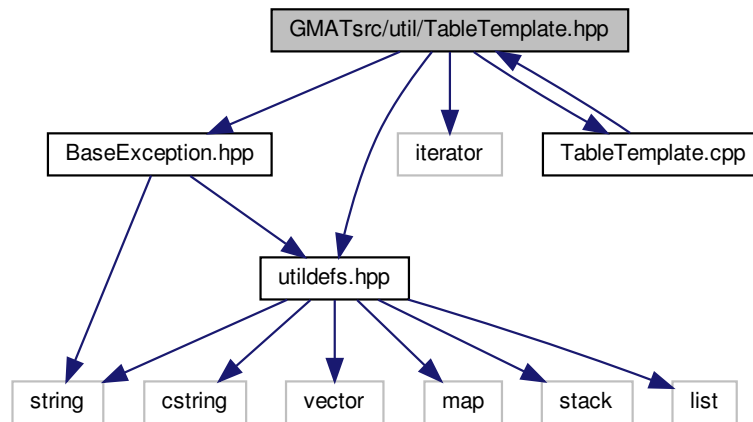
## 9.89 GMATsrc/util/TableTemplate.hpp File Reference

```

#include "utildefs.hpp"
#include "BaseException.hpp"
#include <iterator>
#include "TableTemplate.cpp"

```

Include dependency graph for TableTemplate.hpp:



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



## Classes

- class [TableTemplateExceptions](#)
- class [TableTemplateExceptions::DimensionError](#)
- class [TableTemplateExceptions::OutOfBounds](#)
- class [TableTemplateExceptions::UnsizeTable](#)
- class [TableTemplateExceptions::TableAlreadySized](#)
- class [TableTemplateExceptions::IllegalSize](#)
- class [TableTemplate< T >](#)

## 9.90 GMATsrc/util/TimeSystemConverter.cpp File Reference

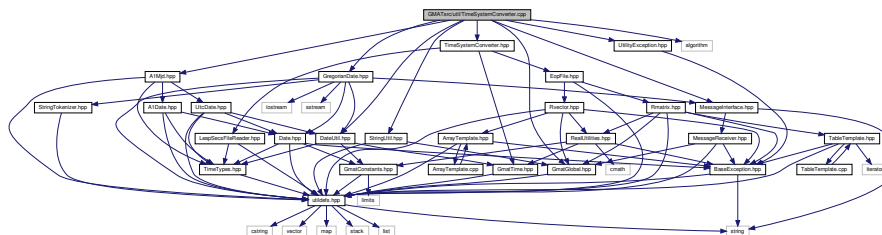
```

#include "TimeSystemConverter.hpp"
#include "AlMjd.hpp"
#include "GregorianDate.hpp"
#include "DateUtil.hpp"
#include "StringUtil.hpp"
#include "GmatGlobal.hpp"
#include "MessageInterface.hpp"
#include "UtilityException.hpp"

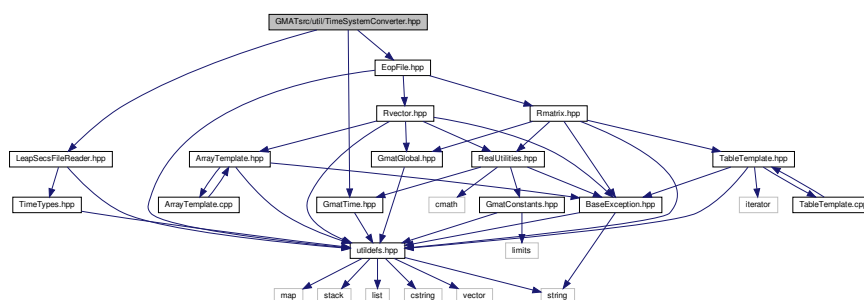
```



Include dependency graph for TimeSystemConverter.cpp:



```
#include "EopFile.hpp"
#include "LeapSecsFileReader.hpp"
#include "GmatTime.hpp"
Include dependency graph for TimeSystemConverter.hpp:
```



```

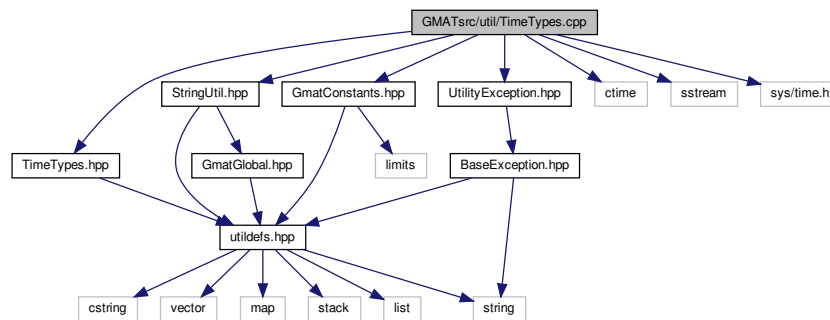
graph BT
 EopFile[GMatSrc/Util/EopFile.cpp] --> Header[GMatSrc/Util/TimeSystemConverter.hpp]
 TimeSystemConverter[GMatSrc/Util/TimeSystemConverter.cpp] --> Header
 UtcDate[GMatSrc/Util/UtcDate.cpp] --> Header

```

- class `UnimplementedException`
- class `TimeFileException`
- class `TimeFormatException`
- class `InvalidTimeException`
- class `TimeSystemConverter`

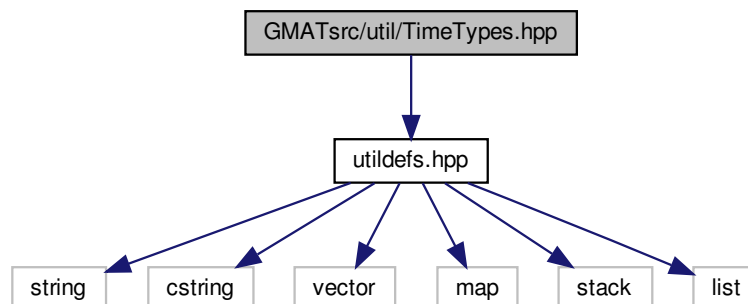
## 9.92 GMATsrc/util/TimeTypes.cpp File Reference

```
#include "TimeTypes.hpp"
#include "StringUtil.hpp"
#include "GmatConstants.hpp"
#include "UtilityException.hpp"
#include <ctime>
#include <sstream>
#include <sys/time.h>
Include dependency graph for TimeTypes.cpp:
```



## 9.93 GMATsrc/util/TimeTypes.hpp File Reference

```
#include "utildefs.hpp"
Include dependency graph for TimeTypes.hpp:
```



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



## Classes

- class [GmatTimeUtil::CalDate](#)
- class [GmatTimeUtil::ElapsedDate](#)

## Namespaces

- [GmatTimeUtil](#)

## Typedefs

- typedef [Real](#) [UtcMjd](#)
- typedef [Real](#) [Ut1Mjd](#)
- typedef [Integer](#) [YearNumber](#)
- typedef [Integer](#) [DayOfYear](#)
- typedef [Integer](#) [MonthOfYear](#)
- typedef [Integer](#) [DayOfMonth](#)
- typedef [Integer](#) [HourOfDay](#)
- typedef [Integer](#) [MinuteOfHour](#)

## Functions

- bool [GMATUTIL\\_API GmatTimeUtil::IsValidMonthName](#) (const std::string &str)
- std::string [GMATUTIL\\_API GmatTimeUtil::GetMonthName](#) ([Integer](#) month)
- [Integer](#) [GMATUTIL\\_API GmatTimeUtil::GetMonth](#) (const std::string &monthName)
- std::string [GMATUTIL\\_API GmatTimeUtil::FormatCurrentTime](#) ([Integer](#) format=1)
- std::string [GMATUTIL\\_API GmatTimeUtil::GetGregorianFormat](#) ()

### 9.93.1 Typedef Documentation

#### 9.93.1.1 DayOfMonth

```
typedef Integer DayOfMonth
```

#### 9.93.1.2 DayOfYear

```
typedef Integer DayOfYear
```

### 9.93.1.3 HourOfDay

```
typedef Integer HourOfDay
```

### 9.93.1.4 MinuteOfHour

```
typedef Integer MinuteOfHour
```

### 9.93.1.5 MonthOfYear

```
typedef Integer MonthOfYear
```

### 9.93.1.6 Ut1Mjd

```
typedef Real Ut1Mjd
```

### 9.93.1.7 UtcMjd

```
typedef Real UtcMjd
```

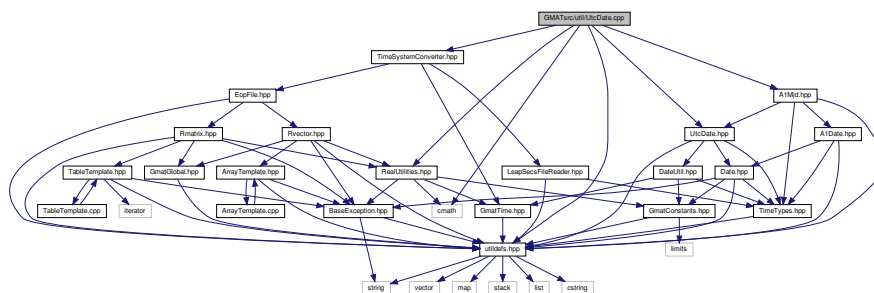
Provides declarations for date & time types.

### 9.93.1.8 YearNumber

```
typedef Integer YearNumber
```

## 9.94 GMATsrc/util/UtcDate.cpp File Reference

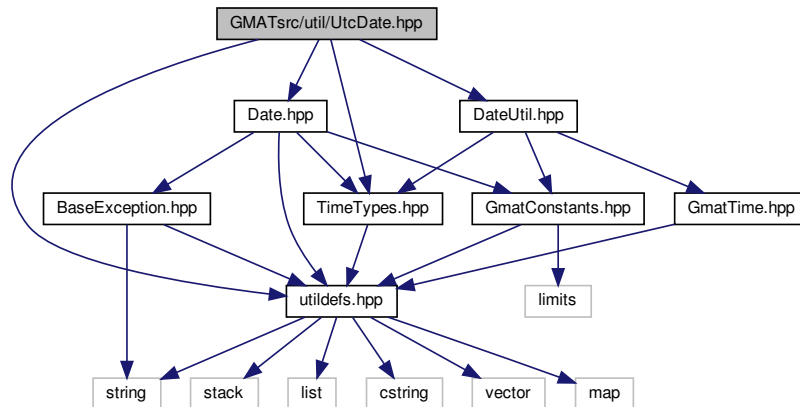
```
#include <cmath>
#include "utildefs.hpp"
#include "RealUtilities.hpp"
#include "AlMjd.hpp"
#include "UtcDate.hpp"
#include "TimeSystemConverter.hpp"
Include dependency graph for UtcDate.cpp:
```



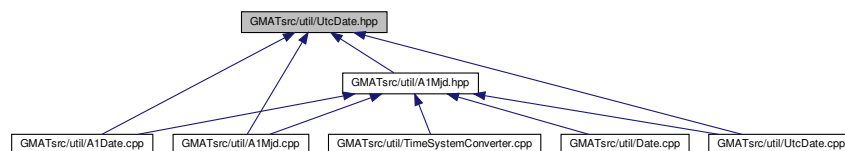
## 9.95 GMATsrc/util/UtcDate.hpp File Reference

```
#include "utildefs.hpp"
#include "TimeTypes.hpp"
#include "Date.hpp"
#include "DateUtil.hpp"
```

Include dependency graph for UtcDate.hpp:



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



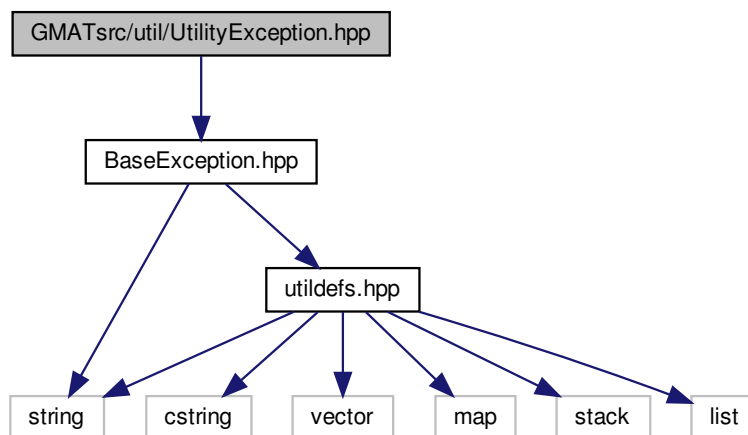
### Classes

- class [UtcDate](#)

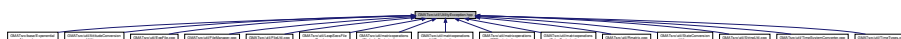
## 9.96 GMATsrc/util/UtilityException.hpp File Reference

```
#include "BaseException.hpp"
```

Include dependency graph for `UtilityException.hpp`:



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



## Classes

- class [UtilityException](#)
- class [GravityFileException](#)
- class [TimeException](#)

# Index

- ~A1Date
  - A1Date, [112](#)
- ~A1Mjd
  - A1Mjd, [115](#)
- ~ArrayTemplate
  - ArrayTemplate, [126](#)
- ~BaseException
  - BaseException, [139](#)
- ~CholeskyFactorization
  - CholeskyFactorization, [145](#)
- ~ConsoleAppException
  - ConsoleAppException, [148](#)
- ~Date
  - Date, [158](#)
- ~ElapsedTime
  - ElapsedTime, [177](#)
- ~EopFile
  - EopFile, [183](#)
- ~ExponentialAtmosphere
  - ExponentialAtmosphere, [189](#)
- ~FileManager
  - FileManager, [197](#)
- ~GmatTime
  - GmatTime, [232](#)
- ~GregorianDate
  - GregorianDate, [242](#)
- ~Interpolator
  - Interpolator, [251](#)
- ~InterpolatorException
  - InterpolatorException, [259](#)
- ~LUFactorization
  - LUFactorization, [277](#)
- ~LagrangeInterpolator
  - LagrangeInterpolator, [265](#)
- ~LeapSecsFileReader
  - LeapSecsFileReader, [273](#)
- ~MatrixFactorization
  - MatrixFactorization, [280](#)
- ~MessageReceiver
  - MessageReceiver, [289](#)
- ~QRFactorization
  - QRFactorization, [299](#)
- ~RandomNumber
  - RandomNumber, [304](#)
- ~Rmatrix
  - Rmatrix, [311](#)
- ~Rmatrix33
  - Rmatrix33, [323](#)
- ~Rmatrix66
  - Rmatrix66, [332](#)
- ~Rvector
  - Rvector, [340](#)
- ~Rvector3
  - Rvector3, [350](#)
- ~Rvector6
  - Rvector6, [358](#)
- ~SchurFactorization
  - SchurFactorization, [366](#)
- ~StringTokenizer
  - StringTokenizer, [403](#)
- ~TableTemplate
  - TableTemplate, [408](#)
- ~UtcDate
  - UtcDate, [435](#)
- A1\_TAI\_OFFSET
  - GmatTimeConstants, [103](#)
- A1Date, [109](#)
  - ~A1Date, [112](#)
  - A1Date, [110–112](#)
  - operator<, [112](#)
  - operator>, [112](#)
  - operator=, [112](#)
- A1MJD\_OF\_J2000
  - GmatTimeConstants, [103](#)
- A1Mjd, [113](#)
  - ~A1Mjd, [115](#)
  - A1Mjd, [114](#)
  - Clone, [115](#)
  - Get, [115](#)
  - GetDataDescriptions, [115](#)
  - GetNumData, [115](#)
  - GetReal, [116](#)
  - J2000, [121](#)
  - operator!=, [116](#)
  - operator<, [118](#)
  - operator<=, [118](#)
  - operator>, [119](#)
  - operator>=, [119](#)
  - operator+, [116](#)
  - operator+=, [117](#)
  - operator-, [117](#)
  - operator-=, [117](#)
  - operator=, [118](#)
  - operator==, [118](#)
  - Set, [119](#)
  - SetReal, [119](#)
  - Subtract, [120](#)
  - ToA1Date, [120](#)

- ToUtcDate, [120](#)
  - ToUtcMjd, [121](#)
  - ToValueStrings, [121](#)
  - UtcMjdToA1Mjd, [121](#)
- ABSOLUTE\_ZERO\_C
  - GmatPhysicalConstants, [50](#)
- ABSOLUTE\_ZERO\_K
  - GmatPhysicalConstants, [50](#)
- ACos
  - GmatMathUtil, [39](#)
- ACosh
  - GmatMathUtil, [39](#)
- ADRASTEIA\_NAME
  - GmatSolarSystemDefaults, [60](#)
- AMALTHEA\_NAME
  - GmatSolarSystemDefaults, [60](#)
- ARIEL\_NAME
  - GmatSolarSystemDefaults, [60](#)
- ASTRONOMICAL\_UNIT
  - GmatPhysicalConstants, [50](#)
- ASin
  - GmatMathUtil, [40](#)
- ASinh
  - GmatMathUtil, [40](#)
- ATLAS\_NAME
  - GmatSolarSystemDefaults, [60](#)
- ATan
  - GmatMathUtil, [40](#)
- ATan2
  - GmatMathUtil, [40](#)
- ATanh
  - GmatMathUtil, [41](#)
- Abs
  - GmatMathUtil, [39](#)
- actualSize
  - LagrangeInterpolator, [269](#)
- AddEnclosingString
  - GmatStringUtil, [79](#)
- AddGmatFunctionPath
  - FileManager, [197](#)
- AddGmatIncludePath
  - FileManager, [197](#)
- AddHiddenCommand
  - GmatGlobal, [216](#)
- AddMatlabFunctionPath
  - FileManager, [197](#), [198](#)
- AddPoint
  - Interpolator, [251](#)
  - LagrangeInterpolator, [265](#)
- AddPythonModulePath
  - FileManager, [198](#)
- AddSeconds
  - GmatTime, [233](#)
- AddToQR
  - QRFactorization, [300](#)
- AdjustSettings
  - FileManager, [198](#)
- AlignmentType
  - GmatStringUtil, [79](#)
- AllocateArrays
  - Interpolator, [252](#)
  - LagrangeInterpolator, [266](#)
- allowExtrapolation
  - Interpolator, [255](#)
- AltEquinoctialToEquinoctial
  - StateConversionUtil, [372](#)
- altitudeBands
  - ExponentialAtmosphere, [192](#)
- AnomalyType
  - StateConversionUtil, [371](#)
- AntiSymmetric
  - Rmatrix, [311](#)
  - Rmatrix33, [324](#)
  - Rmatrix66, [332](#)
- ArcsecToDeg
  - GmatMathUtil, [40](#)
- ArcsecToRad
  - GmatMathUtil, [40](#)
- AreAllBracketsBalanced
  - GmatStringUtil, [80](#)
- AreAllNamesValid
  - GmatStringUtil, [80](#)
- ArgumentError
  - RealUtilitiesExceptions::ArgumentError, [122](#)
- ArrayAlreadySized
  - ArrayTemplateExceptions::ArrayAlreadySized, [124](#)
- ArrayTemplate
  - ~ArrayTemplate, [126](#)
  - ArrayTemplate, [125](#), [126](#)
  - elementD, [129](#)
  - GetDataVector, [126](#)
  - GetElement, [126](#)
  - GetSize, [126](#)
  - init, [126](#)
  - IsSized, [127](#)
  - isSizedD, [129](#)
  - operator!=, [127](#)
  - operator(), [127](#)
  - operator=, [127](#)
  - operator==, [127](#)
  - operator[], [128](#)
  - Resize, [128](#)
  - SetElement, [128](#)
  - SetSize, [128](#)
  - sizeD, [129](#)
- ArrayTemplate< T >, [124](#)
- ArrayTemplateExceptions, [129](#)
- ArrayTemplateExceptions::ArrayAlreadySized, [123](#)
  - ArrayAlreadySized, [124](#)
- ArrayTemplateExceptions::DimensionError, [173](#)
  - DimensionError, [174](#)
- ArrayTemplateExceptions::IllegalSize, [247](#)
  - IllegalSize, [247](#)
- ArrayTemplateExceptions::OutOfBounds, [293](#)
  - OutOfBounds, [294](#)
- ArrayTemplateExceptions::UnSizedArray, [430](#)



