GMAT Source

Generated by Doxygen 1.8.13

Contents

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 to severe for the release. It is commented out so that post release, we can asses 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:

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

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

Chapter 5

Class Index

5.1 Class List

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

Chapter 6

File Index

6.1 File List

Here is a list of all files with brief descriptions:

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

<cart>	data in cartesian
<flattening>	flattening coefficient for the body
<meanRadius>	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

<cart>	data in cartesian
<flattening>	flattening coefficient for the body
<meanRadius>	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

<origValue>	data in given representation
<fromType>	representation from which to convert
<toType>	representation 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

7.1.1.4 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

<*sphEll*>	data in spherical-ellipsoid
<*flattening*>	flattening coefficient for the body
<*meanRadius*>	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

<*sphEll*>	data in spherical-ellipsoid
<*flattening*>	flattening coefficient for the body
<*meanRadius*>	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

<*spherical*>	data in spherical
<*flattening*>	flattening coefficient for the body
<*meanRadius*>	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

<*spherical*>	data in spherical
<*flattening*>	flattening coefficient for the body
<*meanRadius*>	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↩ G_	
INFO_	
DEBUG_	
GENERA↩ 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	**Todo** : 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 &baseIn, 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 &file1Diff←Count, 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 &file1←DiffCount, 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=COM←PARE_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 embeded 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 embeded 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 embeded 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

appName	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 extention 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() [1/2]

```
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=Gmat↩ Global::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::DA↩ TA_WIDTH)
- GMATUTIL_API std::string ToString (const Integer &ival, bool useCurrentFormat=true, Integer width=Gmat↩ Global::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

throws	RealUtilitiesExceptions::ArgumentError if all three Cartesian coordinates to be converted are 0

7.13.2.2 CartesianToRaDec()

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

Exceptions

throws	RealUtilitiesExceptions::ArgumentError 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

output	Output stream
a	Rvector 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

output	Output stream
a	Rmatrix 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 ADRASTEA_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 = "Promethus"
- 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 ADRASTEA_NAME

```
const std::string GmatSolarSystemDefaults::ADRASTEA_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:

```
=
   {
      2.43970000000000e+003,
      6.05190000000000e+003,
      6.3781363E3,
      3.39700000000000e+003,
      7.14920000000000e+004,
      6.02680000000000e+004,
      2.55590000000000e+004,
      2.52690000000000e+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::ASTRONO↩
MICAL_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 removeAll=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=TRAILING)
- 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 startIndex=0)
- GMATUTIL_API std::string Replace (const std::string &str, const std::string &from, const std::string &to, std::string::size_type startIndex=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::DATA_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_WIDTH)
- GMATUTIL_API std::string ToString (const Integer &val, bool useCurrentFormat=true, Integer width=GmatGlobal::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 &close↩ Bracket, 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 ignore↩ Comma=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 blankNameIsOk=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 blankNameIsOk=false)
- GMATUTIL_API bool IsValidName (const std::string &str, bool ignoreBracket=false, bool blankNameIs↩ Ok=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 ignoreSpace↩ AfterQuote=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 blank↩ NameIsOk=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

str	Input string
enStr	Enclosing string

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

value	The Real that needs to be serialized
useExp	Use scientific notation
length	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 )
```

Capitialize 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

str	Input string
delim	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

bracketPair	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: IsStringInside↩ Symbols(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 occurrance.

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

str	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

str	Input string
argStr	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 occurrance 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 occurance 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 precisoin format for numbers ending .999999.

Parameters

val	Real number
valStr	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 = 1.1574074074074074074074074074074e-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.

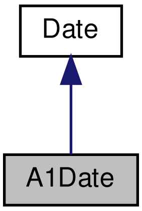
Chapter 8

Class Documentation

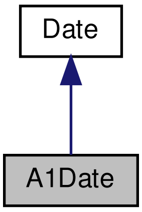
8.1 A1Date Class Reference

```
#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 doy, 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

<time>	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>()

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

Comparison operator >

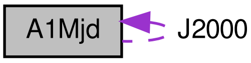
The documentation for this class was generated from the following files:

- 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

<a1mjd>	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 A1Mjd::GetNumData ( ) const
```

Returns

number of data elements.

8.2.3.5 GetReal()

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

8.2.3.6 operator"!=()

```
bool A1Mjd::operator!= (
             const A1Mjd & right ) const
```

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

Parameters

<*right*>	another object.

Returns

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

8.2.3.7 operator+() [1/2]

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

Subtracts a Real value and return the same object.

8.2.3.15 operator<()

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

8.2.3.16 operator<=()

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

8.2.3.17 operator=()

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

Assignment operator.

Parameters

<*right*>	the object to be copied.

Returns

reference to this object.

8.2.3.18 operator==()

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

Determins 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. Determins if this object value is greater than another object value.

Parameters

<right>	another object.

Returns

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

Parameters

<right>	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 A1Mjd::SetReal (
             Real val )  [inline]
```

8.2.3.23 Subtract() [1/2]

```
Real A1Mjd::Subtract (
             const A1Mjd & right ) const  [inline]
```

8.2.3.24 Subtract() [2/2]

```
Real A1Mjd::Subtract (
             const Real & right ) const  [inline]
```

8.2.3.25 ToA1Date()

```
A1Date A1Mjd::ToA1Date (
             bool handleLeapSecond = false )
```

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

8.2.3.26 ToUtcDate()

```
UtcDate A1Mjd::ToUtcDate ( )
```

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

Parameters

<*right*>	another object.

Returns

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

Parameters

<*right*>	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]
```

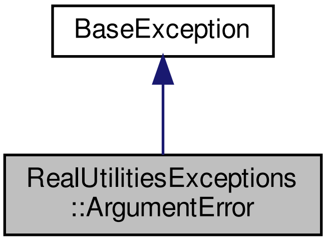
The documentation for this class was generated from the following files:

- GMATsrc/util/A1Mjd.hpp
- GMATsrc/util/A1Mjd.cpp

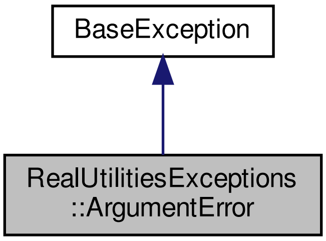
8.3 RealUtilitiesExceptions::ArgumentError Class Reference

```
#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]
```

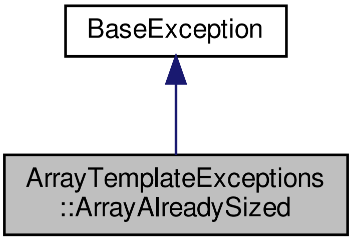
The documentation for this class was generated from the following file:

- GMATsrc/util/RealUtilities.hpp

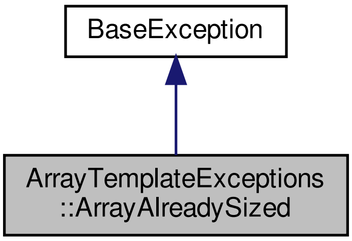
8.4 ArrayTemplateExceptions::ArrayAlreadySized Class Reference

```
#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]
```

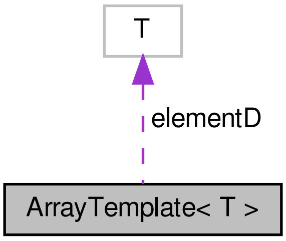
The documentation for this class was generated from the following file:

- 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 UnsizedArray

8.6.1 Detailed Description

Ccontains 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 of 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

eulerRates	the input euler angle rates (radians/sec)
eulerAngles	the input euler angles (radians)
seq1	first entry of the euler sequence
seq2	second entry of the euler sequence
seq3	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

quat1	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

eulerAngles	the input euler angles (radians)
seq1	first entry of the euler sequence
seq2	second entry of the euler sequence
seq3	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

eulerAngles	the input euler angles (radians)
seq1	first entry of the euler sequence
seq2	second entry of the euler sequence
seq3	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

angVel	the input angular velocity (radians/sec)
eulerAngles	the input euler angles (radians)
seq1	first entry of the euler sequence
seq2	second entry of the euler sequence
seq3	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

quat1	the input quaternion.
seq1	first entry of the euler sequence
seq2	second entry of the euler sequence
seq3	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

cosMat	the input cosine matrix.
seq1	first entry of the euler sequence
seq2	second entry of the euler sequence
seq3	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

quat1	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

eulerAngles	the input euler angles (radians)
seq1	first entry of the euler sequence
seq2	second entry of the euler sequence
seq3	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

cosMat	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 qc = q4.

Parameters

MRPs	the input MRP vector.

Returns

the quaternion representation of the input attitude.

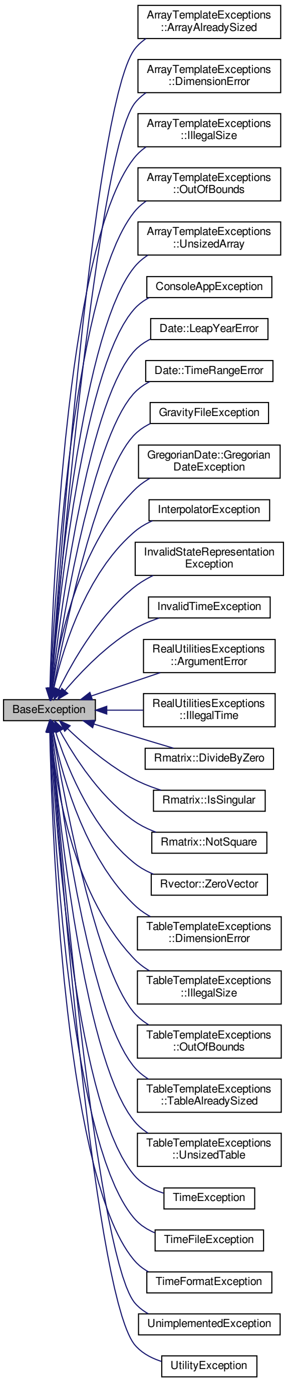
The documentation for this class was generated from the following files:

- GMATsrc/util/AttitudeConversionUtility.hpp
- GMATsrc/util/AttitudeConversionUtility.cpp

8.8 BaseException Class Reference

```
#include <BaseException.hpp>
```

Inheritance diagram for BaseException:


ArrayTemplateExceptions ::ArrayAlreadySized
ArrayTemplateExceptions ::DimensionError
ArrayTemplateExceptions ::IllegalSize
ArrayTemplateExceptions ::OutOfBounds
ArrayTemplateExceptions ::UnsizedArray
ConsoleAppException
Date::LeapYearError
Date::TimeRangeError
GravityFileException
GregorianDate::Gregorian DateException
InterpolatorException
InvalidStateRepresentation Exception
InvalidTimeException
RealUtilitiesExceptions ::ArgumentError
BaseException
RealUtilitiesExceptions ::IllegalTime
Rmatrix::DivideByZero
Rmatrix::IsSingular
Rmatrix::NotSquare
Rvector::ZeroVector
TableTemplateExceptions ::DimensionError
TableTemplateExceptions ::IllegalSize
TableTemplateExceptions ::OutOfBounds
TableTemplateExceptions ::TableAlreadySized
TableTemplateExceptions ::UnsizedTable
TimeException
TimeFileException
TimeFormatException
UnimplementedException
UtilityException

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
```

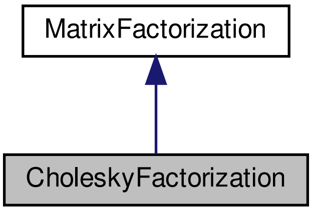
The documentation for this class was generated from the following file:

- GMATsrc/util/TimeTypes.hpp

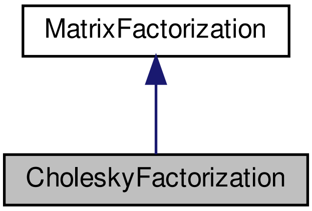
8.10 CholeskyFactorization Class Reference

```
#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

inputMatrix	The matrix to be factored, packed in upper triangular form
R	The matrix the factored result will be stored in
blankMatrix	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

inputMatrix	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

sum1	The matrix to be inverted, packed in upper triangular form
array_size	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

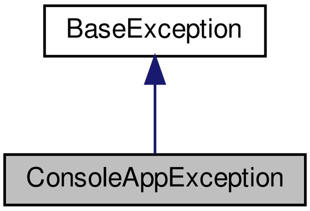
The documentation for this class was generated from the following files:

- GMATsrc/util/matrixoperations/CholeskyFactorization.hpp
- GMATsrc/util/matrixoperations/CholeskyFactorization.cpp

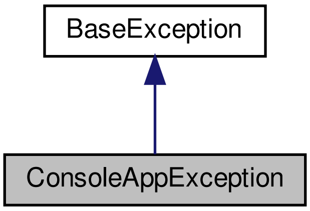
8.11 ConsoleAppException Class Reference

```
#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 ( )
```

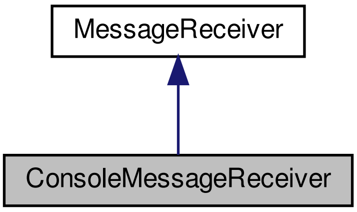
The documentation for this class was generated from the following files:

- GMATsrc/console/ConsoleAppException.hpp
- GMATsrc/console/ConsoleAppException.cpp

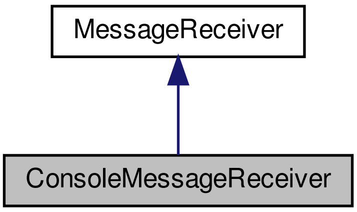
8.12 ConsoleMessageReceiver Class Reference

```
#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

msg	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

msg	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

msgType	The type of message that is displayed, selected from the set {ERROR_, WARNING_, INFO_} enumerated in the Gmat namespace.
msg	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

msgType	The type of message that is displayed, selected from the set {ERROR_, WARNING_, INFO_} enumerated in the Gmat namespace.
msg	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

flag	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

msgString	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

msg	The message, possibly including markers for variable argument substitution.
...	The optional list of parameters that are inserted into the msg string.

Implements MessageReceiver.

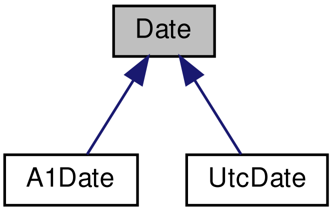
The documentation for this class was generated from the following files:

- 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 ToPackedYYYMMDD () 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 abstrace 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 ToPackedYYYMMDD()

```
Real Date::ToPackedYYYMMDD ( ) 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 abstrace 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

greg	input gregorian string
checkDate	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 Date

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 Date

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]
```

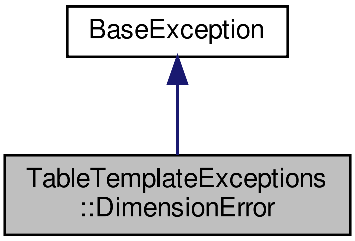
The documentation for this class was generated from the following files:

- GMATsrc/util/DateUtil.hpp
- GMATsrc/util/DateUtil.cpp

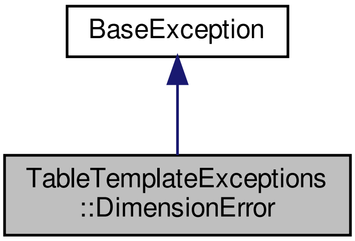
8.15 TableTemplateExceptions::DimensionError Class Reference

```
#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]
```

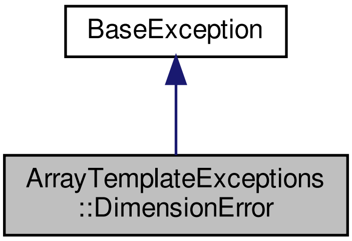
The documentation for this class was generated from the following file:

- GMATsrc/util/TableTemplate.hpp

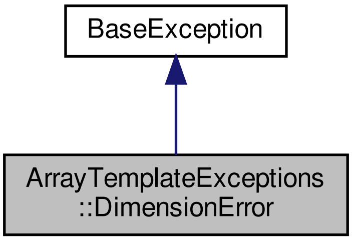
8.16 ArrayTemplateExceptions::DimensionError Class Reference

```
#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]
```

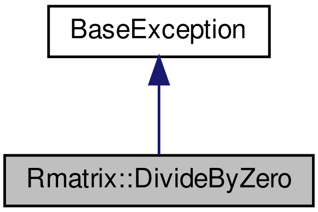
The documentation for this class was generated from the following file:

- GMATsrc/util/ArrayTemplate.hpp

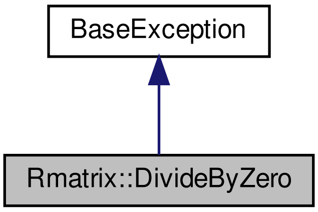
8.17 Rmatrix::DivideByZero Class Reference

```
#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
```

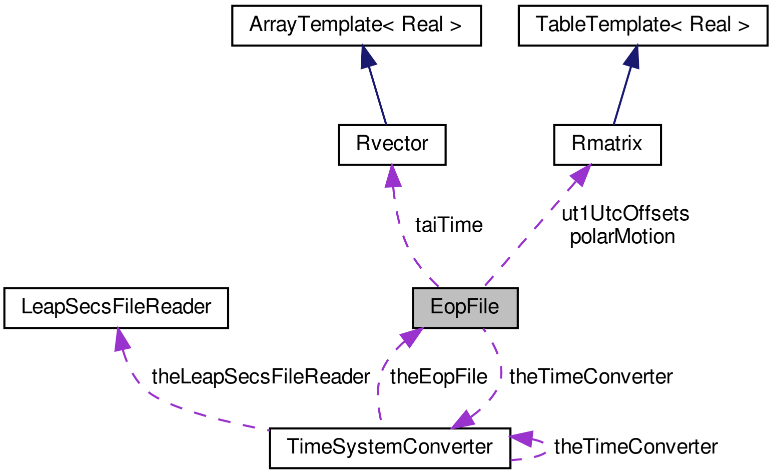
The documentation for this struct was generated from the following file:

- 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

fileNme	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

eopF	EopFile 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

forUtcMjd	time for which to return the data
xval	return X value of polar motion data (arcsec)
yval	return Y value of polar motion data (arcsec)
lodval	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

taiMjd	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

eopF	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, ρ_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 ρ_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

name	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*>	ExponentialAtmosphere 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 exonential 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 ExponentialAtmosphere object whose data to assign to "this" ExponentialAtmosphere.

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, ρ_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 write↩ Warning=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 overwrite↩ IfExists=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 &obj↩ Name, 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

path	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

suffix	The suffix to be appended to the setting
forEntries	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

fileName	The requested filename to be searched Enter blank name if default name to be used for the type
fileType	The file type name of the input file
forInput	Set to true if filename is for input
writeWarning	Set to true if warning should be written when no path found
writeInfo	Set to true if information should be written for output path (currently not used)
objName	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

thrown	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

thrown	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

thrown	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

thrown	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

thrown	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

<typeName>	file type name of which pathname to be returned.

Returns

pathname if type found.

Exceptions

thrown	if type cannot be found.

8.23.4.43 GetPathSeparator()

```
std::string FileManager::GetPathSeparator ( )
```

Returns

path separator; "/" or "\\" dependends 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 to severe for the release. It is commented out so that post release, we can asses 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 absoulute pathname for the type.

Parameters

<*type*>	file type of which path to be set.
<*newpath*>	new pathname.

Exceptions

thrown	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 absoulute pathname for the type.

Parameters

<*type*>	file type of which path to be set.
<*newpath*>	new pathname.

Exceptions

thrown	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

<*type*>	type name of which path to be set.
<*newpath*>	new pathname.

Exceptions

thrown	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

<*type*>	type name of which path to be set.
<*newpath*>	new pathname.

Exceptions

thrown	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 = "" )
```

Reads GMAT startup file.

Parameters

<fileName>	startup file name.

Exceptions

UtilityException	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 IsGUISavable ()
- 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 ∗itrf)
- 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 resouce 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

tf	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

flag	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 resouce flag. Normally this flag is set from the ScriptInterpreter and the ResourceTree retrieves it.