- UnsizeArray, [431](#)
- AttitudeConversionUtility, [130](#)
  - DCMToEulerAxisAndAngle, [130](#)
  - EulerAxisAndAngleToDCM, [131](#)
  - IsValidEulerSequence, [131](#)
  - ToAngularVelocity, [131](#)
  - ToCosineMatrix, [133](#), [134](#)
  - ToEulerAngleRates, [134](#)
  - ToEulerAngles, [135](#)
  - ToMRPs, [136](#)
  - ToQuaternion, [136](#), [137](#)
- BELINDA\_NAME
  - GmatSolarSystemDefaults, [60](#)
- BIANCA\_NAME
  - GmatSolarSystemDefaults, [60](#)
- BUFFER\_SIZE
  - GmatFileUtil, [37](#)
- BaseException, [137](#)
  - ~BaseException, [139](#)
  - BaseException, [139](#)
  - GetDetails, [140](#)
  - GetFullMessage, [140](#)
  - GetMessageType, [140](#)
  - IsFatal, [140](#)
  - MAX\_MESSAGE\_LENGTH, [141](#)
  - operator=, [140](#)
  - SetDetails, [140](#)
  - SetFatal, [141](#)
  - SetMessage, [141](#)
  - SetMessageType, [141](#)
- beginIndex
  - LagrangeInterpolator, [269](#)
- BodyFixedStateConverterUtil, [15](#)
  - CartesianToSpherical, [15](#)
  - CartesianToSphericalEllipsoid, [16](#)
  - Convert, [16](#), [17](#)
  - GetValidRepresentations, [17](#)
  - IsValidStateRepresentation, [17](#)
  - SphericalEllipsoidToCartesian, [17](#)
  - SphericalEllipsoidToSpherical, [19](#)
  - SphericalToCartesian, [19](#)
  - SphericalToSphericalEllipsoid, [20](#)
- BooleanArray
  - gmatdefs.hpp, [448](#)
  - utildefs.hpp, [454](#)
- BrouwerMeanLongToCartesian
  - StateConversionUtil, [372](#)
- BrouwerMeanLongToOsculatingElements
  - StateConversionUtil, [373](#)
- BrouwerMeanShortToCartesian
  - StateConversionUtil, [373](#)
- BrouwerMeanShortToOsculatingElements
  - StateConversionUtil, [373](#)
- bufferSize
  - Interpolator, [256](#)
- BuildDataPoints
  - LagrangeInterpolator, [266](#)
- BuildNumber
  - GmatStringUtil, [80](#)
- Byte
  - gmatdefs.hpp, [449](#)
  - utildefs.hpp, [454](#)
- c
  - GmatPhysicalConstants, [50](#)
- CALLISTO\_NAME
  - GmatSolarSystemDefaults, [61](#)
- CALYPSO\_NAME
  - GmatSolarSystemDefaults, [61](#)
- CHARON\_NAME
  - GmatSolarSystemDefaults, [61](#)
- CORDELIA\_NAME
  - GmatSolarSystemDefaults, [61](#)
- CRESSIDA\_NAME
  - GmatSolarSystemDefaults, [61](#)
- CalDate
  - GmatTimeUtil::CalDate, [142](#)
- CalculateEccentricAnomaly
  - StateConversionUtil, [374](#)
- Capitalize
  - GmatStringUtil, [80](#)
- CartesianToAOP
  - StateConversionUtil, [375](#)
- CartesianToAngularMomentum
  - StateConversionUtil, [374](#)
- CartesianToBrouwerMeanLong
  - StateConversionUtil, [375](#)
- CartesianToBrouwerMeanShort
  - StateConversionUtil, [375](#)
- CartesianToDirOfLineOfNode
  - StateConversionUtil, [376](#)
- CartesianToECC
  - StateConversionUtil, [376](#)
- CartesianToEA
  - StateConversionUtil, [376](#)
- CartesianToEccVector
  - StateConversionUtil, [376](#)
- CartesianToEquinoctial
  - StateConversionUtil, [376](#)
- CartesianToHA
  - StateConversionUtil, [377](#)
- CartesianToINC
  - StateConversionUtil, [377](#)
- CartesianToIncomingAsymptote
  - StateConversionUtil, [377](#)
- CartesianToKeplerian
  - StateConversionUtil, [377–379](#)
- CartesianToKeplerianDerivativeConversion
  - StateConversionUtil, [380](#)
- CartesianToKeplerianDerivativeConversion\_FiniteDiff
  - StateConversionUtil, [380](#)
- CartesianToKeplerianDerivativeConversionWithKepl↔
  - Input
    - StateConversionUtil, [380](#)
  - CartesianToKeplerianDerivativeConversionWithKepl↔
    - Input\_FiniteDiff
      - StateConversionUtil, [381](#)

- CartesianToMA
  - StateConversionUtil, [381](#)
- CartesianToModEquinoctial
  - StateConversionUtil, [381](#)
- CartesianToOutgoingAsymptote
  - StateConversionUtil, [382](#)
- CartesianToPlanetodetic
  - StateConversionUtil, [382](#)
- CartesianToRAAN
  - StateConversionUtil, [382](#)
- CartesianToRaCodec
  - GmatRealUtil, [54](#)
- CartesianToRaDec
  - GmatRealUtil, [54](#)
- CartesianToSMA
  - StateConversionUtil, [383](#)
- CartesianToSpherical
  - BodyFixedStateConverterUtil, [15](#)
- CartesianToSphericalAZFPA
  - StateConversionUtil, [383](#)
- CartesianToSphericalEllipsoid
  - BodyFixedStateConverterUtil, [16](#)
- CartesianToSphericalRADEC
  - StateConversionUtil, [383](#)
- CartesianToTA
  - StateConversionUtil, [383](#)
- Cbrt
  - GmatMathUtil, [41](#)
- Ceiling
  - GmatMathUtil, [41](#)
- ChangeSize
  - TableTemplate, [408](#)
- CholeskyFactorization, [144](#)
  - ~CholeskyFactorization, [145](#)
  - CholeskyFactorization, [145](#)
  - Factor, [145](#)
  - Invert, [145](#), [146](#)
  - operator=, [146](#)
- CleanupArrays
  - Interpolator, [252](#)
  - LagrangeInterpolator, [266](#)
- Clear
  - Interpolator, [252](#)
  - LagrangeInterpolator, [266](#)
- ClearGmatFunctionPath
  - FileManager, [198](#)
- ClearGmatIncludePath
  - FileManager, [198](#)
- ClearHiddenCommands
  - GmatGlobal, [216](#)
- ClearMatlabFunctionPath
  - FileManager, [199](#)
- ClearMessage
  - ConsoleMessageReceiver, [150](#)
  - MessageInterface, [283](#)
  - MessageReceiver, [289](#)
- ClearMessageQueue
  - ConsoleMessageReceiver, [150](#)
- MessageInterface, [283](#)
- MessageReceiver, [289](#)
- Clone
  - A1Mjd, [115](#)
  - ExponentialAtmosphere, [189](#)
  - GmatTime, [233](#)
  - Interpolator, [252](#)
  - LagrangeInterpolator, [266](#)
  - Rvector6, [358](#)
- CloseLogFile
  - ConsoleMessageReceiver, [150](#)
- coDeclinationD
  - GmatRealUtil::RaCodec, [302](#)
- Cofactor
  - Rmatrix, [312](#)
- ColorMap
  - gmatdefs.hpp, [449](#)
  - utildefs.hpp, [454](#)
- colsD
  - TableTemplate, [411](#)
- CompareLines
  - GmatFileUtil, [28](#)
- CompareNumericColumns
  - GmatFileUtil, [28](#), [29](#)
- CompareNumericLines
  - GmatFileUtil, [29](#)
- CompareTextLines
  - GmatFileUtil, [29](#), [30](#)
- CompressNormalMatrix
  - MatrixFactorization, [280](#)
- ComputeLongitudeLatitude
  - Rvector3, [350](#)
- ConsoleAppException, [147](#)
  - ~ConsoleAppException, [148](#)
  - ConsoleAppException, [148](#)
- ConsoleMessageReceiver, [148](#)
  - ClearMessage, [150](#)
  - ClearMessageQueue, [150](#)
  - CloseLogFile, [150](#)
  - GetLogEnable, [150](#)
  - GetLogFileName, [150](#)
  - GetMessage, [150](#)
  - Instance, [151](#)
  - LogMessage, [151](#)
  - OpenLogFile, [152](#)
  - PopupMessage, [152](#)
  - PutMessage, [153](#)
  - SetLogEnable, [153](#)
  - SetLogFile, [153](#)
  - SetLogPath, [153](#)
  - ShowMessage, [154](#)
- Convert
  - BodyFixedStateConverterUtil, [16](#), [17](#)
  - StateConversionUtil, [384](#)
  - TimeSystemConverter, [419–421](#)
- ConvertFromAltEquinoctial
  - StateConversionUtil, [386](#)
- ConvertFromBrouwerMeanLong

- StateConversionUtil, [386](#)
- ConvertFromBrouwerMeanShort
  - StateConversionUtil, [386](#)
- ConvertFromCartesian
  - StateConversionUtil, [386](#)
- ConvertFromDelaunay
  - StateConversionUtil, [387](#)
- ConvertFromEquinoctial
  - StateConversionUtil, [387](#)
- ConvertFromIncomingAsymptote
  - StateConversionUtil, [387](#)
- ConvertFromKeplerian
  - StateConversionUtil, [387](#)
- ConvertFromModEquinoctial
  - StateConversionUtil, [388](#)
- ConvertFromModKeplerian
  - StateConversionUtil, [388](#)
- ConvertFromOutgoingAsymptote
  - StateConversionUtil, [388](#)
- ConvertFromPlanetodetic
  - StateConversionUtil, [388](#)
- ConvertFromSphericalAZFPA
  - StateConversionUtil, [389](#)
- ConvertFromSphericalRADEC
  - StateConversionUtil, [389](#)
- ConvertFromTaiMjd
  - TimeSystemConverter, [422](#)
- ConvertFromTrueAnomaly
  - StateConversionUtil, [389](#)
- ConvertGregorianToMjd
  - TimeSystemConverter, [422](#)
- ConvertGregorianToMjdGT
  - TimeSystemConverter, [423](#)
- ConvertMjdToGregorian
  - TimeSystemConverter, [423](#)
- ConvertToAbsPath
  - FileManager, [199](#)
- ConvertToOsFileName
  - GmatFileUtil, [30](#)
- ConvertToTaiMjd
  - TimeSystemConverter, [424](#)
- ConvertToTrueAnomaly
  - StateConversionUtil, [389](#), [390](#)
- Copy
  - Rvector3, [350](#)
- CopyArrays
  - Interpolator, [252](#)
  - LagrangeInterpolator, [267](#)
- CopyFile
  - FileManager, [199](#)
- Cos
  - GmatMathUtil, [41](#)
- Cosh
  - GmatMathUtil, [41](#)
- CountTokens
  - StringTokenizer, [403](#)
- Cross
  - Rvector3, [354](#)
- Rvector3.cpp, [530](#)
- DATA\_DESCRIPTIONS
  - Date, [162](#)
- DATA\_PRECISION
  - GmatGlobal, [230](#)
- DATA\_WIDTH
  - GmatGlobal, [230](#)
- DAYS\_BEFORE\_MONTH
  - GmatTimeConstants, [104](#)
- DAYS\_IN\_MONTH
  - GmatTimeConstants, [104](#)
- DAYS\_PER\_JULIAN\_CENTURY
  - GmatTimeConstants, [104](#)
- DAYS\_PER\_SEC
  - GmatTimeConstants, [104](#)
- DAYS\_PER\_YEAR
  - GmatTimeConstants, [104](#)
- DCM\_ORTHONORMALITY\_TOLERANCE
  - GmatAttitudeConstants, [26](#)
- DCMToEulerAxisAndAngle
  - AttitudeConversionUtility, [130](#)
- DEFAULT\_TO\_NO\_CLONES
  - gmatdefs.hpp, [448](#)
- DEFAULT\_TO\_NO\_REFOBJECTS
  - gmatdefs.hpp, [448](#)
- DEIMOS\_NAME
  - GmatSolarSystemDefaults, [61](#)
- DESDEMONA\_NAME
  - GmatSolarSystemDefaults, [61](#)
- DESPINA\_NAME
  - GmatSolarSystemDefaults, [61](#)
- DIONE\_NAME
  - GmatSolarSystemDefaults, [62](#)
- dataIncreases
  - Interpolator, [256](#)
- dataIndex
  - LagrangeInterpolator, [269](#)
- Date, [155](#)
  - ~Date, [158](#)
  - DATA\_DESCRIPTIONS, [162](#)
  - Date, [156–158](#)
  - dayD, [162](#)
  - GetDataDescriptions, [158](#)
  - GetDay, [158](#)
  - GetDayName, [158](#)
  - GetDaysPerMonth, [158](#)
  - GetHour, [158](#)
  - GetMinute, [158](#)
  - GetMonth, [159](#)
  - GetMonthName, [159](#)
  - GetNumData, [159](#)
  - GetSecond, [159](#)
  - GetSecondsOfDay, [159](#)
  - GetYear, [159](#)
  - IsValid, [159](#)
  - mPackedString, [162](#)
  - monthD, [162](#)
  - NUM\_DATA, [162](#)

- operator<, 159
- operator>, 160
- secondsOfDayD, 162
- stringValue, 162
- ToDayOfYear, 160
- ToPackedCalendarReal, 160
- ToPackedCalendarString, 160
- ToPackedHHMMSS, 160
- ToPackedYYYYMMDD, 160
- ToValueStrings, 160
- ToYearDOYHourMinSec, 161
- ToYearMonthDayHourMinSec, 161
- yearD, 163
- Date::LeapYearError, 275
  - LeapYearError, 275
- Date::TimeRangeError, 415
  - TimeRangeError, 416
- DateUtil, 163
  - EARLIEST\_VALID\_GREGORIAN, 169
  - EARLIEST\_VALID\_MJD\_VALUE, 169
  - EARLIEST\_VALID\_MJD, 169
  - FormatGregorian, 164
  - IsLeapYear, 165
  - IsValidGregorian, 164
  - IsValidTime, 165
  - JulianDate, 165
  - JulianDay, 165
  - LATEST\_VALID\_GREGORIAN, 170
  - LATEST\_VALID\_MJD\_VALUE, 170
  - LATEST\_VALID\_MJD, 170
  - MAX\_DAY, 170
  - MAX\_HOUR, 170
  - MAX\_MINUTE, 170
  - MAX\_MONTH, 170
  - MAX\_SEC, 170
  - MAX\_YEAR, 171
  - MIN\_DAY, 171
  - MIN\_HOUR, 171
  - MIN\_MINUTE, 171
  - MIN\_MONTH, 171
  - MIN\_SEC, 171
  - MIN\_YEAR, 171
  - ModifiedJulianDate, 166
  - ModifiedJulianDateGT, 166
  - ToDOYFromYearMonthDay, 167
  - ToHMSFromSecondsOfDay, 167
  - ToMonthDayFromYearDOY, 167
  - ToSecondsOfDayFromHMS, 168
  - UnpackDate, 168
  - UnpackDateWithDOY, 168
  - UnpackTime, 169
- DateUtil.cpp
  - IsLeapYear, 469
  - IsValidTime, 469
  - JulianDate, 469
  - ModifiedJulianDate, 470
  - ModifiedJulianDateGT, 470
  - ToDOYFromYearMonthDay, 471
  - ToHMSFromSecondsOfDay, 471
  - ToMonthDayFromYearDOY, 471
  - ToSecondsOfDayFromHMS, 472
  - UnpackDate, 472
  - UnpackDateWithDOY, 472
  - UnpackTime, 473
- DateUtil.hpp
  - IsLeapYear, 474
  - IsValidTime, 474
  - JulianDate, 475
  - ModifiedJulianDate, 475
  - ModifiedJulianDateGT, 476
  - ToDOYFromYearMonthDay, 476
  - ToHMSFromSecondsOfDay, 477
  - ToMonthDayFromYearDOY, 477
  - ToSecondsOfDayFromHMS, 477
  - UnpackDate, 477
  - UnpackDateWithDOY, 478
  - UnpackTime, 478
- day
  - GmatTimeUtil::CalDate, 142
- DayName
  - GmatTimeConstants, 103
- DayOfMonth
  - TimeTypes.hpp, 547
- DayOfYear
  - TimeTypes.hpp, 547
- dayD
  - Date, 162
- Days
  - GmatTime, 239
- days
  - GmatTimeUtil::ElapsedDate, 176
- declinationD
  - GmatRealUtil::RaDec, 303
- DecomposeBy
  - GmatStringUtil, 81
- DefaultMoons
  - GmatSolarSystemDefaults, 59
- DefaultPlanets
  - GmatSolarSystemDefaults, 59
- Deg
  - GmatMathUtil, 42
- DegToRad
  - GmatMathUtil, 42
- DelaunayToKeplerian
  - StateConversionUtil, 390
- Density
  - ExponentialAtmosphere, 189
- dependent
  - Interpolator, 256
- Determinant
  - LUFactorization, 277
  - QRFactorization, 300
  - Rmatrix, 312
  - Rmatrix33, 324
  - Rmatrix66, 332
- Diagonal

- Rmatrix, [312](#)
- dimension
  - Interpolator, [256](#)
- DimensionError
  - ArrayTemplateExceptions::DimensionError, [174](#)
  - TableTemplateExceptions::DimensionError, [173](#)
- DivideByZero
  - Rmatrix::DivideByZero, [175](#)
- DoesDirectoryExist
  - FileManager, [199](#)
  - GmatFileUtil, [30](#)
- DoesFileExist
  - FileManager, [199](#)
  - GmatFileUtil, [30, 31](#)
- EARLIEST\_VALID\_GREGORIAN
  - DateUtil, [169](#)
- EARLIEST\_VALID\_MJD\_VALUE
  - DateUtil, [169](#)
- EARLIEST\_VALID\_MJD
  - DateUtil, [169](#)
- EARTH\_EQ\_RADIUS
  - StateConversionUtil.hpp, [535](#)
- EARTH\_FLATTENING
  - StateConversionUtil.hpp, [535](#)
- EARTH\_MU
  - StateConversionUtil.hpp, [535](#)
- EARTH\_NAME
  - GmatSolarSystemDefaults, [62](#)
- ENCELADUS\_NAME
  - GmatSolarSystemDefaults, [62](#)
- EPIMETHEUS\_NAME
  - GmatSolarSystemDefaults, [62](#)
- EULER\_ANGLE\_TOLERANCE
  - GmatAttitudeConstants, [26](#)
- EUROPA\_NAME
  - GmatSolarSystemDefaults, [62](#)
- EccentricToTrueAnomaly
  - StateConversionUtil, [390](#)
- EchoCommands
  - GmatGlobal, [216](#)
- ElapsedDate
  - GmatTimeUtil::ElapsedDate, [175, 176](#)
- ElapsedTime, [177](#)
  - ~ElapsedTime, [177](#)
- ElapsedTime, [177](#)
  - Get, [178](#)
  - GetDataDescriptions, [178](#)
  - GetNumData, [178](#)
  - operator!=, [178](#)
  - operator<, [179](#)
  - operator<=, [179](#)
  - operator>, [179](#)
  - operator>=, [179](#)
  - operator+, [178](#)
  - operator+=, [178](#)
  - operator-, [178](#)
  - operator-=, [179](#)
  - operator=, [179](#)
- operator==, [179](#)
- Set, [180](#)
- ToElapsedDate, [180](#)
- ToValueStrings, [180](#)
- Element, [180](#)
  - index, [180](#)
  - value, [181](#)
- ElementWiseDivide
  - Rmatrix, [312](#)
- ElementWiseMultiply
  - Rmatrix, [312](#)
- elementD
  - ArrayTemplate, [129](#)
  - TableTemplate, [411](#)
- endIndex
  - LagrangeInterpolator, [269](#)
- EndsWith
  - GmatStringUtil, [81](#)
- EndsWithPathSeparator
  - GmatStringUtil, [81](#)
- eopFType
  - EopFile, [185](#)
- EopFile, [181](#)
  - ~EopFile, [183](#)
  - eopFType, [185](#)
  - EopFile, [182](#)
  - eopFileName, [185](#)
  - GetFileName, [183](#)
  - GetPolarMotionAndLod, [183](#)
  - GetPolarMotionData, [184](#)
  - GetTimeRange, [184](#)
  - GetUt1UtcOffset, [184](#)
  - Initialize, [184](#)
  - IsBlank, [184](#)
  - isInitialized, [186](#)
  - lastIndex, [186](#)
  - lastOffset, [186](#)
  - lastTaiMjd, [186](#)
  - lastUtcJd, [186](#)
  - MAX\_TABLE\_SIZE, [186](#)
  - operator=, [185](#)
  - polarMotion, [186](#)
  - previousIndex, [187](#)
  - ResetEopFile, [185](#)
  - tableSz, [187](#)
  - taiTime, [187](#)
  - theTimeConverter, [187](#)
  - ut1UtcOffsets, [187](#)
- eopFileName
  - EopFile, [185](#)
- EopFileType
  - GmatEop, [26](#)
- EpochArray
  - gmatdefs.hpp, [449](#)
  - utildefs.hpp, [454](#)
- EquinoctialToAltEquinoctial
  - StateConversionUtil, [390](#)
- EquinoctialToCartesian

- StateConversionUtil, 392
- EulerAxisAndAngleToDCM
  - AttitudeConversionUtility, 131
- Exp
  - GmatMathUtil, 42
- Exp10
  - GmatMathUtil, 42
- ExpandNormalMatrixInverse
  - MatrixFactorization, 281
- ExponentialAtmosphere, 188
  - ~ExponentialAtmosphere, 189
  - altitudeBands, 192
  - Clone, 189
  - Density, 189
  - ExponentialAtmosphere, 189
  - FindBand, 190
  - GetScaleHeight, 190
  - operator=, 191
  - refDensity, 192
  - refHeight, 192
  - scaleHeight, 192
  - SetConstants, 191
  - Smooth, 191
  - smoothDensity, 192
- Factor
  - CholeskyFactorization, 145
  - LUFactorization, 278
  - MatrixFactorization, 281
  - QRFactorization, 300
  - SchurFactorization, 366
- FileManager, 193
  - ~FileManager, 197
  - AddGmatFunctionPath, 197
  - AddGmatIncludePath, 197
  - AddMatlabFunctionPath, 197, 198
  - AddPythonModulePath, 198
  - AdjustSettings, 198
  - ClearGmatFunctionPath, 198
  - ClearGmatIncludePath, 198
  - ClearMatlabFunctionPath, 199
  - ConvertToAbsPath, 199
  - CopyFile, 199
  - DoesDirectoryExist, 199
  - DoesFileExist, 199
  - FileType, 195
  - FindMainIconFile, 199
  - FindPath, 200
  - GetAbsPathname, 201
  - GetAllGmatFunctionPaths, 201
  - GetAllGmatIncludePaths, 202
  - GetAllMatlabFunctionPaths, 202
  - GetAllPythonModulePaths, 202
  - GetBinDirectory, 202
  - GetBody3dModelFile, 202
  - GetCurrentWorkingDirectory, 202
  - GetFilename, 203
  - GetFullPathname, 204
  - GetFullStartupFilePath, 205
  - GetGmatFunctionPath, 205
  - GetGmatIncludePath, 205
  - GetGmatWorkingDirectory, 205
  - GetLastFilePathMessage, 205
  - GetMatlabFunctionPath, 206
  - GetPathSeparator, 207
  - GetPathname, 206
  - GetPluginList, 207
  - GetRootPath, 207
  - GetStartupFileDir, 207
  - GetStartupFileName, 208
  - GetTextureMapFile, 208
  - Instance, 208
  - ReadStartupFile, 208
  - RenameFile, 208
  - SetAbsPathname, 209, 210
  - SetBinDirectory, 210
  - SetCurrentWorkingDirectory, 210
  - SetGmatWorkingDirectory, 211
  - ValidatePaths, 211
  - WriteStartupFile, 211
- FileType
  - FileManager, 195
- FindBand
  - ExponentialAtmosphere, 190
- FindFirstAndLast
  - GmatStringUtil, 81
- FindLastParenMatch
  - GmatStringUtil, 81
- FindMainIconFile
  - FileManager, 199
- FindMatchingBracket
  - GmatStringUtil, 82
- FindMatchingParen
  - GmatStringUtil, 82
- FindParenMatch
  - GmatStringUtil, 82
- FindPath
  - FileManager, 200
- FindStartingPoint
  - LagrangeInterpolator, 267
- firstId
  - Gmat::PluginResource, 296
- Fix
  - GmatMathUtil, 42
- Floor
  - GmatMathUtil, 43
- forceInterpolation
  - Interpolator, 256
- FormatCurrentTime
  - GmatTimeUtil, 107
- FormatGregorian
  - DateUtil, 164
- FracSec
  - GmatTime, 239
- GALATEA\_NAME
  - GmatSolarSystemDefaults, 62
- GANYMEDE\_NAME



- GmatSolarSystemDefaults, [62](#)
- GEOPARMS
  - gmatdefs.hpp, [449](#)
- GMAT\_API
  - gmatdefs.hpp, [448](#)
- GMATUTIL\_API
  - utildefs.hpp, [453](#)
- GMATsrc/base/ExponentialAtmosphere.cpp, [441](#)
- GMATsrc/base/ExponentialAtmosphere.hpp, [441](#)
- GMATsrc/console/ConsoleAppException.cpp, [442](#)
- GMATsrc/console/ConsoleAppException.hpp, [443](#)
- GMATsrc/console/ConsoleMessageReceiver.cpp, [444](#)
- GMATsrc/console/ConsoleMessageReceiver.hpp, [445](#)
- GMATsrc/include/gmatdefs.hpp, [446](#)
- GMATsrc/include/utildefs.hpp, [452](#)
- GMATsrc/util/A1Date.cpp, [457](#)
- GMATsrc/util/A1Date.hpp, [458](#)
- GMATsrc/util/A1Mjd.cpp, [458](#)
- GMATsrc/util/A1Mjd.hpp, [459](#)
- GMATsrc/util/ArrayTemplate.cpp, [460](#)
- GMATsrc/util/ArrayTemplate.hpp, [460](#)
- GMATsrc/util/AttitudeConversionUtility.cpp, [461](#)
- GMATsrc/util/AttitudeConversionUtility.hpp, [462](#)
- GMATsrc/util/BaseException.cpp, [463](#)
- GMATsrc/util/BaseException.hpp, [464](#)
- GMATsrc/util/BodyFixedStateConverter.cpp, [464](#)
- GMATsrc/util/BodyFixedStateConverter.hpp, [465](#)
- GMATsrc/util/Date.cpp, [467](#)
- GMATsrc/util/Date.hpp, [467](#)
- GMATsrc/util/DateUtil.cpp, [468](#)
- GMATsrc/util/DateUtil.hpp, [473](#)
- GMATsrc/util/ElapsedTime.cpp, [479](#)
- GMATsrc/util/ElapsedTime.hpp, [479](#)
- GMATsrc/util/EopFile.cpp, [480](#)
- GMATsrc/util/EopFile.hpp, [480](#)
- GMATsrc/util/FileManager.cpp, [481](#)
- GMATsrc/util/FileManager.hpp, [482](#)
- GMATsrc/util/FileTypes.hpp, [483](#)
- GMATsrc/util/FileUtil.cpp, [484](#)
- GMATsrc/util/FileUtil.hpp, [484](#)
- GMATsrc/util/GmatConstants.hpp, [486](#)
- GMATsrc/util/GmatDefaults.hpp, [488](#)
- GMATsrc/util/GmatGlobal.cpp, [491](#)
- GMATsrc/util/GmatGlobal.hpp, [492](#)
- GMATsrc/util/GmatTime.cpp, [493](#)
- GMATsrc/util/GmatTime.hpp, [493](#)
- GMATsrc/util/GregorianDate.cpp, [494](#)
- GMATsrc/util/GregorianDate.hpp, [495](#)
- GMATsrc/util/LeapSecsFileReader.cpp, [500](#)
- GMATsrc/util/LeapSecsFileReader.hpp, [501](#)
- GMATsrc/util/Linear.cpp, [502](#)
- GMATsrc/util/Linear.hpp, [503](#)
- GMATsrc/util/MessageInterface.cpp, [510](#)
- GMATsrc/util/MessageInterface.hpp, [511](#)
- GMATsrc/util/MessageReceiver.cpp, [512](#)
- GMATsrc/util/MessageReceiver.hpp, [513](#)
- GMATsrc/util/RandomNumber.cpp, [514](#)
- GMATsrc/util/RandomNumber.hpp, [514](#)
- GMATsrc/util/RealUtilities.cpp, [515](#)
- GMATsrc/util/RealUtilities.hpp, [516](#)
- GMATsrc/util/Rmatrix.cpp, [518](#)
- GMATsrc/util/Rmatrix.hpp, [521](#)
- GMATsrc/util/Rmatrix33.cpp, [521](#)
- GMATsrc/util/Rmatrix33.hpp, [523](#)
- GMATsrc/util/Rmatrix66.cpp, [524](#)
- GMATsrc/util/Rmatrix66.hpp, [525](#)
- GMATsrc/util/Rvector.cpp, [526](#)
- GMATsrc/util/Rvector.hpp, [528](#)
- GMATsrc/util/Rvector3.cpp, [529](#)
- GMATsrc/util/Rvector3.hpp, [531](#)
- GMATsrc/util/Rvector6.cpp, [532](#)
- GMATsrc/util/Rvector6.hpp, [532](#)
- GMATsrc/util/StateConversionUtil.cpp, [533](#)
- GMATsrc/util/StateConversionUtil.hpp, [534](#)
- GMATsrc/util/StringTokenizer.cpp, [536](#)
- GMATsrc/util/StringTokenizer.hpp, [536](#)
- GMATsrc/util/StringUtil.cpp, [537](#)
- GMATsrc/util/StringUtil.hpp, [538](#)
- GMATsrc/util/TableTemplate.cpp, [542](#)
- GMATsrc/util/TableTemplate.hpp, [543](#)
- GMATsrc/util/TimeSystemConverter.cpp, [544](#)
- GMATsrc/util/TimeSystemConverter.hpp, [545](#)
- GMATsrc/util/TimeTypes.cpp, [546](#)
- GMATsrc/util/TimeTypes.hpp, [546](#)
- GMATsrc/util/UtcDate.cpp, [548](#)
- GMATsrc/util/UtcDate.hpp, [549](#)
- GMATsrc/util/UtilityException.hpp, [549](#)
- GMATsrc/util/interpolator/Interpolator.cpp, [496](#)
- GMATsrc/util/interpolator/Interpolator.hpp, [497](#)
- GMATsrc/util/interpolator/InterpolatorException.cpp, [498](#)
- GMATsrc/util/interpolator/InterpolatorException.hpp, [498](#)
- GMATsrc/util/interpolator/LagrangeInterpolator.cpp, [499](#)
- GMATsrc/util/interpolator/LagrangeInterpolator.hpp, [500](#)
- GMATsrc/util/matrixoperations/CholeskyFactorization.↵  
cpp, [504](#)
- GMATsrc/util/matrixoperations/CholeskyFactorization.↵  
hpp, [504](#)
- GMATsrc/util/matrixoperations/LUFactorization.cpp, [505](#)
- GMATsrc/util/matrixoperations/LUFactorization.hpp, [505](#)
- GMATsrc/util/matrixoperations/MatrixFactorization.cpp, [506](#)
- GMATsrc/util/matrixoperations/MatrixFactorization.hpp, [507](#)
- GMATsrc/util/matrixoperations/QRFactorization.cpp, [507](#)
- GMATsrc/util/matrixoperations/QRFactorization.hpp, [508](#)
- GMATsrc/util/matrixoperations/SchurFactorization.cpp, [509](#)
- GMATsrc/util/matrixoperations/SchurFactorization.hpp, [509](#)
- Gaussian

- RandomNumber, [305](#)
- GaussianArray
  - RandomNumber, [305](#)
- geoparms, [211](#)
  - tkp, [212](#)
  - xtemp, [212](#)
- Get
  - A1Mjd, [115](#)
  - ElapsedTime, [178](#)
  - Rvector3, [351](#)
  - Rvector6, [358](#)
- GetAbsPathname
  - FileManager, [201](#)
- GetActualFormat
  - GmatGlobal, [216](#)
- GetAlignmentString
  - GmatStringUtil, [82](#)
- GetAllGmatFunctionPaths
  - FileManager, [201](#)
- GetAllGmatIncludePaths
  - FileManager, [202](#)
- GetAllMatlabFunctionPaths
  - FileManager, [202](#)
- GetAllPythonModulePaths
  - FileManager, [202](#)
- GetAllTokens
  - StringTokenizer, [403](#)
- GetAnomalyLongText
  - StateConversionUtil, [392](#)
- GetAnomalyShortText
  - StateConversionUtil, [392](#)
- GetAnomalyType
  - StateConversionUtil, [393](#)
- GetApplicationPath
  - GmatFileUtil, [31](#)
- GetArrayCommaIndex
  - GmatStringUtil, [82](#)
- GetArrayIndex
  - GmatStringUtil, [83](#)
- GetArrayIndexVar
  - GmatStringUtil, [83](#)
- GetArrayName
  - GmatStringUtil, [83](#)
- GetBinDirectory
  - FileManager, [202](#)
- GetBody3dModelFile
  - FileManager, [202](#)
- GetBufferSize
  - Interpolator, [253](#)
- GetClosingBracket
  - GmatStringUtil, [84](#)
- GetColumn
  - Rmatrix, [312](#)
- GetCurrentWorkingDirectory
  - FileManager, [202](#)
  - GmatFileUtil, [31](#)
- GetDataDescriptions
  - A1Mjd, [115](#)
- Date, [158](#)
- ElapsedTime, [178](#)
- Rmatrix33, [324](#)
- Rvector3, [351](#)
- Rvector6, [358](#)
- GetDataPrecision
  - GmatGlobal, [217](#)
- GetDataVector
  - ArrayTemplate, [126](#)
  - TableTemplate, [408](#)
- GetDataWidth
  - GmatGlobal, [217](#)
- GetDate
  - GregorianCalendar, [242](#)
- GetDay
  - Date, [158](#)
- GetDayName
  - Date, [158](#)
- GetDays
  - GmatTime, [233](#)
- GetDaysPerMonth
  - Date, [158](#)
- GetDelimiters
  - StringTokenizer, [403](#)
- GetDetailedRunState
  - GmatGlobal, [217](#)
- GetDetails
  - BaseException, [140](#)
- GetElement
  - ArrayTemplate, [126](#)
  - TableTemplate, [409](#)
- GetEopFile
  - GmatGlobal, [217](#)
- GetFileListFromDirectory
  - GmatFileUtil, [31](#)
- GetFileName
  - EopFile, [183](#)
- GetFilename
  - FileManager, [203](#)
- GetFirstLeapSecondMJD
  - LeapSecsFileReader, [273](#)
  - TimeSystemConverter, [424](#)
- GetForceInterpolation
  - Interpolator, [253](#)
- GetFracSec
  - GmatTime, [233](#)
- GetFullMessage
  - BaseException, [140](#)
- GetFullPathname
  - FileManager, [204](#)
- GetFullStartupFilePath
  - FileManager, [205](#)
- GetFunctionOutputTypes
  - GmatFileUtil, [31](#)
- GetGmatFunctionPath
  - FileManager, [205](#)
- GetGmatIncludePath
  - FileManager, [205](#)



- GetGmatPath
  - GmatFileUtil, [31](#)
- GetGmatVersion
  - GmatGlobal, [217](#)
- GetGmatWorkingDirectory
  - FileManager, [205](#)
- GetGregorianFormat
  - GmatTimeUtil, [107](#)
- GetGuiMode
  - GmatGlobal, [217](#)
- GetHiddenCommands
  - GmatGlobal, [217](#)
- GetHour
  - Date, [158](#)
- GetIncludeFoundInScriptResource
  - GmatGlobal, [218](#)
- GetIntegerWidth
  - GmatGlobal, [218](#)
- GetInvalidFileNameMessage
  - GmatFileUtil, [32](#)
- GetInvalidNameMessageFormat
  - GmatStringUtil, [84](#)
- GetItrfCoefficientsFile
  - GmatGlobal, [218](#)
- GetLastFilePathMessage
  - FileManager, [205](#)
- GetLine
  - GmatFileUtil, [32](#)
- GetLogEnable
  - ConsoleMessageReceiver, [150](#)
  - MessageInterface, [283](#)
  - MessageReceiver, [289](#)
- GetLogFileName
  - ConsoleMessageReceiver, [150](#)
  - MessageInterface, [284](#)
  - MessageReceiver, [289](#)
- GetLogFileText
  - MessageReceiver, [290](#)
- GetLogfileName
  - GmatGlobal, [218](#)
- GetLogfileSource
  - GmatGlobal, [218](#)
- GetLongTypeNameList
  - StateConversionUtil, [393](#)
- GetMagnitude
  - Rvector, [340](#)
  - Rvector3, [351](#)
- GetMatlabFuncNameExt
  - GmatGlobal, [218](#)
- GetMatlabFunctionPath
  - FileManager, [206](#)
- GetMatlabMode
  - GmatGlobal, [219](#)
- GetMessage
  - ConsoleMessageReceiver, [150](#)
  - MessageReceiver, [290](#)
- GetMessageReceiver
  - MessageInterface, [284](#)
- GetMessageType
  - BaseException, [140](#)
- GetMinute
  - Date, [158](#)
- GetMjd
  - GmatTime, [233](#)
- GetMonth
  - Date, [159](#)
  - GmatTimeUtil, [107](#)
- GetMonthName
  - Date, [159](#)
  - GmatTimeUtil, [107](#)
- GetName
  - Interpolator, [253](#)
- GetNumColumns
  - TableTemplate, [409](#)
- GetNumData
  - A1Mjd, [115](#)
  - Date, [159](#)
  - ElapsedTime, [178](#)
  - Rvector3, [351](#)
  - Rvector6, [359](#)
- GetNumRows
  - TableTemplate, [409](#)
- GetOutputPath
  - GmatGlobal, [219](#)
- GetParameterMatrix
  - QRFactorization, [301](#)
- GetPathSeparator
  - FileManager, [207](#)
  - GmatFileUtil, [32](#)
- GetPathname
  - FileManager, [206](#)
- GetPlotMode
  - GmatGlobal, [219](#)
- GetPluginList
  - FileManager, [207](#)
- GetPointCount
  - Interpolator, [253](#)
- GetPolarMotionAndLod
  - EopFile, [183](#)
- GetPolarMotionData
  - EopFile, [184](#)
- GetQueuedMessage
  - MessageInterface, [284](#)
- GetRange
  - Interpolator, [253](#)
- GetReal
  - A1Mjd, [116](#)
- GetRealArray
  - Rvector, [341](#)
- GetRealColumns
  - GmatFileUtil, [32](#)
- GetRootPath
  - FileManager, [207](#)
- GetRow
  - Rmatrix, [313](#)
- GetRowOrColumn