Parameters

flag	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 * itrf )
```

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

tf	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 mdj) 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]
```

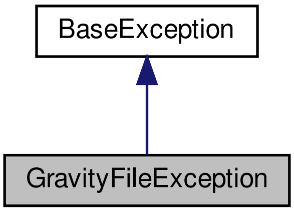
The documentation for this class was generated from the following files:

- GMATsrc/util/GmatTime.hpp
- GMATsrc/util/GmatTime.cpp

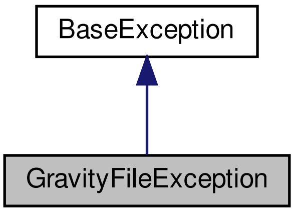
8.27 GravityFileException Class Reference

```
#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 GregorianDate Class Reference

```
#include <GregorianDate.hpp>
```

Classes

- class GregorianDateException

Public Member Functions

- GregorianDate ()
- GregorianDate (const std::string &str)
- GregorianDate (Date ∗newDate, Integer format=1)
- ∼GregorianDate ()
- 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 GregorianDate class base

8.28.2 Constructor & Destructor Documentation

8.28.2.1 GregorianDate() [1/3]

```
GregorianDate::GregorianDate ( )
```

Definition of the Gregorian class base Creates default constructor.

8.28.2.2 GregorianDate() [2/3]

```
GregorianDate::GregorianDate (
             const std::string & str )
```

Creates constructor with parameters.

Parameters

<str>	Given String of Date

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 GregorianDate::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 GregorianDate::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

greg	input gregorian string

Returns

true if time is in valid Gregorian format; otherwise, false

8.28.3.6 SetDate() [1/2]

```
bool GregorianDate::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 GregorianDate::SetDate (
             Date * newDate,
             Integer format = 1 )
```

Set the new date in Date

Parameters

format	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 GregorianDate::SetType (
            const std::string & str )
```

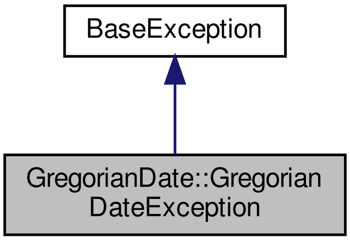
The documentation for this class was generated from the following files:

- GMATsrc/util/GregorianDate.hpp
- GMATsrc/util/GregorianDate.cpp

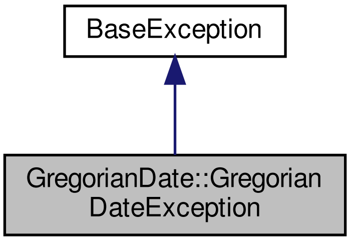
8.29 GregorianDate::GregorianDateException Class Reference

```
#include <GregorianDate.hpp>
```

Inheritance diagram for GregorianDate::GregorianDateException:



Collaboration diagram for GregorianDate::GregorianDateException:



Public Member Functions

- GregorianDateException (const std::string &message="GregorianDateException: Invalid date format")

Additional Inherited Members

8.29.1 Constructor & Destructor Documentation

8.29.1.1 GregorianDateException()

```
GregorianDate::GregorianDateException::GregorianDateException (
            const std::string & message = "GregorianDateException:  Invalid date format" )
[inline]
```

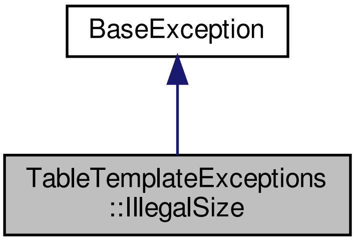
The documentation for this class was generated from the following file:

- GMATsrc/util/GregorianDate.hpp

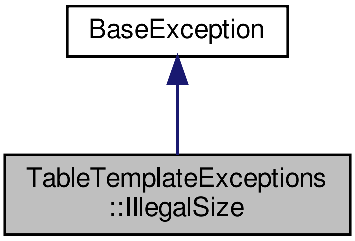
8.30 TableTemplateExceptions::IllegalSize Class Reference

```
#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.\)

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]
```

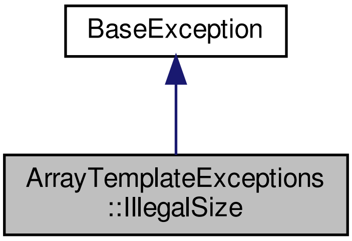
The documentation for this class was generated from the following file:

- GMATsrc/util/TableTemplate.hpp

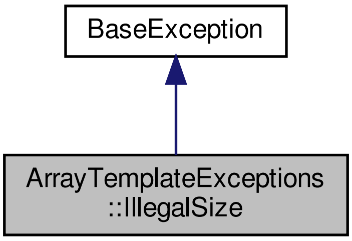
8.31 ArrayTemplateExceptions::IllegalSize Class Reference

```
#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]
```

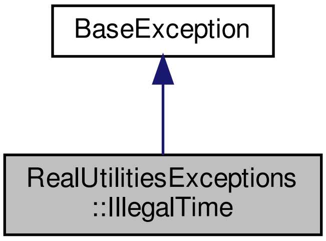
The documentation for this class was generated from the following file:

- GMATsrc/util/ArrayTemplate.hpp

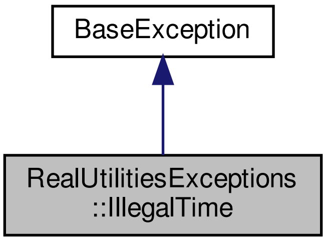
8.32 RealUtilitiesExceptions::IllegalTime Class Reference

```
#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]
```

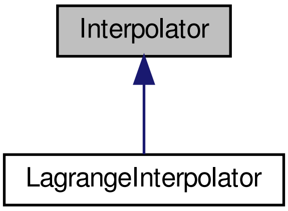
The documentation for this class was generated from the following file:

- 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 feasiblity 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

<*name*>	Name for this interpolator.
<*typestr*>	Text string identifying the type of interpolator.
<*dim*>	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

<*i*>	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

<*ind*>	Value of the independent (domain) variable.
<*data*>	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	The Interpolator 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

lower	[output] lower end of independent variable range
upper	[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, resulint in an array of interpolated data valid at the desired value of the independent variable.

Parameters

<*ind*>	Value of the independent variable at which the data is interpolated.
<*results*>	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

ind	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>	Interpolator 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 feasiblity 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.

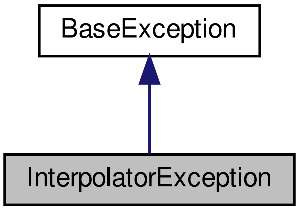
The documentation for this class was generated from the following files:

- GMATsrc/util/interpolator/Interpolator.hpp
- GMATsrc/util/interpolator/Interpolator.cpp

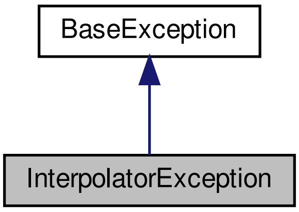
8.34 InterpolatorException Class Reference

```
#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 ( )
```

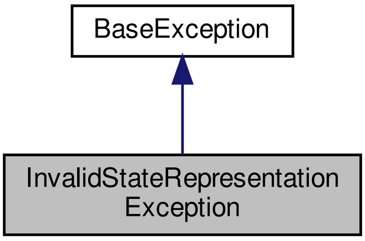
The documentation for this class was generated from the following files:

- GMATsrc/util/interpolator/InterpolatorException.hpp
- GMATsrc/util/interpolator/InterpolatorException.cpp

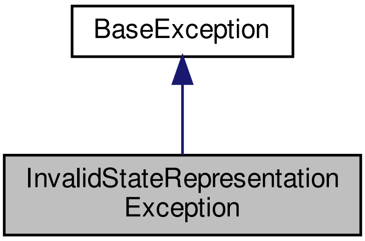
8.35 InvalidStateRepresentationException Class Reference

```
#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 representatic
: " )  [inline]
```

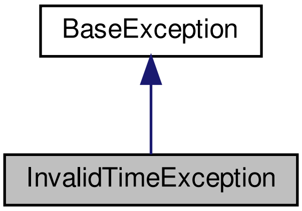
The documentation for this class was generated from the following file:

- GMATsrc/util/BodyFixedStateConverter.hpp

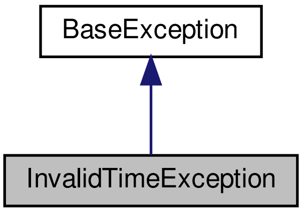
8.36 InvalidTimeException Class Reference

```
#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]
```

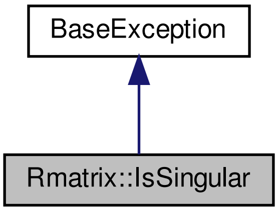
The documentation for this class was generated from the following file:

- GMATsrc/util/TimeSystemConverter.hpp

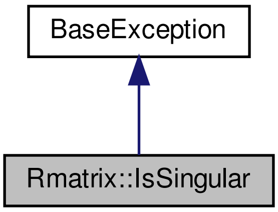
8.37 Rmatrix::IsSingular Class Reference

```
#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]
```

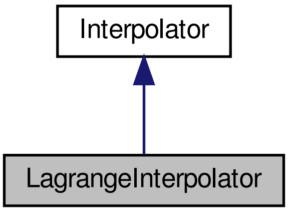
The documentation for this class was generated from the following file:

- GMATsrc/util/Rmatrix.hpp

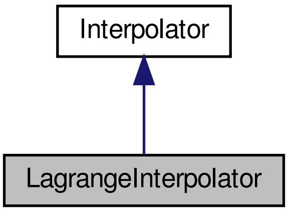
8.38 LagrangeInterpolator Class Reference

```
#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

li	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

thrown	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	The Interpolator 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

ind	The value of the independent parameter.
results	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

ind	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

li	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 coefficent 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 coefficent 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 )
```

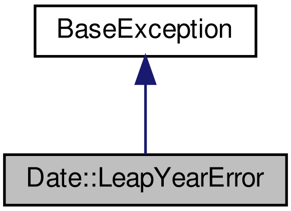
The documentation for this class was generated from the following files:

- GMATsrc/util/LeapSecsFileReader.hpp
- GMATsrc/util/LeapSecsFileReader.cpp

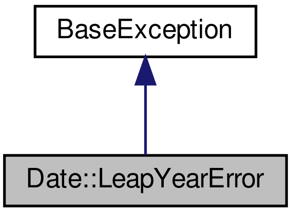
8.41 Date::LeapYearError Class Reference

```
#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]
```

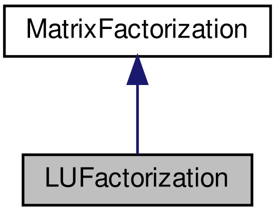
The documentation for this class was generated from the following file:

- GMATsrc/util/Date.hpp

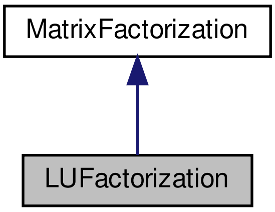
8.42 LUFactorization Class Reference

```
#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

pivotOption	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

A	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

inputMatrix	The matrix that will be factored
&L	The lower triangular matrix
&U	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

&inputMatrix	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 algorthims 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

inputMatrix	The matrix resembling coefficients of equations
b	Column vector representing RHS values of equations from inputMatrix
x	Column vector containing solution values to solve system of equations

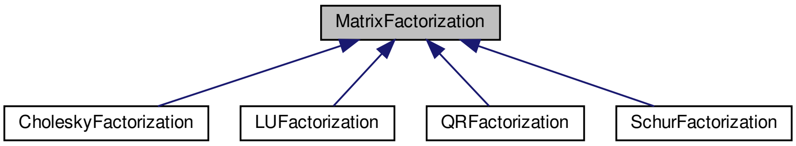
The documentation for this class was generated from the following files:

- 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

infMatrix	The square symmetric matrix for which rows/columns of zeros are to be removed
removedIndexes	The indexes of the rows/columns that were removed
auxVector	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
numRemoved	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

covMatrix	The N-numRemoved x N-numRemoved inverse of the normal matrix
auxVector	The auxiliary vector which was returned by CompressNormalMatrix() when the normal matrix was reduced
numRemoved	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 <= row < N, row <= 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

N	The number of rows and columns in the square symmetric matrix
row	The row index of the element. 0 <= row < N.
col	The column index of the element. row <= 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 rerived from teh 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

msg	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

format	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

msgType	The type of message that is displayed, selected from the set {ERROR_, WARNING_, INFO_} enumerated in the Gmat namespace.
msg	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

msgType	The type of message that is displayed, selected from the set {ERROR_, WARNING_, INFO_} enumerated in the Gmat namespace.
format	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

flag	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

msgString	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

format	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]
```

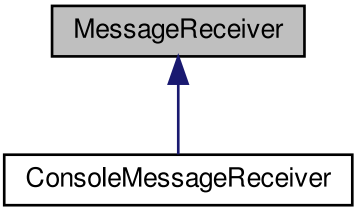
The documentation for this class was generated from the following files:

- 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.

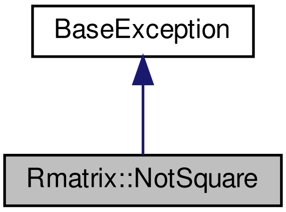
The documentation for this class was generated from the following files:

- GMATsrc/util/MessageReceiver.hpp
- GMATsrc/util/MessageReceiver.cpp

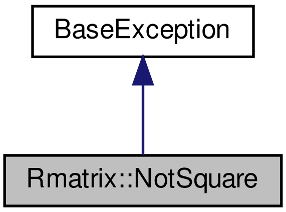
8.46 Rmatrix::NotSquare Class Reference

```
#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]
```

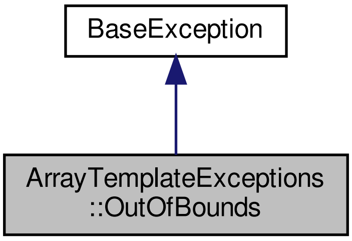
The documentation for this class was generated from the following file:

- GMATsrc/util/Rmatrix.hpp

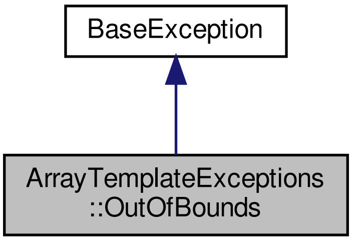
8.47 ArrayTemplateExceptions::OutOfBounds Class Reference

```
#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]
```

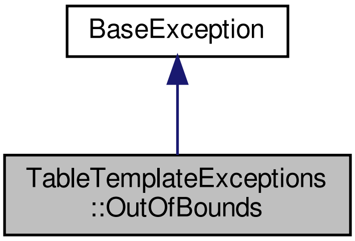
The documentation for this class was generated from the following file:

- GMATsrc/util/ArrayTemplate.hpp

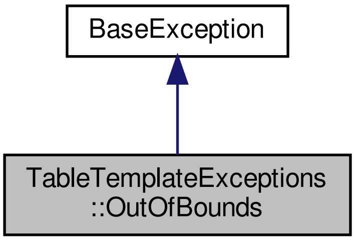
8.48 TableTemplateExceptions::OutOfBounds Class Reference

```
#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.\)

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
```

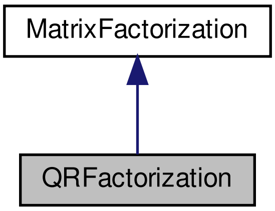
The documentation for this struct was generated from the following file:

- GMATsrc/include/gmatdefs.hpp

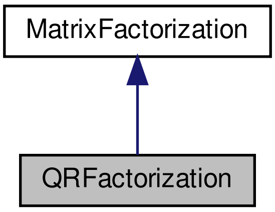
8.50 QRFactorization Class Reference

```
#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

R	Original upper triangular matrix from initial QR factorization
Q	Original orthogonal matrix from initial QR factorization
dimensionToInsert	String entry determining whether a row (entered as "row") or column (entered as "col") should be added
locationToInsert	Index of which row/column from the original factored matrix should be added
&R1	New upper triangular matrix after removing row/column from A
&Q1	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

A	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 (inv(A) = inv(Q)∗inv(R) = inv(R)∗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

R	Original upper triangular matrix from initial QR factorization
Q	Original orthogonal matrix from initial QR factorization
dimensionToRemove	String entry determining whether a row (entered as "row") or column (entered as "col") should be removed
locationToRemove	Index of which row/column from the original factored matrix should be removed
&R1	New upper triangular matrix after removing row/column from A
&Q1	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() [1/2]

```
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 constatns and Math Utilities that are not provided in the C++ Library or provides call-throughs to the routines of the C++ (C) math.h

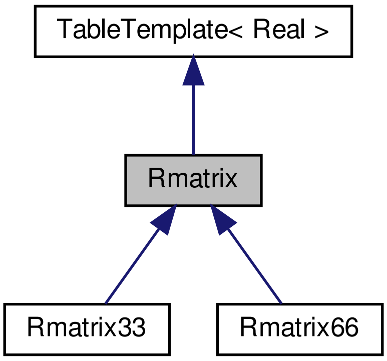
The documentation for this struct was generated from the following file:

- GMATsrc/util/RealUtilities.hpp

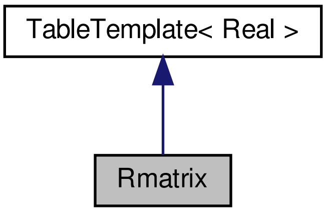
8.55 Rmatrix Class Reference

```
#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::D←ATA_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]
```

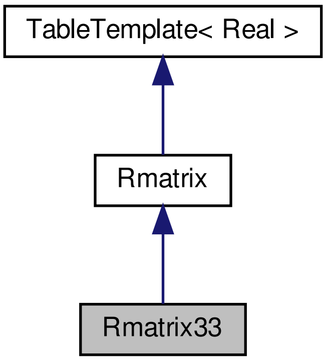
The documentation for this class was generated from the following files:

- GMATsrc/util/Rmatrix.hpp
- GMATsrc/util/Rmatrix.cpp

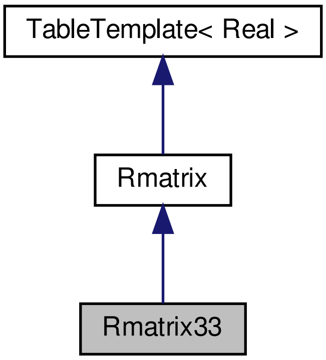
8.56 Rmatrix33 Class Reference

```
#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]
```

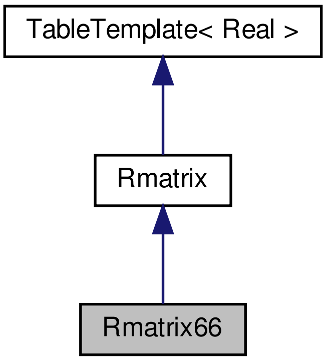
The documentation for this class was generated from the following files:

- GMATsrc/util/Rmatrix33.hpp
- GMATsrc/util/Rmatrix33.cpp

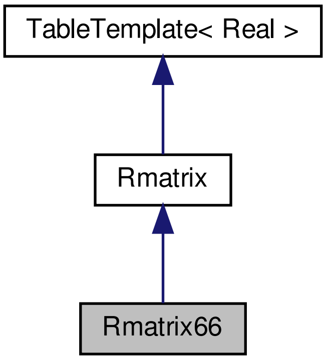
8.57 Rmatrix66 Class Reference

```
#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]
```

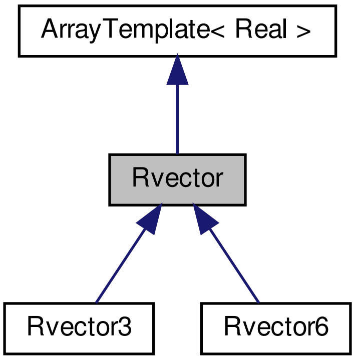
The documentation for this class was generated from the following files:

- GMATsrc/util/Rmatrix66.hpp
- GMATsrc/util/Rmatrix66.cpp

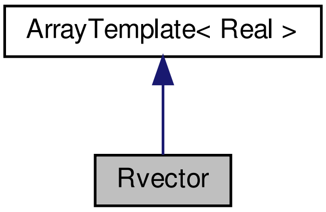
8.58 Rvector Class Reference

```
#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

ascending	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

indices	array containing original indices after sort is done
ascending	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]
```

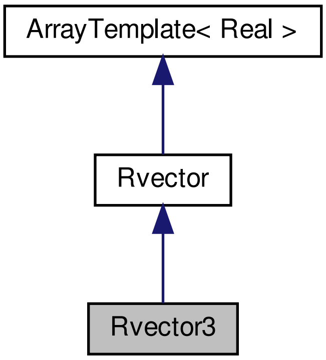
The documentation for this class was generated from the following files:

- GMATsrc/util/Rvector.hpp
- GMATsrc/util/Rvector.cpp

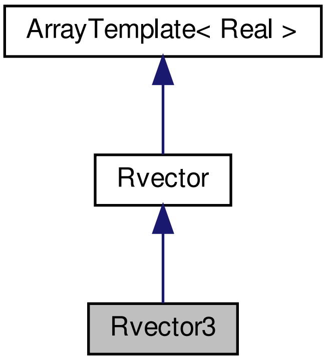
8.59 Rvector3 Class Reference

```
#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() [1/2]

```
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]
```

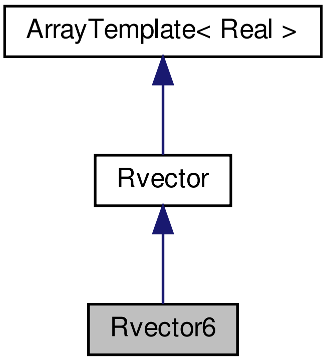
The documentation for this class was generated from the following files:

- GMATsrc/util/Rvector3.hpp
- GMATsrc/util/Rvector3.cpp

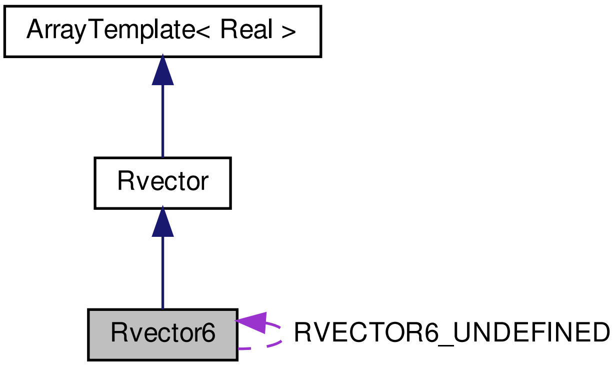
8.60 Rvector6 Class Reference

```
#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 elelmets 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

thrown	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

thrown	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

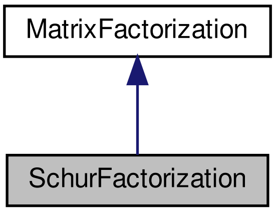
The documentation for this class was generated from the following files:

- GMATsrc/util/Rvector6.hpp
- GMATsrc/util/Rvector6.cpp

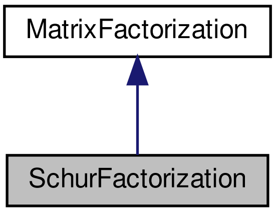
8.61 SchurFactorization Class Reference

```
#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->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

inputMatrix	The matrix to factor
&sdUnitary	Unitary matrix Q where Q'AQ = U
&sdUpper	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

inputMatrix	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

sum1	The matrix to be inverted, packed in upper triangular form
array_size	The size of the sum1 array

8.61.3.4 operator=()

```
SchurFactorization & SchurFactorization::operator= (
            const SchurFactorization & schurfactorization )
```

Assigment 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

sum1	The upper triangle of a square symmetric matrix packed into an array
num_rows	The number of rows and columns in the matrix
row_to_remove	The index of the row/column to remove from the matrix. 0 <= row_to_remove < num_rows
removed_rows	An array containing the indexes of the rows/columns which were removed from the matrix. row_to_remove 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

sum1	The upper triangle of a square symmetric matrix packed into an array
num_rows	The number of rows and columns in the matrix
removed_rows	An array containing the indexes of the rows/columns which were removed from the matrix. row_to_remove 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 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 Rvector6 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 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 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

e	eccentricity
M	mean anomaly return value of ecentric anomaly

8.62.3.7 CartesianToAngularMomentum()

```
Rvector6 StateConversionUtil::CartesianToAngularMomentum (
            Real mu,
            const Rvector3 & pos,
            const Rvector3 & vel )  [static]
```

CartesianTos 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

mu	mu value of primary body
cartesianState	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

mu	mu value of primary body
cartesianState	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

mu	mu value of primary body
keplerState	state presented in Keplerian coordiante system

return derivative state conversion matrix

8.62.3.27 CartesianToKeplerianDerivativeConversionWithKeplInput_FiniteDiff()

```
Rmatrix66 StateConversionUtil::CartesianToKeplerianDerivativeConversionWithKeplInput_Finite↩
Diff (
             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

mu	mu value of primary body
keplerianState	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[GmatSolar←↩
SystemDefaults::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

<*equinoctial*>	Equinoctial state
<*mu*>	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

<*equinoctial*>	Equinoctial state
<*mu*>	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

<*anomalyType*>	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

<r>	Position
<v>	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

<typeStr>	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

<mu>	Gravitational constant for the central body
<state>	Keplerian state
<anomalyType>	Anomaly Type

Returns

Spacecraft orbit state converted from Keplerian to Cartesian

8.62.3.72 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

<keplerian>	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

<modequinoctial>	Modified Equinoctial state
<mu>	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

<modKeplerian>	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

<mu>	Gravitational constant for the central body
<outasym>	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

<planetodetic>	Planetodetic state
<flattening>	flattening coefficient for the central body
<eqRadius>	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

<spherical>	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

taRadians	True anomaly in radians
ecc	Eccentricity
mod↩ By2Pi	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

UtilityException	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 specifiying 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]
```

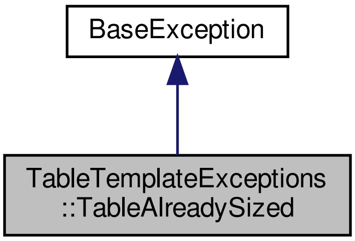
The documentation for this class was generated from the following files:

- GMATsrc/util/StringTokenizer.hpp
- GMATsrc/util/StringTokenizer.cpp

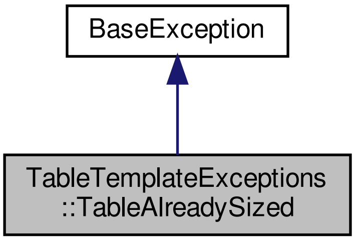
8.64 TableTemplateExceptions::TableAlreadySized Class Reference

```
#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.\)

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]
```

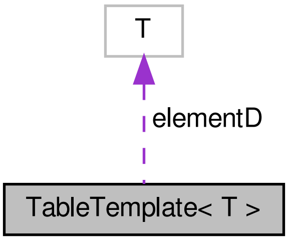
The documentation for this class was generated from the following file:

- 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 UnsizedTable

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)

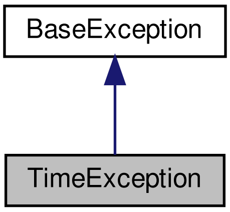
The documentation for this class was generated from the following file:

- GMATsrc/util/TableTemplate.hpp

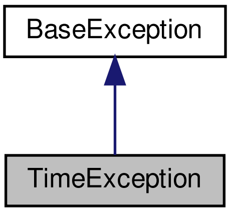
8.67 TimeException Class Reference

```
#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]
```

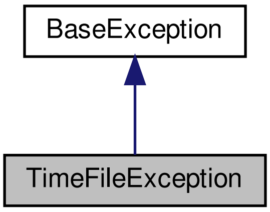
The documentation for this class was generated from the following file:

- GMATsrc/util/UtilityException.hpp

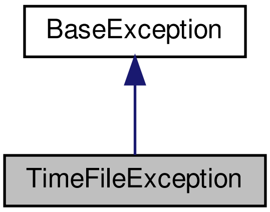
8.68 TimeFileException Class Reference

```
#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]
```

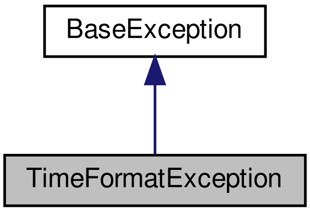
The documentation for this class was generated from the following file:

- GMATsrc/util/TimeSystemConverter.hpp

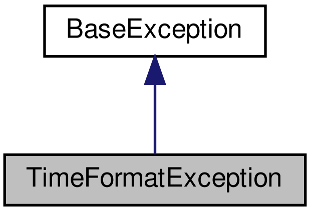
8.69 TimeFormatException Class Reference

`#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]
```

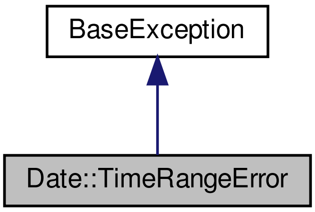
The documentation for this class was generated from the following file:

- GMATsrc/util/TimeSystemConverter.hpp

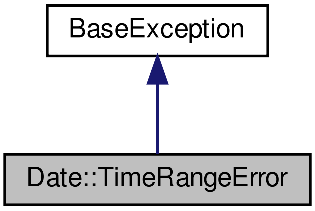
8.70 Date::TimeRangeError Class Reference

```
#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]
```

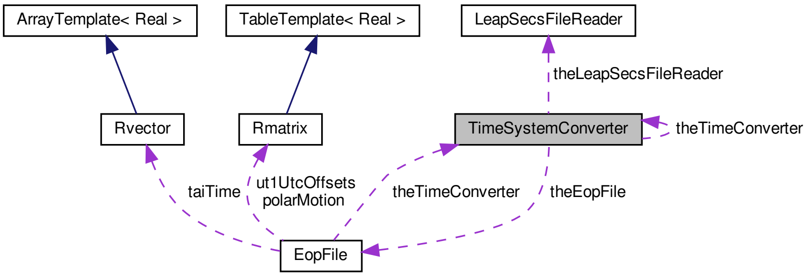
The documentation for this class was generated from the following file:

- 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 Date

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 Date

Exceptions

<*TimeFileException*>	thrown if LeapSecsFileReader is not set
<*TimeFileException*>	thrown if EopFile 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

<mjd>	Input time in MJD
<handleLeapSecond>	Do we need to handle a leap second (UTC only)?
<format>	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

<fromType>	Time type to convert from
<origValue>	Original time value to convert
<refJd>	Reference Julian Date

Exceptions

<TimeFileException>	thrown if LeapSecsFileReader is not set
<TimeFileException>	thrown if EopFile 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

system	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

utcMjd	Modified julian days in UTC
jdOfMjdRef	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

<eopFile>	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",
   }
```

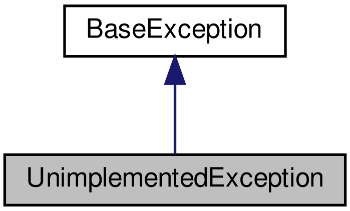
The documentation for this class was generated from the following files:

- GMATsrc/util/TimeSystemConverter.hpp
- GMATsrc/util/TimeSystemConverter.cpp

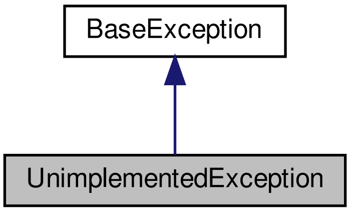
8.72 UnimplementedException Class Reference