- Rmatrix, 313
- GetRunInterrupted
  - GmatGlobal, 219
- GetRunMode
  - GmatGlobal, 219
- GetRunModeStartUp
  - GmatGlobal, 219
- GetRunState
  - GmatGlobal, 219
- GetScaleHeight
  - ExponentialAtmosphere, 190
- GetSec
  - GmatTime, 234
- GetSecond
  - Date, 159
- GetSecondsOfDay
  - Date, 159
- GetSize
  - ArrayTemplate, 126
  - TableTemplate, 409
- GetSpacing
  - GmatGlobal, 220
- GetStartupFileDir
  - FileManager, 207
- GetStartupFileName
  - FileManager, 208
- GetStateTypeList
  - StateConversionUtil, 393
- GetStringVals
  - Rmatrix, 313
- GetTemporaryDirectory
  - GmatFileUtil, 32
- GetTextLines
  - GmatFileUtil, 33
- GetTextureMapFile
  - FileManager, 208
- GetTimeInSec
  - GmatTime, 234
- GetTimePrecision
  - GmatGlobal, 220
- GetTimeRange
  - EopFile, 184
- GetTimeSystemAndFormat
  - TimeSystemConverter, 425
- GetTimeTypeID
  - TimeSystemConverter, 425
- GetTimeWidth
  - GmatGlobal, 220
- GetToken
  - StringTokenizer, 404
- GetType
  - GregorianDate, 242
- GetTypeCount
  - StateConversionUtil, 394
- GetUnitRvector
  - Rvector, 341
- GetUnitVector
  - Rvector3, 351
- GetUt1UtcOffset
  - EopFile, 184
- GetValidRepresentations
  - BodyFixedStateConverterUtil, 17
- GetValidTimeRepresentations
  - TimeSystemConverter, 425
- GetVarNames
  - GmatStringUtil, 84
- GetWritePersonalizationFile
  - GmatGlobal, 220
- GetYMDHMS
  - GregorianDate, 242
- GetYear
  - Date, 159
- GetR
  - Rvector6, 359
- GetV
  - Rvector6, 359
- Gmat, 20
  - MessageType, 21
  - ObjectType, 22
  - PLUGIN\_RESOURCE, 21
  - ParameterType, 23
  - RunState, 24
  - StateElementId, 24
  - WrapperDataType, 25
  - WriteMode, 25
- Gmat::PluginResource, 296
  - firstId, 296
  - handler, 296
  - lastId, 296
  - nodeName, 297
  - parentNodeName, 297
  - PluginResource, 296
  - subtype, 297
  - toolkit, 297
  - trigger, 297
  - type, 297
  - widgetType, 297
- GmatAttitudeConstants, 26
  - DCM\_ORTHONORMALITY\_TOLERANCE, 26
  - EULER\_ANGLE\_TOLERANCE, 26
  - QUAT\_MIN\_MAG, 26
- GmatEop, 26
  - EopFileType, 26
- GmatEpoch
  - gmatdefs.hpp, 449
  - utildefs.hpp, 454
- GmatFile, 27
- GmatFileUtil, 27
  - BUFFER\_SIZE, 37
  - CompareLines, 28
  - CompareNumericColumns, 28, 29
  - CompareNumericLines, 29
  - CompareTextLines, 29, 30
  - ConvertToOsFileName, 30
  - DoesDirectoryExist, 30
  - DoesFileExist, 30, 31

- GetApplicationPath, [31](#)
- GetCurrentWorkingDirectory, [31](#)
- GetFileListFromDirectory, [31](#)
- GetFunctionOutputTypes, [31](#)
- GetGmatPath, [31](#)
- GetInvalidFileNameMessage, [32](#)
- GetLine, [32](#)
- GetPathSeparator, [32](#)
- GetRealColumns, [32](#)
- GetTemporaryDirectory, [32](#)
- GetTextLines, [33](#)
- HasNoPath, [33](#)
- IsAppInstalled, [33](#)
- IsAsciiFile, [33](#)
- IsOsWindows, [33](#)
- IsPathAbsolute, [34](#)
- IsPathRelative, [34](#)
- IsSameFileName, [34](#), [35](#)
- IsValidFileName, [35](#)
- ParseFileExtension, [35](#)
- ParseFileName, [35](#)
- ParseFirstPathName, [36](#)
- ParsePathName, [36](#)
- PrepareCompare, [36](#)
- SetCurrentWorkingDirectory, [36](#)
- SkipHeaderLines, [37](#)
- GmatGlobal, [212](#)
  - AddHiddenCommand, [216](#)
  - ClearHiddenCommands, [216](#)
  - DATA\_PRECISION, [230](#)
  - DATA\_WIDTH, [230](#)
  - EchoCommands, [216](#)
  - GetActualFormat, [216](#)
  - GetDataPrecision, [217](#)
  - GetDataWidth, [217](#)
  - GetDetailedRunState, [217](#)
  - GetEopFile, [217](#)
  - GetGmatVersion, [217](#)
  - GetGuiMode, [217](#)
  - GetHiddenCommands, [217](#)
  - GetIncludeFoundInScriptResource, [218](#)
  - GetIntegerWidth, [218](#)
  - GetItrfCoefficientsFile, [218](#)
  - GetLogfileName, [218](#)
  - GetLogfileSource, [218](#)
  - GetMatlabFuncNameExt, [218](#)
  - GetMatlabMode, [219](#)
  - GetOutputPath, [219](#)
  - GetPlotMode, [219](#)
  - GetRunInterrupted, [219](#)
  - GetRunMode, [219](#)
  - GetRunModeStartUp, [219](#)
  - GetRunState, [219](#)
  - GetSpacing, [220](#)
  - GetTimePrecision, [220](#)
  - GetTimeWidth, [220](#)
  - GetWritePersonalizationFile, [220](#)
  - GuiMode, [215](#)
  - INTEGER\_WIDTH, [231](#)
  - Instance, [220](#)
  - IsBatchMode, [220](#)
  - IsBinaryIn, [220](#)
  - IsBinaryOut, [221](#)
  - IsEventLocationAvailable, [221](#)
  - IsUISavable, [221](#)
  - IsGmatCompiledIn64Bit, [221](#)
  - IsHiddenCommand, [221](#)
  - IsHorizontal, [221](#)
  - IsMatlabAvailable, [222](#)
  - IsMatlabDebugOn, [222](#)
  - IsMissionTreeDebugOn, [222](#)
  - IsNitsClient, [222](#)
  - IsScientific, [222](#)
  - IsWritingFilePathInfo, [222](#)
  - IsWritingGmatKeyword, [222](#)
  - IsWritingParameterInfo, [222](#)
  - LogfileSource, [215](#)
  - MatlabMode, [215](#)
  - PlotMode, [215](#)
  - RemoveHiddenCommand, [223](#)
  - RunMode, [216](#)
  - SetActualFormat, [223](#)
  - SetAppendEol, [223](#)
  - SetBatchMode, [223](#)
  - SetBinaryIn, [223](#)
  - SetBinaryOut, [223](#)
  - SetCommandEchoMode, [224](#)
  - SetCurrentFormat, [224](#)
  - SetDataPrecision, [224](#)
  - SetDataWidth, [224](#)
  - SetDefaultFormat, [224](#)
  - SetDetailedRunState, [225](#)
  - SetEopFile, [225](#)
  - SetEventLocationAvailable, [225](#)
  - SetGuiMode, [225](#)
  - SetHorizontal, [225](#)
  - SetIncludeFoundInScriptResource, [225](#)
  - SetIntegerWidth, [226](#)
  - SetItrfCoefficientsFile, [226](#)
  - SetLogfileName, [226](#)
  - SetLogfileSource, [226](#)
  - SetMatlabAvailable, [226](#)
  - SetMatlabDebug, [226](#)
  - SetMatlabFuncNameExt, [227](#)
  - SetMatlabMode, [227](#)
  - SetMissionTreeDebug, [227](#)
  - SetNitsClient, [227](#)
  - SetOutputPath, [227](#)
  - SetPlotMode, [227](#)
  - SetPrefix, [227](#)
  - SetRunInterrupted, [228](#)
  - SetRunMode, [228](#)
  - SetRunState, [228](#)
  - SetScientific, [228](#)
  - SetShowPoint, [228](#)
  - SetSkipSplashMode, [228](#)

- SetSpacing, [229](#)
- SetTimePrecision, [229](#)
- SetTimeWidth, [229](#)
- SetToCurrentFormat, [229](#)
- SetToDefaultFormat, [229](#)
- SetWriteFilePathInfo, [229](#)
- SetWriteGmatKeyword, [229](#)
- SetWriteParameterInfo, [230](#)
- SetWritePersonalizationFile, [230](#)
- ShowPoint, [230](#)
- SkipSplashMode, [230](#)
- TIME\_PRECISION, [231](#)
- TIME\_WIDTH, [231](#)
- GmatIntegerConstants, [37](#)
  - INTEGER\_UNDEFINED, [37](#)
- GmatMathConstants, [37](#)
  - SIGN, [37](#)
- GmatMathUtil, [38](#)
  - ACos, [39](#)
  - ACosh, [39](#)
  - ASin, [40](#)
  - ASinh, [40](#)
  - ATan, [40](#)
  - ATan2, [40](#)
  - ATanh, [41](#)
  - Abs, [39](#)
  - ArcsecToDeg, [40](#)
  - ArcsecToRad, [40](#)
  - Cbrt, [41](#)
  - Ceiling, [41](#)
  - Cos, [41](#)
  - Cosh, [41](#)
  - Deg, [42](#)
  - DegToRad, [42](#)
  - Exp, [42](#)
  - Exp10, [42](#)
  - Fix, [42](#)
  - Floor, [43](#)
  - IsEqual, [43](#)
  - IsEven, [43](#)
  - IsInf, [43](#)
  - IsNaN, [43](#)
  - IsNegative, [44](#)
  - IsNonNegative, [44](#)
  - IsOdd, [44](#)
  - IsPositive, [44](#)
  - IsZero, [44](#)
  - Ln, [44](#)
  - Log, [45](#)
  - Log10, [45](#)
  - Max, [46](#)
  - Min, [46](#)
  - Mod, [46](#)
  - NearestInt, [46](#)
  - Pow, [46](#)
  - Quotient, [47](#)
  - Rad, [47](#)
  - RadToDeg, [47](#)
  - Rand, [47](#)
  - Randn, [48](#)
  - Rem, [48](#)
  - Round, [48](#)
  - SetSeed, [48](#)
  - SignOf, [48](#)
  - Sin, [48](#)
  - SinXOverX, [49](#)
  - Sinh, [49](#)
  - Sqrt, [49](#)
  - Tan, [49](#)
  - Tanh, [49](#)
- GmatOrbitConstants, [50](#)
- GmatPhysicalConstants, [50](#)
  - ABSOLUTE\_ZERO\_C, [50](#)
  - ABSOLUTE\_ZERO\_K, [50](#)
  - ASTRONOMICAL\_UNIT, [50](#)
  - c, [50](#)
  - SPEED\_OF\_LIGHT\_VACUUM, [50](#)
  - UNIVERSAL\_GRAVITATIONAL\_CONSTANT, [50](#)
- GmatRealConstants, [51](#)
  - INTEGER\_MAX, [51](#)
  - INTEGER\_UNDEFINED, [51](#)
  - REAL\_DIG, [51](#)
  - REAL\_EPSILON, [52](#)
  - REAL\_MAX\_10\_EXP, [52](#)
  - REAL\_MAX\_EXP, [52](#)
  - REAL\_MAX, [52](#)
  - REAL\_MIN\_10\_EXP, [52](#)
  - REAL\_MIN\_EXP, [52](#)
  - REAL\_MIN, [52](#)
  - REAL\_TOL, [52](#)
  - REAL\_UNDEFINED\_LARGE, [53](#)
  - REAL\_UNDEFINED, [53](#)
  - SHORT\_REAL\_RADIX, [53](#)
- GmatRealUtil, [53](#)
  - CartesianToRaCodec, [54](#)
  - CartesianToRaDec, [54](#)
  - Max, [54](#)
  - Min, [54](#)
  - operator<<, [54](#), [55](#)
  - operator>>, [55](#)
  - RaCodecToCartesian, [55](#)
  - RaCodecToRaDec, [56](#)
  - RaDecToCartesian, [56](#)
  - RaDecToRaCodec, [56](#)
  - RealToString, [56](#)
  - ToString, [56](#)
- GmatRealUtil::RaCodec, [302](#)
  - coDeclinationD, [302](#)
  - radiusD, [302](#)
  - rightAscensionD, [303](#)
- GmatRealUtil::RaDec, [303](#)
  - declinationD, [303](#)
  - radiusD, [303](#)
  - rightAscensionD, [303](#)
- GmatSolarSystemDefaults, [57](#)
  - ADRASTEA\_NAME, [60](#)

AMALTHEA\_NAME, [60](#)  
ARIEL\_NAME, [60](#)  
ATLAS\_NAME, [60](#)  
BELINDA\_NAME, [60](#)  
BIANCA\_NAME, [60](#)  
CALLISTO\_NAME, [61](#)  
CALYPSO\_NAME, [61](#)  
CHARON\_NAME, [61](#)  
CORDELIA\_NAME, [61](#)  
CRESSIDA\_NAME, [61](#)  
DEIMOS\_NAME, [61](#)  
DESDEMONA\_NAME, [61](#)  
DESPINA\_NAME, [61](#)  
DIONE\_NAME, [62](#)  
DefaultMoons, [59](#)  
DefaultPlanets, [59](#)  
EARTH\_NAME, [62](#)  
ENCELADUS\_NAME, [62](#)  
EPIMETHEUS\_NAME, [62](#)  
EUROPA\_NAME, [62](#)  
GALATEA\_NAME, [62](#)  
GANYMEDE\_NAME, [62](#)  
HELENE\_NAME, [62](#)  
IAPETUS\_NAME, [63](#)  
IO\_NAME, [63](#)  
JANUS\_NAME, [63](#)  
JULIET\_NAME, [63](#)  
JUPITER\_NAME, [63](#)  
LARISSA\_NAME, [63](#)  
MARS\_NAME, [63](#)  
MERCURY\_NAME, [63](#)  
METIS\_NAME, [64](#)  
MIMAS\_NAME, [64](#)  
MIRANDA\_NAME, [64](#)  
MOON\_CENTRAL\_BODIES, [64](#)  
MOON\_EQUATORIAL\_RADIUS, [64](#)  
MOON\_FLATTENING, [64](#)  
MOON\_MU, [65](#)  
MOON\_NAIF\_IDS, [65](#)  
MOON\_NAMES, [65](#)  
MOON\_NAME, [65](#)  
MOON\_ORIENTATION\_PARAMETERS, [66](#)  
MOON\_SPICE\_FRAME\_ID, [66](#)  
MOON\_TWO\_BODY\_ELEMENTS, [66](#)  
MOON\_TWO\_BODY\_EPOCH, [66](#)  
NAIAD\_NAME, [67](#)  
NEPTUNE\_NAME, [67](#)  
OBERON\_NAME, [67](#)  
OPHELIA\_NAME, [67](#)  
PAN\_NAME, [67](#)  
PANDORA\_NAME, [67](#)  
PHOBOS\_NAME, [68](#)  
PHOEBE\_NAME, [68](#)  
PLANET\_EQUATORIAL\_RADIUS, [68](#)  
PLANET\_FLATTENING, [68](#)  
PLANET\_MU, [68](#)  
PLANET\_NAIF\_IDS, [69](#)  
PLANET\_NAMES, [69](#)  
PLANET\_ORIENTATION\_PARAMETERS, [69](#)  
PLANET\_SPICE\_FRAME\_ID, [70](#)  
PLANET\_TWO\_BODY\_ELEMENTS, [70](#)  
PLANET\_TWO\_BODY\_EPOCH, [70](#)  
PLUTO\_NAME, [71](#)  
PORTIA\_NAME, [71](#)  
PROMETHEUS\_NAME, [71](#)  
PROTEUS\_NAME, [71](#)  
PUCK\_NAME, [71](#)  
RHEA\_NAME, [71](#)  
ROSALIND\_NAME, [72](#)  
SATURN\_NAME, [72](#)  
SOLAR\_SYSTEM\_BARYCENTER\_NAME, [72](#)  
SSB\_MU, [72](#)  
SSB\_NAIF\_ID, [72](#)  
STAR\_EQUATORIAL\_RADIUS, [72](#)  
STAR\_FLATTENING, [72](#)  
STAR\_MU, [73](#)  
STAR\_NAIF\_IDS, [73](#)  
STAR\_ORIENTATION\_PARAMETERS, [73](#)  
STAR\_PHOTOSPHERE\_RADIUS, [73](#)  
STAR\_RADIANT\_POWER, [73](#)  
STAR\_REFERENCE\_DISTANCE, [73](#)  
STAR\_SPICE\_FRAME\_ID, [73](#)  
STAR\_TWO\_BODY\_ELEMENTS, [74](#)  
STAR\_TWO\_BODY\_EPOCH, [74](#)  
SUN\_NAME, [74](#)  
TELESTO\_NAME, [74](#)  
TETHYS\_NAME, [74](#)  
THALASSA\_NAME, [74](#)  
THEBE\_NAME, [74](#)  
TITAN\_NAME, [75](#)  
TITANIA\_NAME, [75](#)  
TRITON\_NAME, [75](#)  
UMBRIEL\_NAME, [75](#)  
URANUS\_NAME, [75](#)  
VENUS\_NAME, [75](#)  
GmatStringUtil, [75](#)  
    AddEnclosingString, [79](#)  
    AlignmentType, [79](#)  
    AreAllBracketsBalanced, [80](#)  
    AreAllNamesValid, [80](#)  
    BuildNumber, [80](#)  
    Capitalize, [80](#)  
    DecomposeBy, [81](#)  
    EndsWith, [81](#)  
    EndsWithPathSeparator, [81](#)  
    FindFirstAndLast, [81](#)  
    FindLastParenMatch, [81](#)  
    FindMatchingBracket, [82](#)  
    FindMatchingParen, [82](#)  
    FindParenMatch, [82](#)  
    GetAlignmentString, [82](#)  
    GetArrayCommaIndex, [82](#)  
    GetArrayIndex, [83](#)  
    GetArrayIndexVar, [83](#)  
    GetArrayName, [83](#)  
    GetClosingBracket, [84](#)

- GetInvalidNameMessageFormat, 84
- GetVarNames, 84
- HasMissingQuote, 84
- HasNoBrackets, 84
- IsBlank, 84
- IsBracketBalanced, 85
- IsBracketPartOfArray, 85
- IsCommaPartOfArray, 85
- IsEnclosedWith, 85
- IsEnclosedWithBraces, 85
- IsEnclosedWithBrackets, 85
- IsEnclosedWithExtraParen, 86
- IsLastNumberPartOfName, 86
- IsMathEquation, 86
- IsMathOperator, 86
- IsNumber, 86
- IsOneElementArray, 86
- IsOuterParen, 87
- IsParenBalanced, 87
- IsParenEmpty, 87
- IsParenPartOfArray, 87
- IsSimpleArrayElement, 87
- IsSingleItem, 87
- IsStringInsideSymbols, 87
- IsThereEqualSign, 88
- IsThereMathSymbol, 88
- IsValidExtendedIdentity, 88
- IsValidFileName, 88
- IsValidFullFileName, 88
- IsValidFunctionCall, 89
- IsValidIdentity, 89
- IsValidName, 89
- IsValidNumber, 89
- IsValidParameterName, 89
- IsValidReal, 90
- MakeCommentLines, 90
- NumberOfOccurrences, 90
- NumberOfScientificNotation, 90
- PadWithBlanks, 90
- ParseFunctionCall, 90
- ParseFunctionName, 91
- ParseName, 91
- ParseParameter, 91
- RealToString, 91, 92
- RemoveAll, 92
- RemoveAllBlanks, 92
- RemoveEnclosingString, 92
- RemoveExtraParen, 93
- RemoveInlineComment, 93
- RemoveLastNumber, 93
- RemoveLastString, 93
- RemoveMathSymbols, 93
- RemoveMultipleSpaces, 93
- RemoveOuterParen, 94
- RemoveOuterString, 94
- RemoveScientificNotation, 94
- RemoveSpaceInBrackets, 94
- RemoveTrailingZeros, 94
- Replace, 95
- ReplaceChainedUnaryOperators, 95
- ReplaceFirst, 95
- ReplaceName, 95
- ReplaceNumber, 95
- SeparateBrackets, 95
- SeparateBy, 96
- SeparateByComma, 96
- SeparateDots, 96
- StartsWith, 96
- StringToWideString, 96
- Strip, 96
- StripType, 79
- ToBoolean, 97
- ToBooleanArray, 97
- ToInteger, 97
- ToIntegerArray, 98
- ToLower, 98
- ToOnOff, 98
- ToOrdinal, 98
- ToReal, 98, 99
- ToRealArray, 99
- ToString, 99, 100
- ToStringArray, 100
- ToStringNoZeros, 100
- ToUnsignedInt, 101
- ToUnsignedIntArray, 101
- ToUpper, 101
- Trim, 101
- WideStringToString, 101, 102
- WriteStringArray, 102
- GmatTime, 231
  - ~GmatTime, 232
  - AddSeconds, 233
  - Clone, 233
  - Days, 239
  - FracSec, 239
  - GetDays, 233
  - GetFracSec, 233
  - GetMjd, 233
  - GetSec, 234
  - GetTimeInSec, 234
  - GmatTime, 232, 233
  - IsNearlyEqual, 234
  - operator!=, 234
  - operator<, 236
  - operator<=, 236
  - operator>, 237
  - operator>=, 237
  - operator\*, 234
  - operator+, 234, 235
  - operator+=, 235
  - operator-, 235
  - operator-=, 235
  - operator/, 236
  - operator=, 236
  - operator==, 237
  - Sec, 239