```
#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]
```

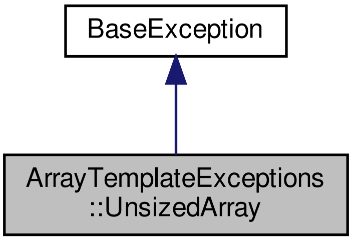
The documentation for this class was generated from the following file:

- GMATsrc/util/TimeSystemConverter.hpp

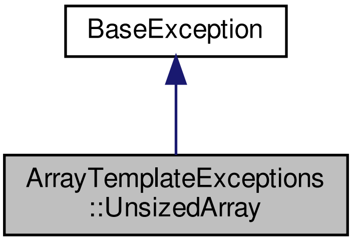
8.73 ArrayTemplateExceptions::UnsizedArray Class Reference

```
#include <ArrayTemplate.hpp>
```

Inheritance diagram for ArrayTemplateExceptions::UnsizedArray:



Collaboration diagram for ArrayTemplateExceptions::UnsizedArray:



Public Member Functions

- UnsizedArray (const std::string &message="ArrayTemplate error : unsized array.")

Additional Inherited Members

8.73.1 Constructor & Destructor Documentation

8.73.1.1 UnsizedArray()

```
ArrayTemplateExceptions::UnsizedArray::UnsizedArray (
            const std::string & message = "ArrayTemplate error : unsized array." )  [inline]
```

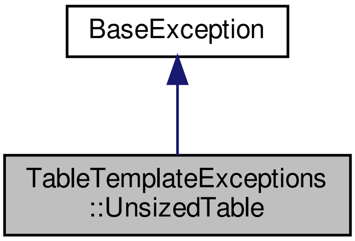
The documentation for this class was generated from the following file:

- GMATsrc/util/ArrayTemplate.hpp

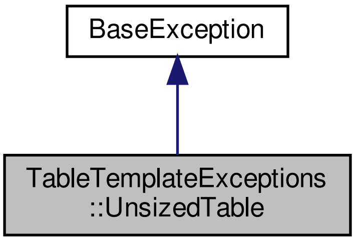
8.74 TableTemplateExceptions::UnsizedTable Class Reference

```
#include <TableTemplate.hpp>
```

Inheritance diagram for TableTemplateExceptions::UnsizedTable:



Collaboration diagram for TableTemplateExceptions::UnsizedTable:



Public Member Functions

- UnsizedTable (const std::string &message="TableTemplate error : unsized table.\)

Additional Inherited Members

8.74.1 Constructor & Destructor Documentation

8.74.1.1 UnsizedTable()

```
TableTemplateExceptions::UnsizedTable::UnsizedTable (
             const std::string & message = "TableTemplate error : unsized table.\n" )  [inline]
```

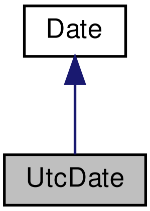
The documentation for this class was generated from the following file:

- GMATsrc/util/TableTemplate.hpp

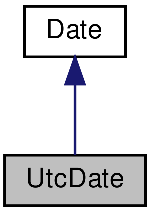
8.75 UtcDate Class Reference

```
#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

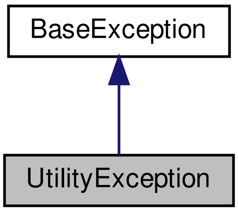
The documentation for this class was generated from the following files:

- GMATsrc/util/UtcDate.hpp
- GMATsrc/util/UtcDate.cpp

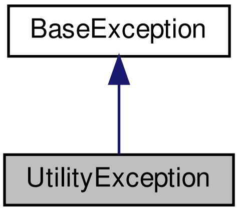
8.76 UtilityException Class Reference

```
#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]
```

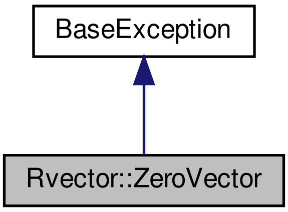
The documentation for this class was generated from the following file:

- GMATsrc/util/UtilityException.hpp

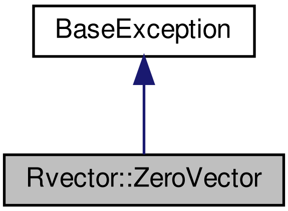
8.77 Rvector::ZeroVector Class Reference

```
#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

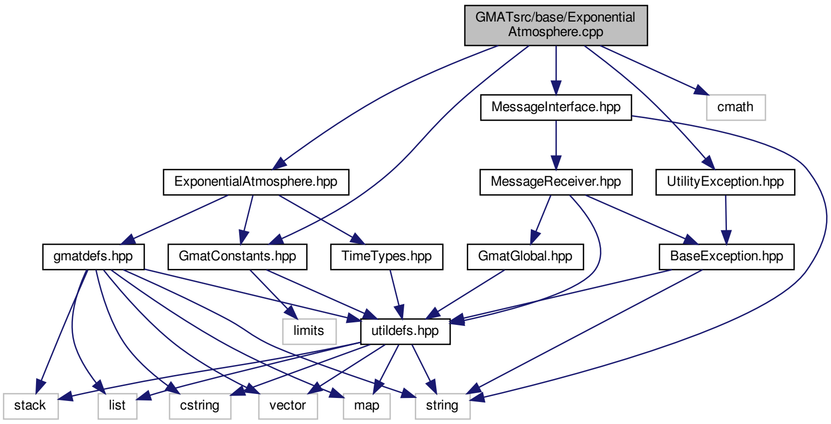
Chapter 9

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:

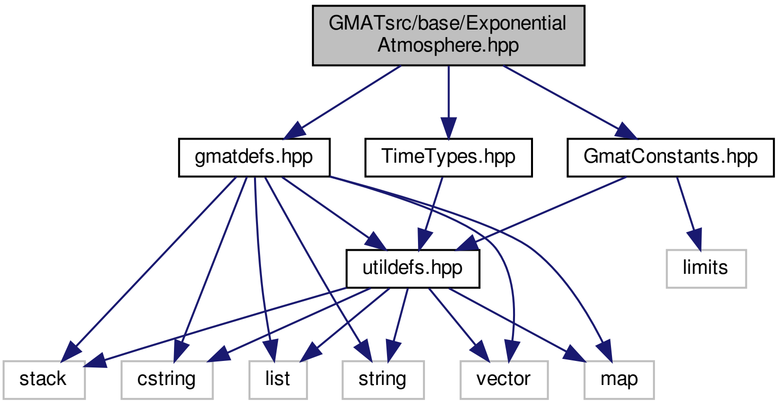


9.2 GMATsrc/base/ExponentialAtmosphere.hpp File Reference

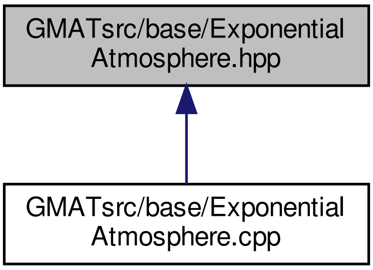
```
#include "gmatdefs.hpp"
#include "TimeTypes.hpp"
```

```
#include "GmatConstants.hpp"
```

Include dependency graph for ExponentialAtmosphere.hpp:



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



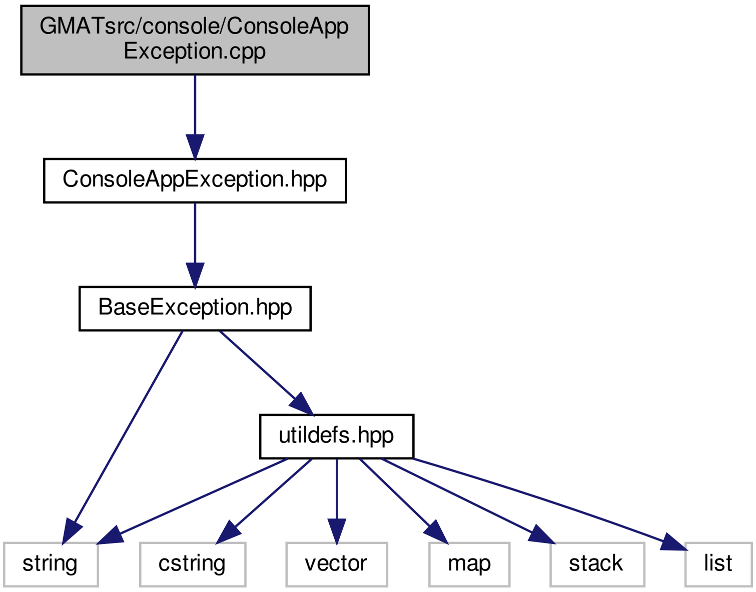
Classes

- class ExponentialAtmosphere

9.3 GMATsrc/console/ConsoleAppException.cpp File Reference

```
#include "ConsoleAppException.hpp"
```

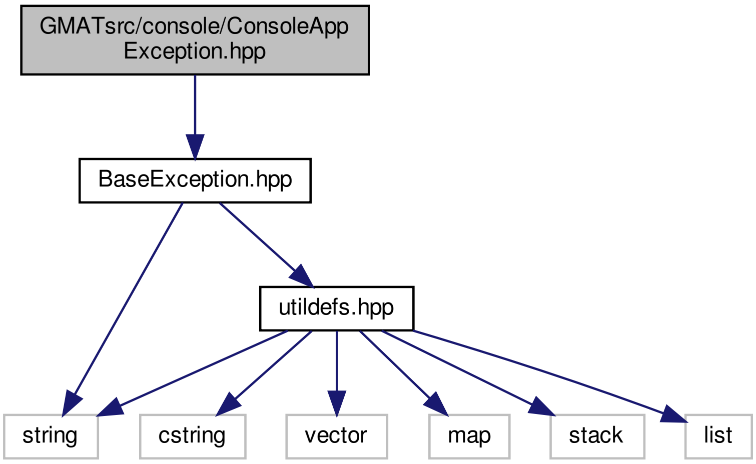
Include dependency graph for ConsoleAppException.cpp:



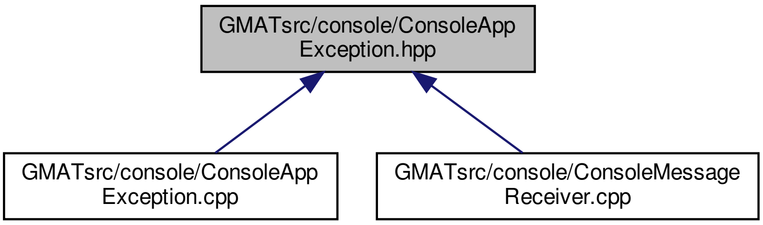
9.4 GMATsrc/console/ConsoleAppException.hpp File Reference