- SetDays, [237](#)
- SetFracSec, [238](#)
- SetMjdString, [238](#)
- SetSec, [238](#)
- SetTimeInSec, [238](#)
- SubtractSeconds, [238](#)
- ToString, [238](#)
- GmatTimeConstants, [102](#)
  - A1\_TAI\_OFFSET, [103](#)
  - A1MJD\_OF\_J2000, [103](#)
  - DAYS\_BEFORE\_MONTH, [104](#)
  - DAYS\_IN\_MONTH, [104](#)
  - DAYS\_PER\_JULIAN\_CENTURY, [104](#)
  - DAYS\_PER\_SEC, [104](#)
  - DAYS\_PER\_YEAR, [104](#)
  - DayName, [103](#)
  - JD\_JAN\_5\_1941, [104](#)
  - JD\_MJD\_OFFSET, [105](#)
  - JD\_NOV\_17\_1858, [105](#)
  - JD\_OF\_J2000, [105](#)
  - JULIAN\_DATE\_OF\_010541, [105](#)
  - LEAP\_YEAR\_DAYS\_BEFORE\_MONTH, [105](#)
  - LEAP\_YEAR\_DAYS\_IN\_MONTH, [105](#)
  - MJD\_EPOCH\_PRECISION, [106](#)
  - MJD\_OF\_J2000, [106](#)
  - MonthName, [103](#)
  - SECS\_PER\_DAY, [106](#)
  - SECS\_PER\_HOUR, [106](#)
  - SECS\_PER\_MINUTE, [106](#)
  - TIME\_OF\_J2000, [106](#)
  - TT\_TAI\_OFFSET, [106](#)
- GmatTimeUtil, [107](#)
  - FormatCurrentTime, [107](#)
  - GetGregorianFormat, [107](#)
  - GetMonth, [107](#)
  - GetMonthName, [107](#)
  - IsValidMonthName, [107](#)
- GmatTimeUtil::CalDate, [141](#)
  - CalDate, [142](#)
  - day, [142](#)
  - hour, [142](#)
  - minute, [143](#)
  - month, [143](#)
  - second, [143](#)
  - year, [143](#)
- GmatTimeUtil::ElapsedDate, [175](#)
  - days, [176](#)
  - ElapsedDate, [175](#), [176](#)
  - hours, [176](#)
  - minutes, [176](#)
  - seconds, [176](#)
- gmatdefs.hpp
  - BooleanArray, [448](#)
  - Byte, [449](#)
  - ColorMap, [449](#)
  - DEFAULT\_TO\_NO\_CLONES, [448](#)
  - DEFAULT\_TO\_NO\_REFOBJECTS, [448](#)
  - EpochArray, [449](#)
  - GEOPARMS, [449](#)
  - GMAT\_API, [448](#)
  - GmatEpoch, [449](#)
  - Integer, [449](#)
  - IntegerArray, [449](#)
  - IntegerMap, [450](#)
  - ObjectArray, [450](#)
  - ObjectMap, [450](#)
  - ObjectMapStack, [450](#)
  - Radians, [450](#)
  - Real, [450](#)
  - RealArray, [450](#)
  - StateArray, [450](#)
  - StringArray, [451](#)
  - UnsignedInt, [451](#)
  - UnsignedIntArray, [451](#)
  - WrapperArray, [451](#)
  - WrapperMap, [451](#)
- GravityFileException, [239](#)
  - GravityFileException, [240](#)
- GregorianCalendar, [240](#)
  - ~GregorianCalendar, [242](#)
  - GetDate, [242](#)
  - GetType, [242](#)
  - GetYMDHMS, [242](#)
  - GregorianCalendar, [241](#)
  - IsValid, [242](#), [243](#)
  - SetDate, [243](#)
  - SetType, [244](#)
- GregorianCalendar::GregorianCalendarException, [244](#)
  - GregorianCalendarException, [245](#)
- GregorianCalendarException
  - GregorianCalendar::GregorianCalendarException, [245](#)
- GuiMode
  - GmatGlobal, [215](#)
- HELENE\_NAME
  - GmatSolarSystemDefaults, [62](#)
- handler
  - Gmat::PluginResource, [296](#)
- HasMissingQuote
  - GmatStringUtil, [84](#)
- HasNoBrackets
  - GmatStringUtil, [84](#)
- HasNoPath
  - GmatFileUtil, [33](#)
- hour
  - GmatTimeUtil::CalDate, [142](#)
- HourOfDay
  - TimeTypes.hpp, [547](#)
- hours
  - GmatTimeUtil::ElapsedDate, [176](#)
- HyperbolicToTrueAnomaly
  - StateConversionUtil, [394](#)
- IAPETUS\_NAME
  - GmatSolarSystemDefaults, [63](#)
- INTEGER\_MAX
  - GmatRealConstants, [51](#)



- INTEGER\_UNDEFINED
  - GmatIntegerConstants, [37](#)
  - GmatRealConstants, [51](#)
- INTEGER\_WIDTH
  - GmatGlobal, [231](#)
- IO\_NAME
  - GmatSolarSystemDefaults, [63](#)
- Identity
  - Rmatrix, [313](#)
- IllegalSize
  - ArrayTemplateExceptions::IllegalSize, [247](#)
  - TableTemplateExceptions::IllegalSize, [246](#)
- IllegalTime
  - RealUtilitiesExceptions::IllegalTime, [249](#)
- IncomingAsymptoteToCartesian
  - StateConversionUtil, [394](#)
- independent
  - Interpolator, [256](#)
- index
  - Element, [180](#)
- init
  - ArrayTemplate, [126](#)
  - TableTemplate, [409](#)
- Initialize
  - EopFile, [184](#)
  - LeapSecsFileReader, [273](#)
- Instance
  - ConsoleMessageReceiver, [151](#)
  - FileManager, [208](#)
  - GmatGlobal, [220](#)
  - RandomNumber, [306](#)
  - TimeSystemConverter, [425](#)
- instanceName
  - Interpolator, [257](#)
- Integer
  - gmatdefs.hpp, [449](#)
  - utildefs.hpp, [454](#)
- IntegerArray
  - gmatdefs.hpp, [449](#)
  - utildefs.hpp, [454](#)
- IntegerMap
  - gmatdefs.hpp, [450](#)
  - utildefs.hpp, [454](#)
- Interpolate
  - Interpolator, [254](#)
  - LagrangeInterpolator, [267](#)
- Interpolator, [249](#)
  - ~Interpolator, [251](#)
  - AddPoint, [251](#)
  - AllocateArrays, [252](#)
  - allowExtrapolation, [255](#)
  - bufferSize, [256](#)
  - CleanupArrays, [252](#)
  - Clear, [252](#)
  - Clone, [252](#)
  - CopyArrays, [252](#)
  - dataIncreases, [256](#)
  - dependent, [256](#)
  - dimension, [256](#)
  - forceInterpolation, [256](#)
  - GetBufferSize, [253](#)
  - GetForceInterpolation, [253](#)
  - GetName, [253](#)
  - GetPointCount, [253](#)
  - GetRange, [253](#)
  - independent, [256](#)
  - instanceName, [257](#)
  - Interpolate, [254](#)
  - Interpolator, [250](#), [251](#)
  - IsInterpolationFeasible, [254](#)
  - latestPoint, [257](#)
  - operator=, [255](#)
  - pointCount, [257](#)
  - previousX, [257](#)
  - range, [257](#)
  - rangeCalculated, [257](#)
  - requiredPoints, [258](#)
  - SetExtrapolation, [255](#)
  - SetForceInterpolation, [255](#)
  - SetRange, [255](#)
- InterpolatorException, [258](#)
  - ~InterpolatorException, [259](#)
  - InterpolatorException, [259](#)
- InvalidStateRepresentationException, [259](#)
  - InvalidStateRepresentationException, [260](#)
- InvalidTimeException, [261](#)
  - InvalidTimeException, [261](#)
- Inverse
  - Rmatrix, [313](#)
  - Rmatrix33, [324](#)
  - Rmatrix66, [332](#)
- Invert
  - CholeskyFactorization, [145](#), [146](#)
  - LUFactorization, [278](#)
  - MatrixFactorization, [281](#)
  - QRFactorization, [301](#)
  - SchurFactorization, [366](#), [367](#)
- IsApplInstalled
  - GmatFileUtil, [33](#)
- IsAsciiFile
  - GmatFileUtil, [33](#)
- IsBatchMode
  - GmatGlobal, [220](#)
- IsBinaryIn
  - GmatGlobal, [220](#)
- IsBinaryOut
  - GmatGlobal, [221](#)
- IsBlank
  - EopFile, [184](#)
  - GmatStringUtil, [84](#)
- IsBracketBalanced
  - GmatStringUtil, [85](#)
- IsBracketPartOfArray
  - GmatStringUtil, [85](#)
- IsCommaPartOfArray
  - GmatStringUtil, [85](#)



- IsDataNearCenter
  - LagrangeInterpolator, 268
- IsEnclosedWith
  - GmatStringUtil, 85
- IsEnclosedWithBraces
  - GmatStringUtil, 85
- IsEnclosedWithBrackets
  - GmatStringUtil, 85
- IsEnclosedWithExtraParen
  - GmatStringUtil, 86
- IsEqual
  - GmatMathUtil, 43
- IsEven
  - GmatMathUtil, 43
- IsEventLocationAvailable
  - GmatGlobal, 221
- IsFatal
  - BaseException, 140
- IsUISavable
  - GmatGlobal, 221
- IsGmatCompiledIn64Bit
  - GmatGlobal, 221
- IsHiddenCommand
  - GmatGlobal, 221
- IsHorizontal
  - GmatGlobal, 221
- IsInLeapSecond
  - LeapSecsFileReader, 274
  - TimeSystemConverter, 425
- IsInf
  - GmatMathUtil, 43
- isInitialized
  - EopFile, 186
- IsInterpolationFeasible
  - Interpolator, 254
  - LagrangeInterpolator, 268
- IsLastNumberPartOfName
  - GmatStringUtil, 86
- IsLeapYear
  - DateUtil, 165
  - DateUtil.cpp, 469
  - DateUtil.hpp, 474
- IsMathEquation
  - GmatStringUtil, 86
- IsMathOperator
  - GmatStringUtil, 86
- IsMatlabAvailable
  - GmatGlobal, 222
- IsMatlabDebugEnabled
  - GmatGlobal, 222
- IsMissionTreeDebugEnabled
  - GmatGlobal, 222
- IsNaN
  - GmatMathUtil, 43
- IsNearlyEqual
  - GmatTime, 234
- IsNegative
  - GmatMathUtil, 44
- IsNitsClient
  - GmatGlobal, 222
- IsNonNegative
  - GmatMathUtil, 44
- IsNumber
  - GmatStringUtil, 86
- IsOdd
  - GmatMathUtil, 44
- IsOneElementArray
  - GmatStringUtil, 86
- IsOrthogonal
  - Rmatrix, 313
  - Rmatrix33, 324
  - Rmatrix66, 332
- IsOrthonormal
  - Rmatrix, 314
  - Rmatrix33, 324
  - Rmatrix66, 332
- IsOsWindows
  - GmatFileUtil, 33
- IsOuterParen
  - GmatStringUtil, 87
- IsParenBalanced
  - GmatStringUtil, 87
- IsParenEmpty
  - GmatStringUtil, 87
- IsParenPartOfArray
  - GmatStringUtil, 87
- IsPathAbsolute
  - GmatFileUtil, 34
- IsPathRelative
  - GmatFileUtil, 34
- IsPositive
  - GmatMathUtil, 44
- IsRvValid
  - StateConversionUtil, 394
- IsSameFileName
  - GmatFileUtil, 34, 35
- IsScientific
  - GmatGlobal, 222
- IsSimpleArrayElement
  - GmatStringUtil, 87
- IsSingleItem
  - GmatStringUtil, 87
- IsSingular
  - Rmatrix::IsSingular, 263
- IsSized
  - ArrayTemplate, 127
  - TableTemplate, 409
- isSizedD
  - ArrayTemplate, 129
  - TableTemplate, 411
- IsStringInsideSymbols
  - GmatStringUtil, 87
- IsThereEqualSign
  - GmatStringUtil, 88
- IsThereMathSymbol
  - GmatStringUtil, 88

- IsValid
  - Date, [159](#)
  - GregorianCalendar, [242](#), [243](#)
  - Rvector6, [360](#)
- IsValidAnomalyType
  - StateConversionUtil, [395](#)
- IsValidEulerSequence
  - AttitudeConversionUtility, [131](#)
- IsValidExtendedIdentity
  - GmatStringUtil, [88](#)
- IsValidFileName
  - GmatFileUtil, [35](#)
  - GmatStringUtil, [88](#)
- IsValidFullFileName
  - GmatStringUtil, [88](#)
- IsValidFunctionCall
  - GmatStringUtil, [89](#)
- IsValidGregorian
  - DateUtil, [164](#)
- IsValidIdentity
  - GmatStringUtil, [89](#)
- IsValidLogFile
  - MessageReceiver, [290](#)
- IsValidMonthName
  - GmatTimeUtil, [107](#)
- IsValidName
  - GmatStringUtil, [89](#)
- IsValidNumber
  - GmatStringUtil, [89](#)
- IsValidParameterName
  - GmatStringUtil, [89](#)
- IsValidReal
  - GmatStringUtil, [90](#)
- IsValidStateRepresentation
  - BodyFixedStateConverterUtil, [17](#)
- IsValidTime
  - DateUtil, [165](#)
  - DateUtil.cpp, [469](#)
  - DateUtil.hpp, [474](#)
- IsValidTimeSystem
  - TimeSystemConverter, [425](#)
- IsWritingFilePathInfo
  - GmatGlobal, [222](#)
- IsWritingGmatKeyword
  - GmatGlobal, [222](#)
- IsWritingParameterInfo
  - GmatGlobal, [222](#)
- IsZero
  - GmatMathUtil, [44](#)
- IsZeroVector
  - Rvector, [341](#)
- J2000
  - A1Mjd, [121](#)
- JANUS\_NAME
  - GmatSolarSystemDefaults, [63](#)
- JD\_JAN\_5\_1941
  - GmatTimeConstants, [104](#)
- JD\_MJD\_OFFSET
  - GmatTimeConstants, [105](#)
- JD\_NOV\_17\_1858
  - GmatTimeConstants, [105](#)
- JD\_OF\_J2000
  - GmatTimeConstants, [105](#)
- JULIAN\_DATE\_OF\_010541
  - GmatTimeConstants, [105](#)
- JULIET\_NAME
  - GmatSolarSystemDefaults, [63](#)
- JUPITER\_NAME
  - GmatSolarSystemDefaults, [63](#)
- JulianDate
  - DateUtil, [165](#)
  - DateUtil.cpp, [469](#)
  - DateUtil.hpp, [475](#)
- julianDate
  - LeapSecondInformation, [271](#)
- JulianDay
  - DateUtil, [165](#)
- KeplerianToCartesian
  - StateConversionUtil, [395](#), [396](#)
- KeplerianToDelaunay
  - StateConversionUtil, [396](#)
- KeplerianToModKeplerian
  - StateConversionUtil, [396](#)
- L\_B
  - TimeSystemConverter, [427](#)
- LARISSA\_NAME
  - GmatSolarSystemDefaults, [63](#)
- LATEST\_VALID\_GREGORIAN
  - DateUtil, [170](#)
- LATEST\_VALID\_MJD\_VALUE
  - DateUtil, [170](#)
- LATEST\_VALID\_MJD
  - DateUtil, [170](#)
- LEAP\_YEAR\_DAYS\_BEFORE\_MONTH
  - GmatTimeConstants, [105](#)
- LEAP\_YEAR\_DAYS\_IN\_MONTH
  - GmatTimeConstants, [105](#)
- LUFactorization, [276](#)
  - ~LUFactorization, [277](#)
  - Determinant, [277](#)
  - Factor, [278](#)
  - Invert, [278](#)
  - LUFactorization, [277](#)
  - operator=, [278](#)
  - SolveSystem, [279](#)
- LagrangeInterpolator, [263](#)
  - ~LagrangeInterpolator, [265](#)
  - actualSize, [269](#)
  - AddPoint, [265](#)
  - AllocateArrays, [266](#)
  - beginIndex, [269](#)
  - BuildDataPoints, [266](#)
  - CleanupArrays, [266](#)
  - Clear, [266](#)
  - Clone, [266](#)

- CopyArrays, [267](#)
- dataIndex, [269](#)
- endIndex, [269](#)
- FindStartingPoint, [267](#)
- Interpolate, [267](#)
- IsDataNearCenter, [268](#)
- IsInterpolationFeasible, [268](#)
- LagrangeInterpolator, [265](#)
- lastX, [270](#)
- MAX\_BUFFER\_SIZE, [270](#)
- operator=, [268](#)
- order, [270](#)
- startPoint, [270](#)
- UpdateBeginAndEndIndex, [269](#)
- x, [270](#)
- y, [270](#)
- lastId
  - Gmat::PluginResource, [296](#)
- lastIndex
  - EopFile, [186](#)
- lastOffset
  - EopFile, [186](#)
- lastTaiMjd
  - EopFile, [186](#)
- lastUtcJd
  - EopFile, [186](#)
- lastX
  - LagrangeInterpolator, [270](#)
- latestPoint
  - Interpolator, [257](#)
- LeapSecondInformation, [271](#)
  - julianDate, [271](#)
  - offset1, [271](#)
  - offset2, [271](#)
  - offset3, [272](#)
  - taiMJD, [272](#)
- LeapSecsFileReader, [272](#)
  - ~LeapSecsFileReader, [273](#)
  - GetFirstLeapSecondMJD, [273](#)
  - Initialize, [273](#)
  - IsInLeapSecond, [274](#)
  - LeapSecsFileReader, [272](#), [273](#)
  - NumberOfLeapSecondsFrom, [274](#)
  - operator=, [274](#)
- LeapYearError
  - Date::LeapYearError, [275](#)
- Ln
  - GmatMathUtil, [44](#)
- Log
  - GmatMathUtil, [45](#)
- Log10
  - GmatMathUtil, [45](#)
- LogMessage
  - ConsoleMessageReceiver, [151](#)
  - MessageInterface, [284](#)
  - MessageReceiver, [290](#)
- LogfileSource
  - GmatGlobal, [215](#)
- LowerLeft
  - Rmatrix66, [333](#)
- LowerRight
  - Rmatrix66, [333](#)
- M\_E\_COEFF1
  - TimeSystemConverter, [427](#)
- M\_E\_OFFSET
  - TimeSystemConverter, [427](#)
- MARS\_NAME
  - GmatSolarSystemDefaults, [63](#)
- MAX\_BUFFER\_SIZE
  - LagrangeInterpolator, [270](#)
- MAX\_DAY
  - DateUtil, [170](#)
- MAX\_HOUR
  - DateUtil, [170](#)
- MAX\_MESSAGE\_LENGTH
  - BaseException, [141](#)
  - MessageInterface, [287](#)
- MAX\_MINUTE
  - DateUtil, [170](#)
- MAX\_MONTH
  - DateUtil, [170](#)
- MAX\_SEC
  - DateUtil, [170](#)
- MAX\_TABLE\_SIZE
  - EopFile, [186](#)
- MAX\_YEAR
  - DateUtil, [171](#)
- MERCURY\_NAME
  - GmatSolarSystemDefaults, [63](#)
- METIS\_NAME
  - GmatSolarSystemDefaults, [64](#)
- MIMAS\_NAME
  - GmatSolarSystemDefaults, [64](#)
- MIN\_DAY
  - DateUtil, [171](#)
- MIN\_HOUR
  - DateUtil, [171](#)
- MIN\_MINUTE
  - DateUtil, [171](#)
- MIN\_MONTH
  - DateUtil, [171](#)
- MIN\_SEC
  - DateUtil, [171](#)
- MIN\_YEAR
  - DateUtil, [171](#)
- MIRANDA\_NAME
  - GmatSolarSystemDefaults, [64](#)
- MJD\_EPOCH\_PRECISION
  - GmatTimeConstants, [106](#)
- MJD\_OF\_J2000
  - GmatTimeConstants, [106](#)
- MOON\_CENTRAL\_BODIES
  - GmatSolarSystemDefaults, [64](#)
- MOON\_EQUATORIAL\_RADIUS
  - GmatSolarSystemDefaults, [64](#)
- MOON\_FLATTENING

- GmatSolarSystemDefaults, [64](#)
- MOON\_MU
  - GmatSolarSystemDefaults, [65](#)
- MOON\_NAIF\_IDS
  - GmatSolarSystemDefaults, [65](#)
- MOON\_NAMES
  - GmatSolarSystemDefaults, [65](#)
- MOON\_NAME
  - GmatSolarSystemDefaults, [65](#)
- MOON\_ORIENTATION\_PARAMETERS
  - GmatSolarSystemDefaults, [66](#)
- MOON\_SPICE\_FRAME\_ID
  - GmatSolarSystemDefaults, [66](#)
- MOON\_TWO\_BODY\_ELEMENTS
  - GmatSolarSystemDefaults, [66](#)
- MOON\_TWO\_BODY\_EPOCH
  - GmatSolarSystemDefaults, [66](#)
- mPackedString
  - Date, [162](#)
- MakeCommentLines
  - GmatStringUtil, [90](#)
- MakeOneColumnMatrix
  - Rmatrix, [314](#)
- MakeOneRowMatrix
  - Rmatrix, [314](#)
- MakeZeroVector
  - Rvector, [341](#)
- MatlabMode
  - GmatGlobal, [215](#)
- MatrixFactorization, [279](#)
  - ~MatrixFactorization, [280](#)
  - CompressNormalMatrix, [280](#)
  - ExpandNormalMatrixInverse, [281](#)
  - Factor, [281](#)
  - Invert, [281](#)
  - MatrixFactorization, [280](#)
  - operator=, [282](#)
  - PackedArrayIndex, [282](#)
- MatrixTimesTranspose
  - Rmatrix, [319](#)
  - Rmatrix.cpp, [519](#)
  - Rmatrix33, [328](#)
  - Rmatrix33.cpp, [522](#)
  - Rmatrix66, [336](#)
  - Rmatrix66.cpp, [524](#)
- Max
  - GmatMathUtil, [46](#)
  - GmatRealUtil, [54](#)
  - Rvector, [341](#)
- MeanToTrueAnomaly
  - StateConversionUtil, [397](#)
- MessageInterface, [282](#)
  - ClearMessage, [283](#)
  - ClearMessageQueue, [283](#)
  - GetLogEnable, [283](#)
  - GetLogFileName, [284](#)
  - GetMessageReceiver, [284](#)
  - GetQueuedMessage, [284](#)
  - LogMessage, [284](#)
  - MAX\_MESSAGE\_LENGTH, [287](#)
  - PopupMessage, [285](#)
  - PutMessage, [285](#), [286](#)
  - SetLogEnable, [286](#)
  - SetLogFile, [286](#)
  - SetLogPath, [286](#)
  - SetMessageReceiver, [286](#)
  - ShowMessage, [287](#)
- MessageReceiver, [288](#)
  - ~MessageReceiver, [289](#)
  - ClearMessage, [289](#)
  - ClearMessageQueue, [289](#)
  - GetLogEnable, [289](#)
  - GetLogFileName, [289](#)
  - GetLogFileText, [290](#)
  - GetMessage, [290](#)
  - IsValidLogFile, [290](#)
  - LogMessage, [290](#)
  - MessageReceiver, [289](#)
  - PopupMessage, [290](#), [291](#)
  - PutMessage, [291](#)
  - SetLogEnable, [291](#)
  - SetLogFile, [291](#)
  - SetLogPath, [291](#)
  - ShowMessage, [292](#)
- MessageType
  - Gmat, [21](#)
- Min
  - GmatMathUtil, [46](#)
  - GmatRealUtil, [54](#)
  - Rvector, [341](#)
- minute
  - GmatTimeUtil::CalDate, [143](#)
- MinuteOfHour
  - TimeTypes.hpp, [548](#)
- minutes
  - GmatTimeUtil::ElapsedDate, [176](#)
- Mod
  - GmatMathUtil, [46](#)
- ModEquinoctialToCartesian
  - StateConversionUtil, [397](#)
- ModKeplerianToKeplerian
  - StateConversionUtil, [397](#)
- ModifiedJulianDate
  - DateUtil, [166](#)
  - DateUtil.cpp, [470](#)
  - DateUtil.hpp, [475](#)
- ModifiedJulianDateGT
  - DateUtil, [166](#)
  - DateUtil.cpp, [470](#)
  - DateUtil.hpp, [476](#)
- month
  - GmatTimeUtil::CalDate, [143](#)
- MonthName
  - GmatTimeConstants, [103](#)
- MonthOfYear
  - TimeTypes.hpp, [548](#)

- monthD
  - Date, [162](#)
- NAIAD\_NAME
  - GmatSolarSystemDefaults, [67](#)
- NEPTUNE\_NAME
  - GmatSolarSystemDefaults, [67](#)
- NUM\_DATA\_INIT
  - Rvector6.hpp, [533](#)
- NUM\_DATA
  - Date, [162](#)
- NUM\_SECS
  - TimeSystemConverter, [427](#)
- NearestInt
  - GmatMathUtil, [46](#)
- nodeName
  - Gmat::PluginResource, [297](#)
- Norm
  - Rvector, [342](#)
- Normalize
  - Rvector, [342](#)
  - Rvector3, [351](#)
- NotSquare
  - Rmatrix::NotSquare, [293](#)
- NumberOfLeapSecondsFrom
  - LeapSecsFileReader, [274](#)
  - TimeSystemConverter, [426](#)
- NumberOfOccurrences
  - GmatStringUtil, [90](#)
- NumberOfScientificNotation
  - GmatStringUtil, [90](#)
- OBERON\_NAME
  - GmatSolarSystemDefaults, [67](#)
- OPHELIA\_NAME
  - GmatSolarSystemDefaults, [67](#)
- ObjectArray
  - gmatdefs.hpp, [450](#)
  - utildefs.hpp, [455](#)
- ObjectMap
  - gmatdefs.hpp, [450](#)
  - utildefs.hpp, [455](#)
- ObjectMapStack
  - gmatdefs.hpp, [450](#)
  - utildefs.hpp, [455](#)
- ObjectType
  - Gmat, [22](#)
- ObjectTypeArray
  - utildefs.hpp, [455](#)
- ObjectTypeArrayMap
  - utildefs.hpp, [455](#)
- ObjectTypeMap
  - utildefs.hpp, [455](#)
- offset1
  - LeapSecondInformation, [271](#)
- offset2
  - LeapSecondInformation, [271](#)
- offset3
  - LeapSecondInformation, [272](#)
- OpenLogFile
  - ConsoleMessageReceiver, [152](#)
- operator!=
  - A1Mjd, [116](#)
  - ArrayTemplate, [127](#)
  - ElapsedTime, [178](#)
  - GmatTime, [234](#)
  - Rmatrix, [314](#)
  - Rmatrix33, [325](#)
  - Rmatrix66, [333](#)
  - Rvector, [342](#)
  - Rvector3, [352](#)
  - Rvector6, [360](#)
  - TableTemplate, [410](#)
- operator<
  - A1Date, [112](#)
  - A1Mjd, [118](#)
  - Date, [159](#)
  - ElapsedTime, [179](#)
  - GmatTime, [236](#)
  - Rvector.cpp, [527](#)
  - Rvector.hpp, [529](#)
- operator<<
  - GmatRealUtil, [54, 55](#)
  - Rmatrix, [319](#)
  - Rmatrix.cpp, [520](#)
  - Rvector, [347](#)
  - Rvector.cpp, [527](#)
  - Rvector3.cpp, [530](#)
- operator<=
  - A1Mjd, [118](#)
  - ElapsedTime, [179](#)
  - GmatTime, [236](#)
- operator>
  - A1Date, [112](#)
  - A1Mjd, [119](#)
  - Date, [160](#)
  - ElapsedTime, [179](#)
  - GmatTime, [237](#)
- operator>>
  - GmatRealUtil, [55](#)
  - Rmatrix, [320](#)
  - Rmatrix.cpp, [520](#)
  - Rvector, [347](#)
  - Rvector.cpp, [528](#)
  - Rvector3.cpp, [530](#)
- operator>=
  - A1Mjd, [119](#)
  - ElapsedTime, [179](#)
  - GmatTime, [237](#)
- operator\*
  - GmatTime, [234](#)
  - Rmatrix, [314, 315, 319](#)
  - Rmatrix.cpp, [519](#)
  - Rmatrix33, [325, 328](#)
  - Rmatrix33.cpp, [522](#)
  - Rmatrix66, [333, 336](#)
  - Rmatrix66.cpp, [524](#)

- Rvector, [342](#), [346](#)
- Rvector.cpp, [527](#)
- Rvector3, [352](#), [354](#)
- Rvector3.cpp, [530](#)
- Rvector6, [360](#)
- operator\*=
  - Rmatrix, [315](#)
  - Rmatrix33, [325](#)
  - Rmatrix66, [333](#), [334](#)
  - Rvector, [343](#)
  - Rvector3, [352](#)
  - Rvector6, [360](#), [361](#)
- operator()
  - ArrayTemplate, [127](#)
  - TableTemplate, [410](#)
- operator+
  - A1Mjd, [116](#)
  - ElapsedTime, [178](#)
  - GmatTime, [234](#), [235](#)
  - Rmatrix, [315](#), [319](#)
  - Rmatrix.cpp, [519](#)
  - Rmatrix33, [326](#)
  - Rmatrix66, [334](#)
  - Rvector, [343](#)
  - Rvector3, [352](#)
  - Rvector6, [361](#)
- operator+=
  - A1Mjd, [117](#)
  - ElapsedTime, [178](#)
  - GmatTime, [235](#)
  - Rmatrix, [315](#)
  - Rmatrix33, [326](#)
  - Rmatrix66, [334](#)
  - Rvector, [343](#)
  - Rvector3, [353](#)
  - Rvector6, [361](#)
- operator-
  - A1Mjd, [117](#)
  - ElapsedTime, [178](#)
  - GmatTime, [235](#)
  - Rmatrix, [316](#), [319](#)
  - Rmatrix.cpp, [519](#)
  - Rmatrix33, [326](#)
  - Rmatrix66, [334](#)
  - Rvector, [343](#)
  - Rvector3, [353](#)
  - Rvector6, [361](#)
- operator-=
  - A1Mjd, [117](#)
  - ElapsedTime, [179](#)
  - GmatTime, [235](#)
  - Rmatrix, [316](#)
  - Rmatrix33, [326](#)
  - Rmatrix66, [334](#)
  - Rvector, [344](#)
  - Rvector3, [353](#)
  - Rvector6, [361](#)
- operator/
  - GmatTime, [236](#)
  - Rmatrix, [316](#), [319](#)
  - Rmatrix.cpp, [519](#)
  - Rmatrix33, [326](#)
  - Rmatrix66, [334](#), [335](#)
  - Rvector, [344](#)
  - Rvector3, [353](#)
  - Rvector6, [361](#), [362](#)
- operator/=
  - Rmatrix, [317](#)
  - Rmatrix33, [327](#)
  - Rmatrix66, [335](#)
  - Rvector, [344](#)
  - Rvector3, [353](#), [354](#)
  - Rvector6, [362](#)
- operator=
  - A1Date, [112](#)
  - A1Mjd, [118](#)
  - ArrayTemplate, [127](#)
  - BaseException, [140](#)
  - CholeskyFactorization, [146](#)
  - ElapsedTime, [179](#)
  - EopFile, [185](#)
  - ExponentialAtmosphere, [191](#)
  - GmatTime, [236](#)
  - Interpolator, [255](#)
  - LUFactorization, [278](#)
  - LagrangeInterpolator, [268](#)
  - LeapSecsFileReader, [274](#)
  - MatrixFactorization, [282](#)
  - QRFactorization, [301](#)
  - Rmatrix, [317](#)
  - Rmatrix33, [327](#)
  - Rmatrix66, [335](#)
  - Rvector, [344](#)
  - Rvector3, [354](#)
  - Rvector6, [362](#)
  - SchurFactorization, [367](#)
  - TableTemplate, [410](#)
  - UtcDate, [436](#)
- operator==
  - A1Mjd, [118](#)
  - ArrayTemplate, [127](#)
  - ElapsedTime, [179](#)
  - GmatTime, [237](#)
  - Rmatrix, [317](#)
  - Rmatrix33, [327](#)
  - Rmatrix66, [335](#)
  - Rvector, [344](#)
  - Rvector3, [354](#)
  - Rvector6, [362](#)
  - TableTemplate, [410](#)
- operator[]
  - ArrayTemplate, [128](#)
- order
  - LagrangeInterpolator, [270](#)
- OutOfBounds
  - ArrayTemplateExceptions::OutOfBounds, [294](#)

- TableTemplateExceptions::OutOfBounds, 295
- Outerproduct
  - Rvector, 347
  - Rvector.cpp, 528
  - Rvector3, 355
  - Rvector3.cpp, 531
- OutgoingAsymptoteToCartesian
  - StateConversionUtil, 398
- PAN\_NAME
  - GmatSolarSystemDefaults, 67
- PANDORA\_NAME
  - GmatSolarSystemDefaults, 67
- PHOBOS\_NAME
  - GmatSolarSystemDefaults, 68
- PHOEBE\_NAME
  - GmatSolarSystemDefaults, 68
- PLANET\_EQUATORIAL\_RADIUS
  - GmatSolarSystemDefaults, 68
- PLANET\_FLATTENING
  - GmatSolarSystemDefaults, 68
- PLANET\_MU
  - GmatSolarSystemDefaults, 68
- PLANET\_NAIF\_IDS
  - GmatSolarSystemDefaults, 69
- PLANET\_NAMES
  - GmatSolarSystemDefaults, 69
- PLANET\_ORIENTATION\_PARAMETERS
  - GmatSolarSystemDefaults, 69
- PLANET\_SPICE\_FRAME\_ID
  - GmatSolarSystemDefaults, 70
- PLANET\_TWO\_BODY\_ELEMENTS
  - GmatSolarSystemDefaults, 70
- PLANET\_TWO\_BODY\_EPOCH
  - GmatSolarSystemDefaults, 70
- PLUGIN\_RESOURCE
  - Gmat, 21
- PLUTO\_NAME
  - GmatSolarSystemDefaults, 71
- PORTIA\_NAME
  - GmatSolarSystemDefaults, 71
- PROMETHEUS\_NAME
  - GmatSolarSystemDefaults, 71
- PROTEUS\_NAME
  - GmatSolarSystemDefaults, 71
- PUCK\_NAME
  - GmatSolarSystemDefaults, 71
- PackedArrayIndex
  - MatrixFactorization, 282
- PadWithBlanks
  - GmatStringUtil, 90
- ParameterType
  - Gmat, 23
- parentNodeName
  - Gmat::PluginResource, 297
- ParseFileExtension
  - GmatFileUtil, 35
- ParseFileName
  - GmatFileUtil, 35
- ParseFirstPathName
  - GmatFileUtil, 36
- ParseFunctionCall
  - GmatStringUtil, 90
- ParseFunctionName
  - GmatStringUtil, 91
- ParseName
  - GmatStringUtil, 91
- ParseParameter
  - GmatStringUtil, 91
- ParsePathName
  - GmatFileUtil, 36
- PlanetodeticToCartesian
  - StateConversionUtil, 398
- PlotMode
  - GmatGlobal, 215
- PluginResource
  - Gmat::PluginResource, 296
- pointCount
  - Interpolator, 257
- polarMotion
  - EopFile, 186
- PopupMessage
  - ConsoleMessageReceiver, 152
  - MessageInterface, 285
  - MessageReceiver, 290, 291
- Pow
  - GmatMathUtil, 46
- PrepareCompare
  - GmatFileUtil, 36
- previousIndex
  - EopFile, 187
- previousX
  - Interpolator, 257
- Pseudoinverse
  - Rmatrix, 317
- PutMessage
  - ConsoleMessageReceiver, 153
  - MessageInterface, 285, 286
  - MessageReceiver, 291
- QRFactorization, 298
  - ~QRFactorization, 299
  - AddToQR, 300
  - Determinant, 300
  - Factor, 300
  - GetParameterMatrix, 301
  - Invert, 301
  - operator=, 301
  - QRFactorization, 299
  - RemoveFromQR, 301
- QUAT\_MIN\_MAG
  - GmatAttitudeConstants, 26
- Quotient
  - GmatMathUtil, 47
- REAL\_DIG
  - GmatRealConstants, 51
- REAL\_EPSILON