```
#include "BaseException.hpp"
```

Include dependency graph for ConsoleAppException.hpp:



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


GMATsrc/console/ConsoleApp Exception.hpp
GMATsrc/console/ConsoleApp Exception.cpp
GMATsrc/console/ConsoleMessage Receiver.cpp

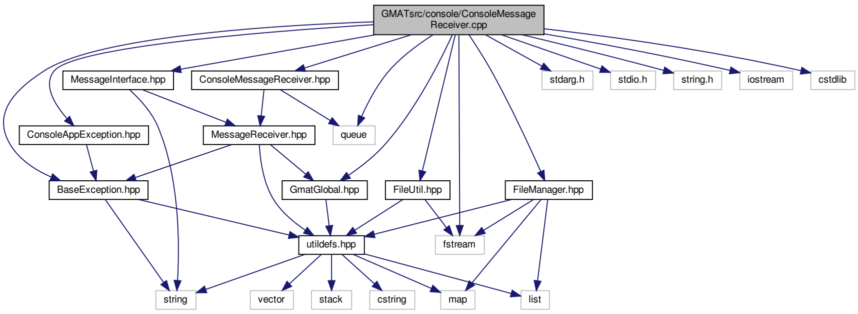
Classes

- 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>
```

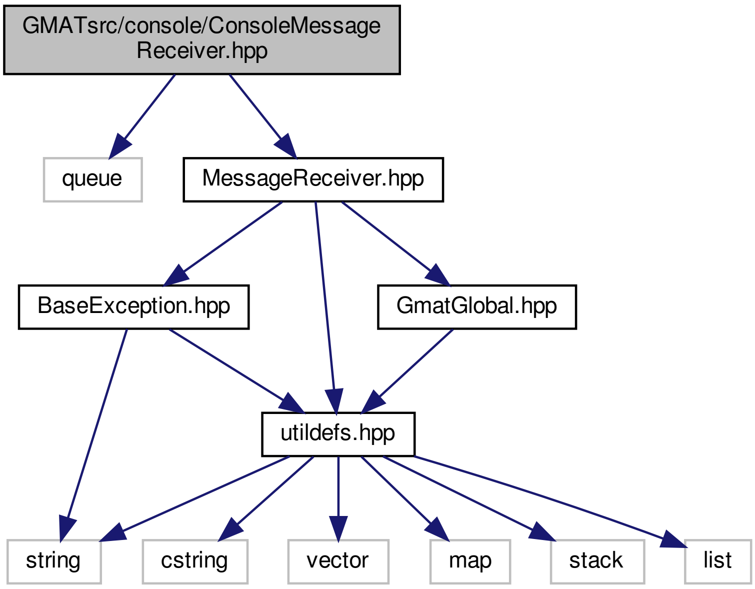
Include dependency graph for ConsoleMessageReceiver.cpp:



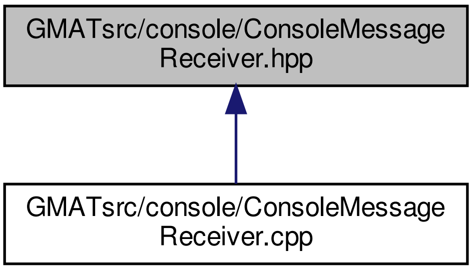
9.6 GMATsrc/console/ConsoleMessageReceiver.hpp File Reference

```
#include <queue>
#include "MessageReceiver.hpp"
```

Include dependency graph for ConsoleMessageReceiver.hpp:



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


GMATsrc/console/ConsoleMessage Receiver.hpp
GMATsrc/console/ConsoleMessage Receiver.cpp

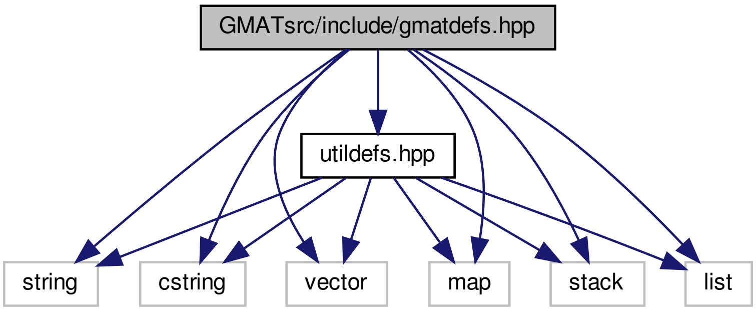
Classes

- class ConsoleMessageReceiver

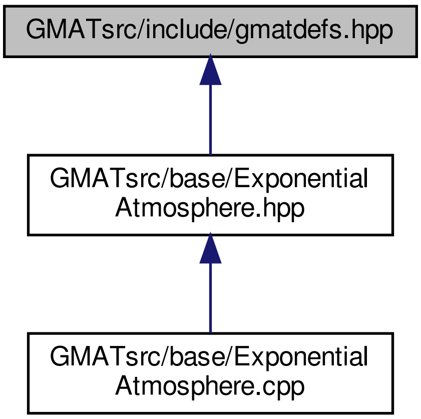
9.7 GMATsrc/include/gmatdefs.hpp File Reference

```
#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:


GMATsrc/include/gmatdefs.hpp
GMATsrc/base/Exponential Atmosphere.hpp
GMATsrc/base/Exponential Atmosphere.cpp

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::OBTYPE, 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<A1Mjd*> 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
```

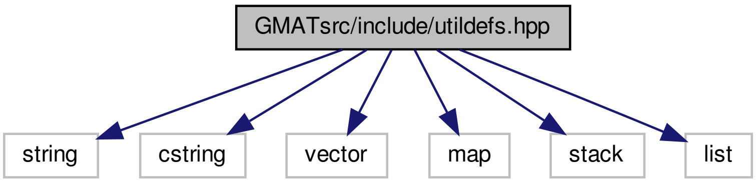
9.7.2.21 WrapperMap

```
typedef std::map<std::string, ElementWrapper*> WrapperMap
```

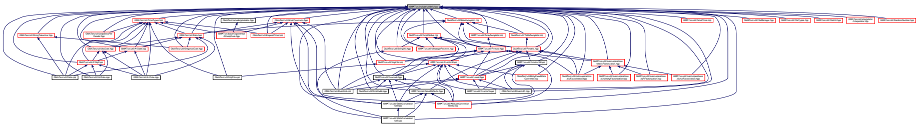
9.8 GMATsrc/include/utildefs.hpp File Reference

```
#include <string>
#include <cstring>
#include <vector>
#include <map>
#include <stack>
#include <list>
```

Include dependency graph for utildefs.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<A1Mjd*> 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
```

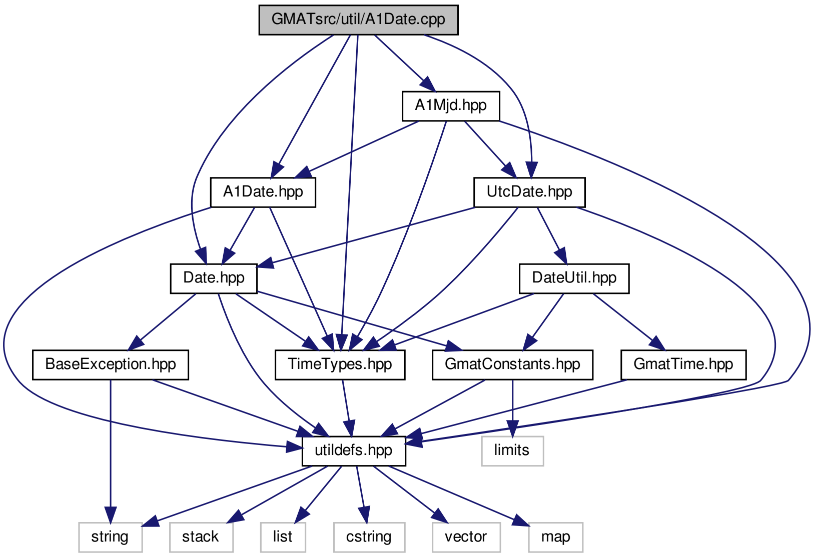
9.8.2.24 WrapperTypeArray

```
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"
```

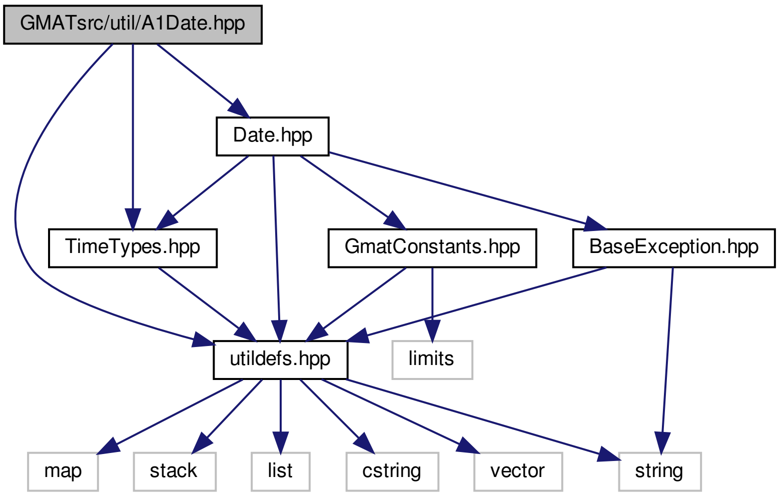
Include dependency graph for A1Date.cpp:



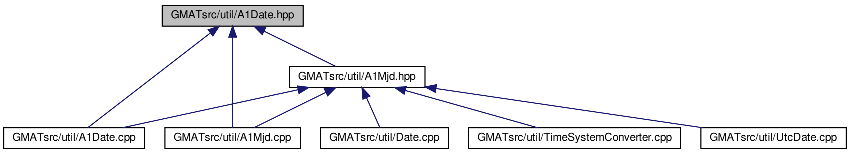
9.10 GMATsrc/util/A1Date.hpp File Reference