- GmatRealConstants, [52](#)
- REAL\_MAX\_10\_EXP
  - GmatRealConstants, [52](#)
- REAL\_MAX\_EXP
  - GmatRealConstants, [52](#)
- REAL\_MAX
  - GmatRealConstants, [52](#)
- REAL\_MIN\_10\_EXP
  - GmatRealConstants, [52](#)
- REAL\_MIN\_EXP
  - GmatRealConstants, [52](#)
- REAL\_MIN
  - GmatRealConstants, [52](#)
- REAL\_TOL
  - GmatRealConstants, [52](#)
- REAL\_UNDEFINED\_LARGE
  - GmatRealConstants, [53](#)
- REAL\_UNDEFINED
  - GmatRealConstants, [53](#)
- RHEA\_NAME
  - GmatSolarSystemDefaults, [71](#)
- ROSALIND\_NAME
  - GmatSolarSystemDefaults, [72](#)
- RVECTOR6\_UNDEFINED
  - Rvector6, [363](#)
- RaCodecToCartesian
  - GmatRealUtil, [55](#)
- RaCodecToRaDec
  - GmatRealUtil, [56](#)
- RaDecToCartesian
  - GmatRealUtil, [56](#)
- RaDecToRaCodec
  - GmatRealUtil, [56](#)
- Rad
  - GmatMathUtil, [47](#)
- RadToDeg
  - GmatMathUtil, [47](#)
- Radians
  - gmatdefs.hpp, [450](#)
  - utildefs.hpp, [455](#)
- radiusD
  - GmatRealUtil::RaCodec, [302](#)
  - GmatRealUtil::RaDec, [303](#)
- Rand
  - GmatMathUtil, [47](#)
- Randn
  - GmatMathUtil, [48](#)
- RandomNumber, [304](#)
  - ~RandomNumber, [304](#)
  - Gaussian, [305](#)
  - GaussianArray, [305](#)
  - Instance, [306](#)
  - SetClockSeed, [306](#)
  - SetSeed, [306](#)
  - Uniform, [306](#)
  - UniformArray, [307](#)
- range
  - Interpolator, [257](#)
- rangeCalculated
  - Interpolator, [257](#)
- ReadStartupFile
  - FileManager, [208](#)
- Real
  - gmatdefs.hpp, [450](#)
  - utildefs.hpp, [456](#)
- RealArray
  - gmatdefs.hpp, [450](#)
  - utildefs.hpp, [456](#)
- RealToString
  - GmatRealUtil, [56](#)
  - GmatStringUtil, [91](#), [92](#)
- RealUtilitiesExceptions, [308](#)
- RealUtilitiesExceptions::ArgumentError, [122](#)
  - ArgumentError, [122](#)
- RealUtilitiesExceptions::IllegalTime, [248](#)
  - IllegalTime, [249](#)
- refDensity
  - ExponentialAtmosphere, [192](#)
- refHeight
  - ExponentialAtmosphere, [192](#)
- Rem
  - GmatMathUtil, [48](#)
- RemoveAll
  - GmatStringUtil, [92](#)
- RemoveAllBlanks
  - GmatStringUtil, [92](#)
- RemoveEnclosingString
  - GmatStringUtil, [92](#)
- RemoveExtraParen
  - GmatStringUtil, [93](#)
- RemoveFromQR
  - QRFactorization, [301](#)
- RemoveHiddenCommand
  - GmatGlobal, [223](#)
- RemoveInlineComment
  - GmatStringUtil, [93](#)
- RemoveLastNumber
  - GmatStringUtil, [93](#)
- RemoveLastString
  - GmatStringUtil, [93](#)
- RemoveMathSymbols
  - GmatStringUtil, [93](#)
- RemoveMultipleSpaces
  - GmatStringUtil, [93](#)
- RemoveOuterParen
  - GmatStringUtil, [94](#)
- RemoveOuterString
  - GmatStringUtil, [94](#)
- RemoveRowCol
  - SchurFactorization, [367](#)
- RemoveScientificNotation
  - GmatStringUtil, [94](#)
- RemoveSpaceInBrackets
  - GmatStringUtil, [94](#)
- RemoveTrailingZeros
  - GmatStringUtil, [94](#)



- RenameFile
  - FileManager, 208
- Replace
  - GmatStringUtil, 95
- ReplaceChainedUnaryOperators
  - GmatStringUtil, 95
- ReplaceFirst
  - GmatStringUtil, 95
- ReplaceName
  - GmatStringUtil, 95
- ReplaceNumber
  - GmatStringUtil, 95
- requiredPoints
  - Interpolator, 258
- RequiresCelestialBodyOrigin
  - StateConversionUtil, 398
- RequiresFixedCoordinateSystem
  - StateConversionUtil, 399
- ResetEopFile
  - EopFile, 185
- Resize
  - ArrayTemplate, 128
- RestoreAllRowCols
  - SchurFactorization, 368
- rightAscensionD
  - GmatRealUtil::RaCodec, 303
  - GmatRealUtil::RaDec, 303
- Rmatrix, 308
  - ~Rmatrix, 311
  - AntiSymmetric, 311
  - Cofactor, 312
  - Determinant, 312
  - Diagonal, 312
  - ElementWiseDivide, 312
  - ElementWiseMultiply, 312
  - GetColumn, 312
  - GetRow, 313
  - GetRowOrColumn, 313
  - GetStringVals, 313
  - Identity, 313
  - Inverse, 313
  - IsOrthogonal, 313
  - IsOrthonormal, 314
  - MakeOneColumnMatrix, 314
  - MakeOneRowMatrix, 314
  - MatrixTimesTranspose, 319
  - operator!=, 314
  - operator<<, 319
  - operator>>, 320
  - operator\*, 314, 315, 319
  - operator\*==, 315
  - operator+, 315, 319
  - operator+=, 315
  - operator-, 316, 319
  - operator-=, 316
  - operator/, 316, 319
  - operator/==, 317
  - operator=, 317
  - operator==, 317
  - Pseudoinverse, 317
  - Rmatrix, 311
  - Rvector, 320, 347
  - Rvector3, 320
  - SkewSymmetric4by4, 320
  - stringVals, 321
  - Symmetric, 317
  - ToRowString, 317
  - ToString, 318
  - Trace, 318
  - Transpose, 318
  - TransposeTimesMatrix, 320
  - TransposeTimesTranspose, 320
- Rmatrix.cpp
  - MatrixTimesTranspose, 519
  - operator<<, 520
  - operator>>, 520
  - operator\*, 519
  - operator+, 519
  - operator-, 519
  - operator/, 519
  - SkewSymmetric4by4, 520
  - TransposeTimesMatrix, 520
  - TransposeTimesTranspose, 520
- Rmatrix33, 321
  - ~Rmatrix33, 323
  - AntiSymmetric, 324
  - Determinant, 324
  - GetDataDescriptions, 324
  - Inverse, 324
  - IsOrthogonal, 324
  - IsOrthonormal, 324
  - MatrixTimesTranspose, 328
  - operator!=, 325
  - operator\*, 325, 328
  - operator\*==, 325
  - operator+, 326
  - operator+=, 326
  - operator-, 326
  - operator-=, 326
  - operator/, 326
  - operator/=, 327
  - operator=, 327
  - operator==, 327
  - Rmatrix33, 323
  - Rvector3, 328, 355
  - Set, 327
  - SkewSymmetric, 328
  - Symmetric, 327
  - Trace, 328
  - Transpose, 328
  - TransposeTimesMatrix, 329
  - TransposeTimesTranspose, 329
- Rmatrix33.cpp
  - MatrixTimesTranspose, 522
  - operator\*, 522
  - SkewSymmetric, 522

- TransposeTimesRmatrix, [523](#)
- TransposeTimesTranspose, [523](#)
- Rmatrix66, [329](#)
  - ~Rmatrix66, [332](#)
  - AntiSymmetric, [332](#)
  - Determinant, [332](#)
  - Inverse, [332](#)
  - IsOrthogonal, [332](#)
  - IsOrthonormal, [332](#)
  - LowerLeft, [333](#)
  - LowerRight, [333](#)
  - MatrixTimesTranspose, [336](#)
  - operator!=, [333](#)
  - operator\*, [333](#), [336](#)
  - operator\*==, [333](#), [334](#)
  - operator+, [334](#)
  - operator+=, [334](#)
  - operator-, [334](#)
  - operator-=, [334](#)
  - operator/, [334](#), [335](#)
  - operator/==, [335](#)
  - operator/, [335](#)
  - operator=, [335](#)
  - operator==, [335](#)
  - Rmatrix66, [331](#)
  - Rvector6, [337](#), [363](#)
  - Set, [335](#)
  - SetUndefined, [335](#)
  - SkewSymmetric, [337](#)
  - Symmetric, [336](#)
  - Trace, [336](#)
  - Transpose, [336](#)
  - TransposeTimesMatrix, [337](#)
  - TransposeTimesTranspose, [337](#)
  - UpperLeft, [336](#)
  - UpperRight, [336](#)
- Rmatrix66.cpp
  - MatrixTimesTranspose, [524](#)
  - operator\*, [524](#)
  - SkewSymmetric, [525](#)
  - TransposeTimesMatrix, [525](#)
  - TransposeTimesTranspose, [525](#)
- Rmatrix::DivideByZero, [174](#)
  - DivideByZero, [175](#)
- Rmatrix::IsSingular, [262](#)
  - IsSingular, [263](#)
- Rmatrix::NotSquare, [292](#)
  - NotSquare, [293](#)
- Round
  - GmatMathUtil, [48](#)
- rowsD
  - TableTemplate, [411](#)
- RunMode
  - GmatGlobal, [216](#)
- RunState
  - Gmat, [24](#)
- Rvector, [338](#)
  - ~Rvector, [340](#)
  - GetMagnitude, [340](#)
  - GetRealArray, [341](#)
  - GetUnitRvector, [341](#)
  - IsZeroVector, [341](#)
  - MakeZeroVector, [341](#)
  - Max, [341](#)
  - Min, [341](#)
  - Norm, [342](#)
  - Normalize, [342](#)
  - operator!=, [342](#)
  - operator<<, [347](#)
  - operator>>, [347](#)
  - operator\*, [342](#), [346](#)
  - operator\*==, [343](#)
  - operator+, [343](#)
  - operator+=, [343](#)
  - operator-, [343](#)
  - operator-=, [344](#)
  - operator/, [344](#)
  - operator/==, [344](#)
  - operator/, [344](#)
  - operator=, [344](#)
  - operator==, [344](#)
  - Outerproduct, [347](#)
  - Rmatrix, [320](#), [347](#)
  - Rvector, [339](#), [340](#)
  - Set, [345](#)
  - Sort, [345](#)
  - ToString, [346](#)
- Rvector.cpp
  - operator<, [527](#)
  - operator<<, [527](#)
  - operator>>, [528](#)
  - operator\*, [527](#)
  - Outerproduct, [528](#)
- Rvector.hpp
  - operator<, [529](#)
- Rvector3, [348](#)
  - ~Rvector3, [350](#)
  - ComputeLongitudeLatitude, [350](#)
  - Copy, [350](#)
  - Cross, [354](#)
  - Get, [351](#)
  - GetDataDescriptions, [351](#)
  - GetMagnitude, [351](#)
  - GetNumData, [351](#)
  - GetUnitVector, [351](#)
  - Normalize, [351](#)
  - operator!=, [352](#)
  - operator\*, [352](#), [354](#)
  - operator\*==, [352](#)
  - operator+, [352](#)
  - operator+=, [353](#)
  - operator-, [353](#)
  - operator-=, [353](#)
  - operator/, [353](#)
  - operator/==, [353](#), [354](#)
  - operator=, [354](#)
  - operator==, [354](#)
  - Outerproduct, [355](#)

- Rmatrix, [320](#)
- Rmatrix33, [328](#), [355](#)
- Rvector3, [349](#), [350](#)
- Set, [354](#)
- Rvector3.cpp
  - Cross, [530](#)
  - operator<<, [530](#)
  - operator>>, [530](#)
  - operator\*, [530](#)
  - Outerproduct, [531](#)
- Rvector6, [355](#)
  - ~Rvector6, [358](#)
  - Clone, [358](#)
  - Get, [358](#)
  - GetDataDescriptions, [358](#)
  - GetNumData, [359](#)
  - GetR, [359](#)
  - GetV, [359](#)
  - IsValid, [360](#)
  - operator!=, [360](#)
  - operator\*, [360](#)
  - operator\*=: [360](#), [361](#)
  - operator+, [361](#)
  - operator+=, [361](#)
  - operator-, [361](#)
  - operator-=, [361](#)
  - operator/, [361](#), [362](#)
  - operator/=, [362](#)
  - operator=, [362](#)
  - operator==, [362](#)
  - RVECTOR6\_UNDEFINED, [363](#)
  - Rmatrix66, [337](#), [363](#)
  - Rvector6, [357](#), [358](#)
  - Set, [362](#), [363](#)
  - SetR, [363](#)
  - SetV, [363](#)
  - UTIL\_REAL\_UNDEFINED, [364](#)
- Rvector6.hpp
  - NUM\_DATA\_INIT, [533](#)
- Rvector::ZeroVector, [438](#)
  - ZeroVector, [439](#)
- SATURN\_NAME
  - GmatSolarSystemDefaults, [72](#)
- SECS\_PER\_DAY
  - GmatTimeConstants, [106](#)
- SECS\_PER\_HOUR
  - GmatTimeConstants, [106](#)
- SECS\_PER\_MINUTE
  - GmatTimeConstants, [106](#)
- SHORT\_REAL\_RADIX
  - GmatRealConstants, [53](#)
- SIGN
  - GmatMathConstants, [37](#)
- SOLAR\_SYSTEM\_BARYCENTER\_NAME
  - GmatSolarSystemDefaults, [72](#)
- SPEED\_OF\_LIGHT\_VACUUM
  - GmatPhysicalConstants, [50](#)
- SSB\_MU
  - GmatSolarSystemDefaults, [72](#)
- SSB\_NAIF\_ID
  - GmatSolarSystemDefaults, [72](#)
- STAR\_EQUATORIAL\_RADIUS
  - GmatSolarSystemDefaults, [72](#)
- STAR\_FLATTENING
  - GmatSolarSystemDefaults, [72](#)
- STAR\_MU
  - GmatSolarSystemDefaults, [73](#)
- STAR\_NAIF\_IDS
  - GmatSolarSystemDefaults, [73](#)
- STAR\_ORIENTATION\_PARAMETERS
  - GmatSolarSystemDefaults, [73](#)
- STAR\_PHOTOSPHERE\_RADIUS
  - GmatSolarSystemDefaults, [73](#)
- STAR\_RADIANT\_POWER
  - GmatSolarSystemDefaults, [73](#)
- STAR\_REFERENCE\_DISTANCE
  - GmatSolarSystemDefaults, [73](#)
- STAR\_SPICE\_FRAME\_ID
  - GmatSolarSystemDefaults, [73](#)
- STAR\_TWO\_BODY\_ELEMENTS
  - GmatSolarSystemDefaults, [74](#)
- STAR\_TWO\_BODY\_EPOCH
  - GmatSolarSystemDefaults, [74](#)
- SUN\_NAME
  - GmatSolarSystemDefaults, [74](#)
- scaleHeight
  - ExponentialAtmosphere, [192](#)
- SchurFactorization, [364](#)
  - ~SchurFactorization, [366](#)
  - Factor, [366](#)
  - Invert, [366](#), [367](#)
  - operator=, [367](#)
  - RemoveRowCol, [367](#)
  - RestoreAllRowCols, [368](#)
  - SchurFactorization, [365](#)
- Sec
  - GmatTime, [239](#)
- second
  - GmatTimeUtil::CalDate, [143](#)
- seconds
  - GmatTimeUtil::ElapsedDate, [176](#)
- secondsOfDayD
  - Date, [162](#)
- SeparateBrackets
  - GmatStringUtil, [95](#)
- SeparateBy
  - GmatStringUtil, [96](#)
- SeparateByComma
  - GmatStringUtil, [96](#)
- SeparateDots
  - GmatStringUtil, [96](#)
- Set
  - A1Mjd, [119](#)
  - ElapsedTime, [180](#)
  - Rmatrix33, [327](#)
  - Rmatrix66, [335](#)

- Rvector, [345](#)
- Rvector3, [354](#)
- Rvector6, [362](#), [363](#)
- StringTokenizer, [404](#)
- SetAbsPathname
  - FileManager, [209](#), [210](#)
- SetActualFormat
  - GmatGlobal, [223](#)
- SetAppendEol
  - GmatGlobal, [223](#)
- SetBatchMode
  - GmatGlobal, [223](#)
- SetBinDirectory
  - FileManager, [210](#)
- SetBinaryIn
  - GmatGlobal, [223](#)
- SetBinaryOut
  - GmatGlobal, [223](#)
- SetClockSeed
  - RandomNumber, [306](#)
- SetCommandEchoMode
  - GmatGlobal, [224](#)
- SetConstants
  - ExponentialAtmosphere, [191](#)
- SetCurrentFormat
  - GmatGlobal, [224](#)
- SetCurrentWorkingDirectory
  - FileManager, [210](#)
  - GmatFileUtil, [36](#)
- SetDataPrecision
  - GmatGlobal, [224](#)
- SetDataWidth
  - GmatGlobal, [224](#)
- SetDate
  - GregorianCalendar, [243](#)
- SetDays
  - GmatTime, [237](#)
- SetDefaultFormat
  - GmatGlobal, [224](#)
- SetDelimiters
  - StringTokenizer, [405](#)
- SetDetailedRunState
  - GmatGlobal, [225](#)
- SetDetails
  - BaseException, [140](#)
- SetElement
  - ArrayTemplate, [128](#)
  - TableTemplate, [410](#)
- SetEopFile
  - GmatGlobal, [225](#)
  - TimeSystemConverter, [426](#)
- SetEventLocationAvailable
  - GmatGlobal, [225](#)
- SetExtrapolation
  - Interpolator, [255](#)
- SetFatal
  - BaseException, [141](#)
- SetForceInterpolation
  - Interpolator, [255](#)
- SetFracSec
  - GmatTime, [238](#)
- SetGmatWorkingDirectory
  - FileManager, [211](#)
- SetGuiMode
  - GmatGlobal, [225](#)
- SetHorizontal
  - GmatGlobal, [225](#)
- SetIncludeFoundInScriptResource
  - GmatGlobal, [225](#)
- SetIntegerWidth
  - GmatGlobal, [226](#)
- SetItrfCoefficientsFile
  - GmatGlobal, [226](#)
- SetLeapSecsFileReader
  - TimeSystemConverter, [426](#)
- SetLogEnable
  - ConsoleMessageReceiver, [153](#)
  - MessageInterface, [286](#)
  - MessageReceiver, [291](#)
- SetLogFile
  - ConsoleMessageReceiver, [153](#)
  - MessageInterface, [286](#)
  - MessageReceiver, [291](#)
- SetLogPath
  - ConsoleMessageReceiver, [153](#)
  - MessageInterface, [286](#)
  - MessageReceiver, [291](#)
- SetLogfileName
  - GmatGlobal, [226](#)
- SetLogfileSource
  - GmatGlobal, [226](#)
- SetMatlabAvailable
  - GmatGlobal, [226](#)
- SetMatlabDebug
  - GmatGlobal, [226](#)
- SetMatlabFuncNameExt
  - GmatGlobal, [227](#)
- SetMatlabMode
  - GmatGlobal, [227](#)
- SetMessage
  - BaseException, [141](#)
- SetMessageReceiver
  - MessageInterface, [286](#)
- SetMessageType
  - BaseException, [141](#)
- SetMissionTreeDebug
  - GmatGlobal, [227](#)
- SetMjdString
  - GmatTime, [238](#)
- SetNitsClient
  - GmatGlobal, [227](#)
- SetOutputPath
  - GmatGlobal, [227](#)
- SetPlotMode
  - GmatGlobal, [227](#)
- SetPrefix

- GmatGlobal, [227](#)
- SetRange
  - Interpolator, [255](#)
- SetReal
  - A1Mjd, [119](#)
- SetRunInterrupted
  - GmatGlobal, [228](#)
- SetRunMode
  - GmatGlobal, [228](#)
- SetRunState
  - GmatGlobal, [228](#)
- SetScientific
  - GmatGlobal, [228](#)
- SetSec
  - GmatTime, [238](#)
- SetSeed
  - GmatMathUtil, [48](#)
  - RandomNumber, [306](#)
- SetShowPoint
  - GmatGlobal, [228](#)
- SetSize
  - ArrayTemplate, [128](#)
  - TableTemplate, [411](#)
- SetSkipSplashMode
  - GmatGlobal, [228](#)
- SetSpacing
  - GmatGlobal, [229](#)
- SetTimeInSec
  - GmatTime, [238](#)
- SetTimePrecision
  - GmatGlobal, [229](#)
- SetTimeWidth
  - GmatGlobal, [229](#)
- SetToCurrentFormat
  - GmatGlobal, [229](#)
- SetToDefaultFormat
  - GmatGlobal, [229](#)
- SetType
  - GregorianDate, [244](#)
- SetUndefined
  - Rmatrix66, [335](#)
- SetWriteFilePathInfo
  - GmatGlobal, [229](#)
- SetWriteGmatKeyword
  - GmatGlobal, [229](#)
- SetWriteParameterInfo
  - GmatGlobal, [230](#)
- SetWritePersonalizationFile
  - GmatGlobal, [230](#)
- SetR
  - Rvector6, [363](#)
- SetV
  - Rvector6, [363](#)
- ShowMessage
  - ConsoleMessageReceiver, [154](#)
  - MessageInterface, [287](#)
  - MessageReceiver, [292](#)
- ShowPoint
  - GmatGlobal, [230](#)
- SignOf
  - GmatMathUtil, [48](#)
- Sin
  - GmatMathUtil, [48](#)
- SinXOverX
  - GmatMathUtil, [49](#)
- Sinh
  - GmatMathUtil, [49](#)
- sizeD
  - ArrayTemplate, [129](#)
- SkewSymmetric
  - Rmatrix33, [328](#)
  - Rmatrix33.cpp, [522](#)
  - Rmatrix66, [337](#)
  - Rmatrix66.cpp, [525](#)
- SkewSymmetric4by4
  - Rmatrix, [320](#)
  - Rmatrix.cpp, [520](#)
- SkipHeaderLines
  - GmatFileUtil, [37](#)
- SkipSplashMode
  - GmatGlobal, [230](#)
- Smooth
  - ExponentialAtmosphere, [191](#)
- smoothDensity
  - ExponentialAtmosphere, [192](#)
- SolveSystem
  - LUFactorization, [279](#)
- Sort
  - Rvector, [345](#)
- SphericalAZFPAToCartesian
  - StateConversionUtil, [399](#)
- SphericalEllipsoidToCartesian
  - BodyFixedStateConverterUtil, [17](#)
- SphericalEllipsoidToSpherical
  - BodyFixedStateConverterUtil, [19](#)
- SphericalRADECToCartesian
  - StateConversionUtil, [400](#)
- SphericalToCartesian
  - BodyFixedStateConverterUtil, [19](#)
- SphericalToSphericalEllipsoid
  - BodyFixedStateConverterUtil, [20](#)
- Sqrt
  - GmatMathUtil, [49](#)
- startPoint
  - LagrangeInterpolator, [270](#)
- StartsWith
  - GmatStringUtil, [96](#)
- StateArray
  - gmatdefs.hpp, [450](#)
  - utildefs.hpp, [456](#)
- StateConversionUtil, [368](#)
  - AltEquinoctialToEquinoctial, [372](#)
  - AnomalyType, [371](#)
  - BrouwerMeanLongToCartesian, [372](#)
  - BrouwerMeanLongToOsculatingElements, [373](#)
  - BrouwerMeanShortToCartesian, [373](#)

- BrouwerMeanShortToOsculatingElements, 373
- CalculateEccentricAnomaly, 374
- CartesianToAOP, 375
- CartesianToAngularMomentum, 374
- CartesianToBrouwerMeanLong, 375
- CartesianToBrouwerMeanShort, 375
- CartesianToDirOfLineOfNode, 376
- CartesianToECC, 376
- CartesianToEA, 376
- CartesianToEccVector, 376
- CartesianToEquinoctial, 376
- CartesianToHA, 377
- CartesianToINC, 377
- CartesianToIncomingAsymptote, 377
- CartesianToKeplerian, 377–379
- CartesianToKeplerianDerivativeConversion, 380
- CartesianToKeplerianDerivativeConversion\_↔  
FiniteDiff, 380
- CartesianToKeplerianDerivativeConversionWith↔  
KeplInput, 380
- CartesianToKeplerianDerivativeConversionWith↔  
KeplInput\_FiniteDiff, 381
- CartesianToMA, 381
- CartesianToModEquinoctial, 381
- CartesianToOutgoingAsymptote, 382
- CartesianToPlanetodetic, 382
- CartesianToRAAN, 382
- CartesianToSMA, 383
- CartesianToSphericalAZFPA, 383
- CartesianToSphericalRADEC, 383
- CartesianToTA, 383
- Convert, 384
- ConvertFromAltEquinoctial, 386
- ConvertFromBrouwerMeanLong, 386
- ConvertFromBrouwerMeanShort, 386
- ConvertFromCartesian, 386
- ConvertFromDelaunay, 387
- ConvertFromEquinoctial, 387
- ConvertFromIncomingAsymptote, 387
- ConvertFromKeplerian, 387
- ConvertFromModEquinoctial, 388
- ConvertFromModKeplerian, 388
- ConvertFromOutgoingAsymptote, 388
- ConvertFromPlanetodetic, 388
- ConvertFromSphericalAZFPA, 389
- ConvertFromSphericalRADEC, 389
- ConvertFromTrueAnomaly, 389
- ConvertToTrueAnomaly, 389, 390
- DelaunayToKeplerian, 390
- EccentricToTrueAnomaly, 390
- EquinoctialToAltEquinoctial, 390
- EquinoctialToCartesian, 392
- GetAnomalyLongText, 392
- GetAnomalyShortText, 392
- GetAnomalyType, 393
- GetLongTypeNameList, 393
- GetStateTypeList, 393
- GetTypeCount, 394
- HyperbolicToTrueAnomaly, 394
- IncomingAsymptoteToCartesian, 394
- IsRvValid, 394
- IsValidAnomalyType, 395
- KeplerianToCartesian, 395, 396
- KeplerianToDelaunay, 396
- KeplerianToModKeplerian, 396
- MeanToTrueAnomaly, 397
- ModEquinoctialToCartesian, 397
- ModKeplerianToKeplerian, 397
- OutgoingAsymptoteToCartesian, 398
- PlanetodeticToCartesian, 398
- RequiresCelestialBodyOrigin, 398
- RequiresFixedCoordinateSystem, 399
- SphericalAZFPAToCartesian, 399
- SphericalRADECToCartesian, 400
- StateType, 371
- TrueToEccentricAnomaly, 400
- TrueToHyperbolicAnomaly, 400
- TrueToMeanAnomaly, 401
- ValidateValue, 401
- StateConversionUtil.hpp
  - EARTH\_EQ\_RADIUS, 535
  - EARTH\_FLATTENING, 535
  - EARTH\_MU, 535
- StateElementId
  - Gmat, 24
- StateType
  - StateConversionUtil, 371
- StringArray
  - gmatdefs.hpp, 451
  - utildefs.hpp, 456
- StringToWideString
  - GmatStringUtil, 96
- StringTokenizer, 401
  - ~StringTokenizer, 403
  - CountTokens, 403
  - GetAllTokens, 403
  - GetDelimiters, 403
  - GetToken, 404
  - Set, 404
  - SetDelimiters, 405
  - StringTokenizer, 402
- stringVals
  - Rmatrix, 321
- stringValue
  - Date, 162
- Strip
  - GmatStringUtil, 96
- StripType
  - GmatStringUtil, 79
- Subtract
  - A1Mjd, 120
- SubtractSeconds
  - GmatTime, 238
- subtype
  - Gmat::PluginResource, 297
- Symmetric

- Rmatrix, 317
- Rmatrix33, 327
- Rmatrix66, 336
- T TT\_COEFF1
  - TimeSystemConverter, 428
- T TT\_OFFSET
  - TimeSystemConverter, 428
- TDB\_COEFF1
  - TimeSystemConverter, 428
- TDB\_COEFF2
  - TimeSystemConverter, 428
- TELESTO\_NAME
  - GmatSolarSystemDefaults, 74
- TETHYS\_NAME
  - GmatSolarSystemDefaults, 74
- THALASSA\_NAME
  - GmatSolarSystemDefaults, 74
- THEBE\_NAME
  - GmatSolarSystemDefaults, 74
- TIME\_OF\_J2000
  - GmatTimeConstants, 106
- TIME\_PRECISION
  - GmatGlobal, 231
- TIME\_SYSTEM\_TEXT
  - TimeSystemConverter, 428
- TIME\_WIDTH
  - GmatGlobal, 231
- TITAN\_NAME
  - GmatSolarSystemDefaults, 75
- TITANIA\_NAME
  - GmatSolarSystemDefaults, 75
- TRITON\_NAME
  - GmatSolarSystemDefaults, 75
- TT\_TAI\_OFFSET
  - GmatTimeConstants, 106
- TableAlreadySized
  - TableTemplateExceptions::TableAlreadySized, 406
- tableSz
  - EopFile, 187
- TableTemplate
  - ~TableTemplate, 408
  - ChangeSize, 408
  - colsD, 411
  - elementD, 411
  - GetDataVector, 408
  - GetElement, 409
  - GetNumColumns, 409
  - GetNumRows, 409
  - GetSize, 409
  - init, 409
  - IsSized, 409
  - isSizedD, 411
  - operator!=, 410
  - operator(), 410
  - operator=, 410
  - operator==, 410
  - rowsD, 411
  - SetElement, 410
  - SetSize, 411
  - TableTemplate, 407, 408
  - TableTemplate< T >, 406
  - TableTemplateExceptions, 412
  - TableTemplateExceptions::DimensionError, 172
    - DimensionError, 173
  - TableTemplateExceptions::IllegalSize, 245
    - IllegalSize, 246
  - TableTemplateExceptions::OutOfBounds, 294
    - OutOfBounds, 295
  - TableTemplateExceptions::TableAlreadySized, 405
    - TableAlreadySized, 406
  - TableTemplateExceptions::UnsizeTable, 431
    - UnsizeTable, 432
- taiMJD
  - LeapSecondInformation, 272
- taiTime
  - EopFile, 187
- Tan
  - GmatMathUtil, 49
- Tanh
  - GmatMathUtil, 49
- theEopFile
  - TimeSystemConverter, 428
- theLeapSecsFileReader
  - TimeSystemConverter, 428
- theTimeConverter
  - EopFile, 187
  - TimeSystemConverter, 428
- TimeException, 412
  - TimeException, 413
- TimeFileException, 413
  - TimeFileException, 414
- TimeFormatException, 414
  - TimeFormatException, 415
- TimeRangeError
  - Date::TimeRangeError, 416
- TimeSystemConverter, 417
  - Convert, 419–421
  - ConvertFromTaiMjd, 422
  - ConvertGregorianToMjd, 422
  - ConvertGregorianToMjdGT, 423
  - ConvertMjdToGregorian, 423
  - ConvertToTaiMjd, 424
  - GetFirstLeapSecondMJD, 424
  - GetTimeSystemAndFormat, 425
  - GetTimeTypeID, 425
  - GetValidTimeRepresentations, 425
  - Instance, 425
  - IsInLeapSecond, 425
  - IsValidTimeSystem, 425
  - L\_B, 427
  - M\_E\_COEFF1, 427
  - M\_E\_OFFSET, 427
  - NUM\_SECS, 427
  - NumberOfLeapSecondsFrom, 426
  - SetEopFile, 426
  - SetLeapSecsFileReader, 426



- T TT\_COEFF1, [428](#)
- T TT\_OFFSET, [428](#)
- TDB\_COEFF1, [428](#)
- TDB\_COEFF2, [428](#)
- TIME\_SYSTEM\_TEXT, [428](#)
- theEopFile, [428](#)
- theLeapSecsFileReader, [428](#)
- theTimeConverter, [428](#)
- TimeSystemConverter, [419](#)
- TimeSystemTypes, [419](#)
- ValidateTimeFormat, [427](#)
- ValidateTimeSystem, [427](#)
- TimeSystemTypes
  - TimeSystemConverter, [419](#)
- TimeTypes.hpp
  - DayOfMonth, [547](#)
  - DayOfYear, [547](#)
  - HourOfDay, [547](#)
  - MinuteOfHour, [548](#)
  - MonthOfYear, [548](#)
  - Ut1Mjd, [548](#)
  - UtcMjd, [548](#)
  - YearNumber, [548](#)
- tkp
  - geoparms, [212](#)
- ToA1Date
  - A1Mjd, [120](#)
- ToA1Mjd
  - UtcDate, [436](#)
- ToAngularVelocity
  - AttitudeConversionUtility, [131](#)
- ToBoolean
  - GmatStringUtil, [97](#)
- ToBooleanArray
  - GmatStringUtil, [97](#)
- ToCosineMatrix
  - AttitudeConversionUtility, [133](#), [134](#)
- ToDOYFromYearMonthDay
  - DateUtil, [167](#)
  - DateUtil.cpp, [471](#)
  - DateUtil.hpp, [476](#)
- ToDayOfYear
  - Date, [160](#)
- ToElapsedDate
  - ElapsedTime, [180](#)
- ToEulerAngleRates
  - AttitudeConversionUtility, [134](#)
- ToEulerAngles
  - AttitudeConversionUtility, [135](#)
- ToHMSFromSecondsOfDay
  - DateUtil, [167](#)
  - DateUtil.cpp, [471](#)
  - DateUtil.hpp, [477](#)
- ToInteger
  - GmatStringUtil, [97](#)
- ToIntegerArray
  - GmatStringUtil, [98](#)
- ToLower
  - GmatStringUtil, [98](#)
- ToMRPs
  - AttitudeConversionUtility, [136](#)
- ToMonthDayFromYearDOY
  - DateUtil, [167](#)
  - DateUtil.cpp, [471](#)
  - DateUtil.hpp, [477](#)
- ToOnOff
  - GmatStringUtil, [98](#)
- ToOrdinal
  - GmatStringUtil, [98](#)
- ToPackedCalendarReal
  - Date, [160](#)
- ToPackedCalendarString
  - Date, [160](#)
- ToPackedHHMMSS
  - Date, [160](#)
- ToPackedYYMMDD
  - Date, [160](#)
- ToQuaternion
  - AttitudeConversionUtility, [136](#), [137](#)
- ToReal
  - GmatStringUtil, [98](#), [99](#)
- ToRealArray
  - GmatStringUtil, [99](#)
- ToRowString
  - Rmatrix, [317](#)
- ToSecondsOfDayFromHMS
  - DateUtil, [168](#)
  - DateUtil.cpp, [472](#)
  - DateUtil.hpp, [477](#)
- ToString
  - GmatRealUtil, [56](#)
  - GmatStringUtil, [99](#), [100](#)
  - GmatTime, [238](#)
  - Rmatrix, [318](#)
  - Rvector, [346](#)
- ToStringArray
  - GmatStringUtil, [100](#)
- ToStringNoZeros
  - GmatStringUtil, [100](#)
- ToUnsignedInt
  - GmatStringUtil, [101](#)
- ToUnsignedIntArray
  - GmatStringUtil, [101](#)
- ToUpper
  - GmatStringUtil, [101](#)
- ToUtcDate
  - A1Mjd, [120](#)
- ToUtcMjd
  - A1Mjd, [121](#)
- ToValueStrings
  - A1Mjd, [121](#)
  - Date, [160](#)
  - ElapsedTime, [180](#)
- ToYearDOYHourMinSec
  - Date, [161](#)
- ToYearMonthDayHourMinSec



- Date, [161](#)
- toolkit
  - Gmat::PluginResource, [297](#)
- Trace
  - Rmatrix, [318](#)
  - Rmatrix33, [328](#)
  - Rmatrix66, [336](#)
- Transpose
  - Rmatrix, [318](#)
  - Rmatrix33, [328](#)
  - Rmatrix66, [336](#)
- TransposeTimesMatrix
  - Rmatrix, [320](#)
  - Rmatrix.cpp, [520](#)
  - Rmatrix33, [329](#)
  - Rmatrix66, [337](#)
  - Rmatrix66.cpp, [525](#)
- TransposeTimesRmatrix
  - Rmatrix33.cpp, [523](#)
- TransposeTimesTranspose
  - Rmatrix, [320](#)
  - Rmatrix.cpp, [520](#)
  - Rmatrix33, [329](#)
  - Rmatrix33.cpp, [523](#)
  - Rmatrix66, [337](#)
  - Rmatrix66.cpp, [525](#)
- trigger
  - Gmat::PluginResource, [297](#)
- Trim
  - GmatStringUtil, [101](#)
- TrueToEccentricAnomaly
  - StateConversionUtil, [400](#)
- TrueToHyperbolicAnomaly
  - StateConversionUtil, [400](#)
- TrueToMeanAnomaly
  - StateConversionUtil, [401](#)
- type
  - Gmat::PluginResource, [297](#)
- UMBRIEL\_NAME
  - GmatSolarSystemDefaults, [75](#)
- UNIVERSAL\_GRAVITATIONAL\_CONSTANT
  - GmatPhysicalConstants, [50](#)
- URANUS\_NAME
  - GmatSolarSystemDefaults, [75](#)
- UTIL\_REAL\_UNDEFINED
  - Rvector6, [364](#)
- Uniform
  - RandomNumber, [306](#)
- UniformArray
  - RandomNumber, [307](#)
- UnimplementedException, [429](#)
  - UnimplementedException, [430](#)
- UnpackDate
  - DateUtil, [168](#)
  - DateUtil.cpp, [472](#)
  - DateUtil.hpp, [477](#)
- UnpackDateWithDOY
  - DateUtil, [168](#)
- DateUtil.cpp, [472](#)
- DateUtil.hpp, [478](#)
- UnpackTime
  - DateUtil, [169](#)
  - DateUtil.cpp, [473](#)
  - DateUtil.hpp, [478](#)
- UnsignedInt
  - gmatdefs.hpp, [451](#)
  - utildefs.hpp, [456](#)
- UnsignedIntArray
  - gmatdefs.hpp, [451](#)
  - utildefs.hpp, [456](#)
- UnsignedArray
  - ArrayTemplateExceptions::UnsignedArray, [431](#)
- UnsignedTable
  - TableTemplateExceptions::UnsignedTable, [432](#)
- UpdateBeginAndEndIndex
  - LagrangeInterpolator, [269](#)
- UpperLeft
  - Rmatrix66, [336](#)
- UpperRight
  - Rmatrix66, [336](#)
- Ut1Mjd
  - TimeTypes.hpp, [548](#)
- ut1UtcOffsets
  - EopFile, [187](#)
- UtcDate, [433](#)
  - ~UtcDate, [435](#)
  - operator=, [436](#)
  - ToA1Mjd, [436](#)
  - UtcDate, [434](#), [435](#)
- UtcMjd
  - TimeTypes.hpp, [548](#)
- UtcMjdToA1Mjd
  - A1Mjd, [121](#)
- utildefs.hpp
  - BooleanArray, [454](#)
  - Byte, [454](#)
  - ColorMap, [454](#)
  - EpochArray, [454](#)
  - GMATUTIL\_API, [453](#)
  - GmatEpoch, [454](#)
  - Integer, [454](#)
  - IntegerArray, [454](#)
  - IntegerMap, [454](#)
  - ObjectArray, [455](#)
  - ObjectMap, [455](#)
  - ObjectMapStack, [455](#)
  - ObjectTypeArray, [455](#)
  - ObjectTypeArrayMap, [455](#)
  - ObjectTypeMap, [455](#)
  - Radians, [455](#)
  - Real, [456](#)
  - RealArray, [456](#)
  - StateArray, [456](#)
  - StringArray, [456](#)
  - UnsignedInt, [456](#)
  - UnsignedIntArray, [456](#)

- WrapperArray, [456](#)
- WrapperMap, [457](#)
- WrapperTypeArray, [457](#)
- UtilityException, [437](#)
  - UtilityException, [437](#)
- VENUS\_NAME
  - GmatSolarSystemDefaults, [75](#)
- ValidatePaths
  - FileManager, [211](#)
- ValidateTimeFormat
  - TimeSystemConverter, [427](#)
- ValidateTimeSystem
  - TimeSystemConverter, [427](#)
- ValidateValue
  - StateConversionUtil, [401](#)
- value
  - Element, [181](#)
- WideStringToString
  - GmatStringUtil, [101](#), [102](#)
- widgetType
  - Gmat::PluginResource, [297](#)
- WrapperArray
  - gmatdefs.hpp, [451](#)
  - utildefs.hpp, [456](#)
- WrapperDataType
  - Gmat, [25](#)
- WrapperMap
  - gmatdefs.hpp, [451](#)
  - utildefs.hpp, [457](#)
- WrapperTypeArray
  - utildefs.hpp, [457](#)
- WriteMode
  - Gmat, [25](#)
- WriteStartupFile
  - FileManager, [211](#)
- WriteStringArray
  - GmatStringUtil, [102](#)
- x
  - LagrangeInterpolator, [270](#)
- xtemp
  - geoparms, [212](#)
- y
  - LagrangeInterpolator, [270](#)
- year
  - GmatTimeUtil::CalDate, [143](#)
- YearNumber
  - TimeTypes.hpp, [548](#)
- yearD
  - Date, [163](#)
- ZeroVector
  - Rvector::ZeroVector, [439](#)