```
#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

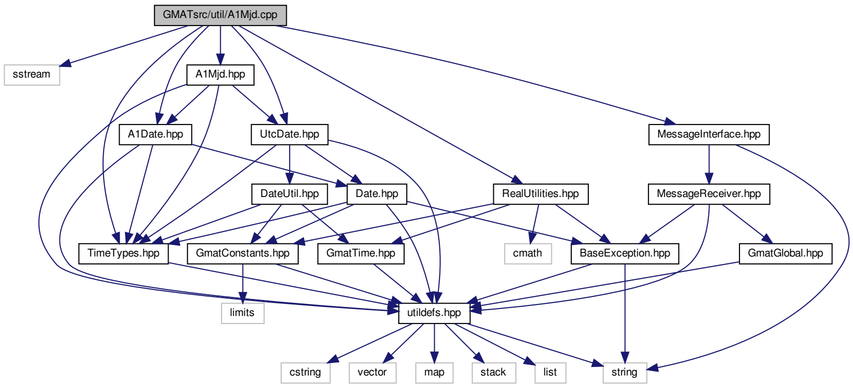
- class A1Date

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"
```

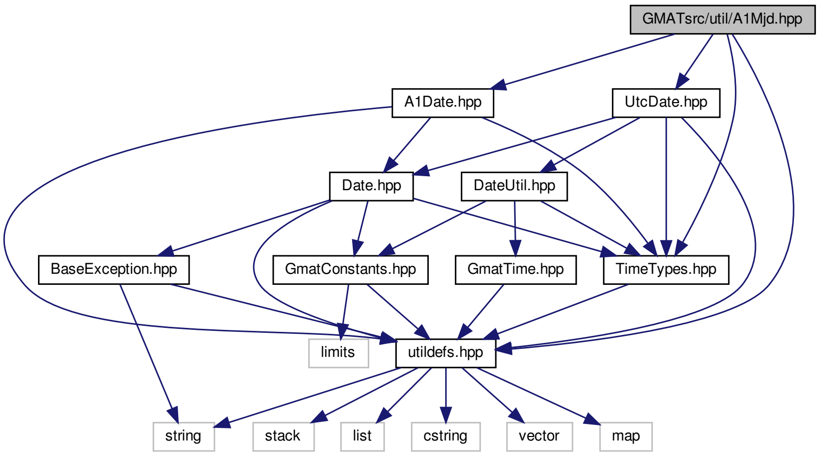
Include dependency graph for A1Mjd.cpp:



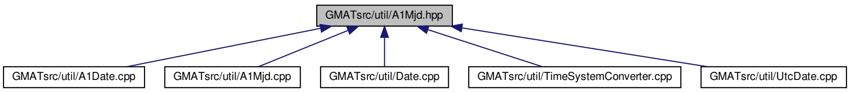
9.12 GMATsrc/util/A1Mjd.hpp File Reference

```
#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:



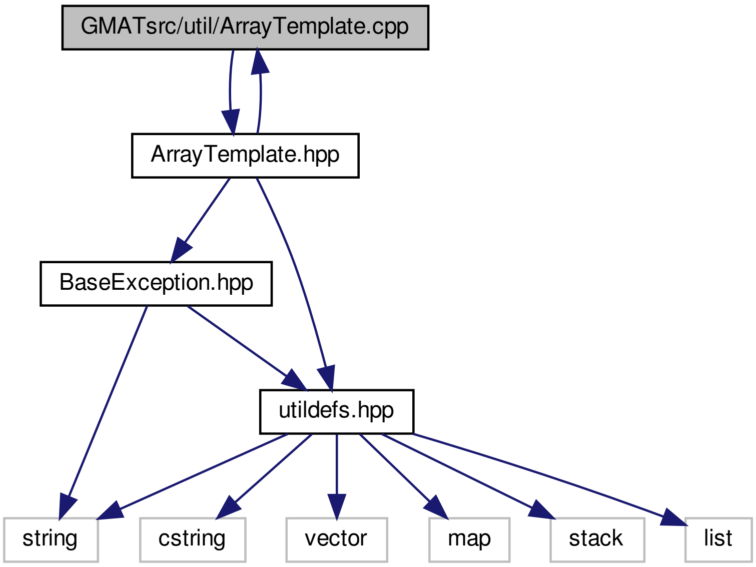
Classes

- class A1Mjd

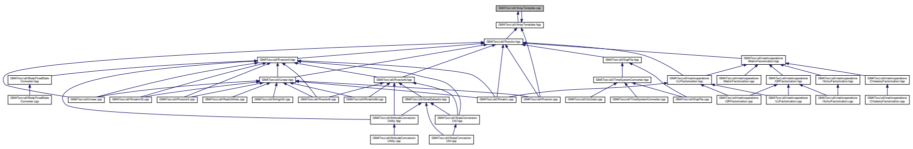
9.13 GMATsrc/util/ArrayTemplate.cpp File Reference

```
#include "ArrayTemplate.hpp"
```

Include dependency graph for ArrayTemplate.cpp:



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

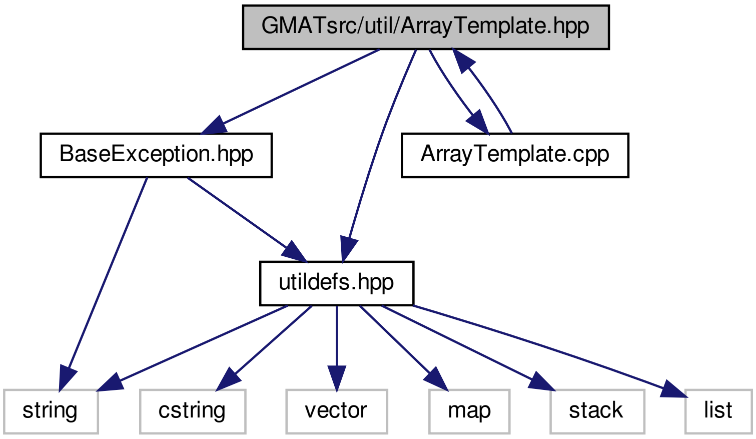


9.14 GMATsrc/util/ArrayTemplate.hpp File Reference

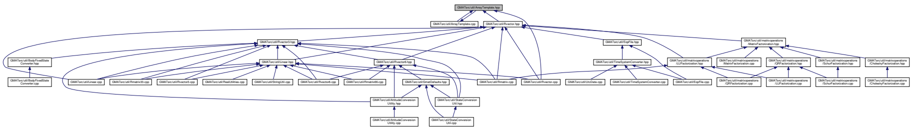
```
#include "utildefs.hpp"
#include "BaseException.hpp"
```

```
#include "ArrayTemplate.cpp"
```

Include dependency graph for ArrayTemplate.hpp:



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



Classes

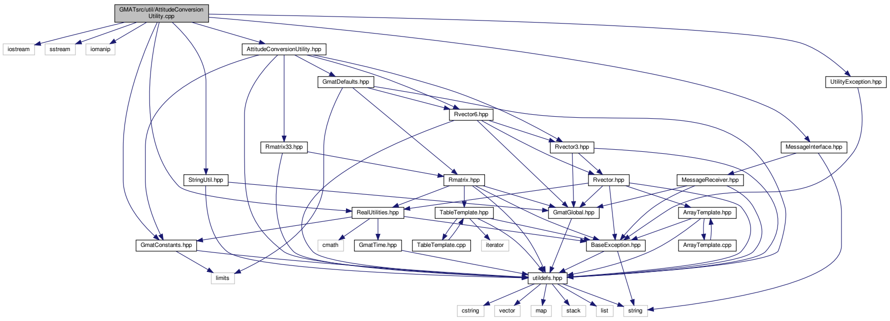
- class ArrayTemplateExceptions
- class ArrayTemplateExceptions::OutOfBounds
- class ArrayTemplateExceptions::DimensionError
- class ArrayTemplateExceptions::UnsizedArray
- class ArrayTemplateExceptions::ArrayAlreadySized
- class ArrayTemplateExceptions::IllegalSize
- class ArrayTemplate< T >

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 "UtilityException.hpp"
```

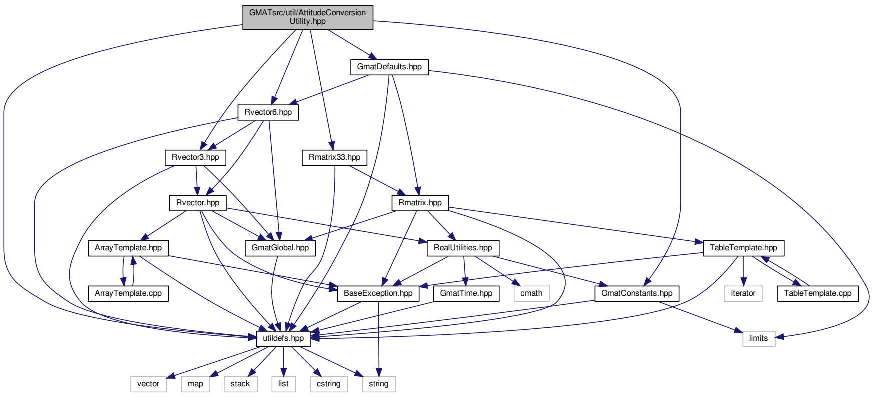
Include dependency graph for AttitudeConversionUtility.cpp:



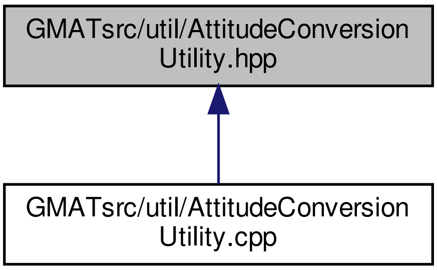
9.16 GMATsrc/util/AttitudeConversionUtility.hpp File Reference

```
#include "utildefs.hpp"
#include "GmatConstants.hpp"
#include "GmatDefaults.hpp"
#include "Rmatrix33.hpp"
#include "Rvector3.hpp"
#include "Rvector6.hpp"
```

Include dependency graph for AttitudeConversionUtility.hpp:



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


GMATsrc/util/AttitudeConversion Utility.hpp
GMATsrc/util/AttitudeConversion Utility.cpp

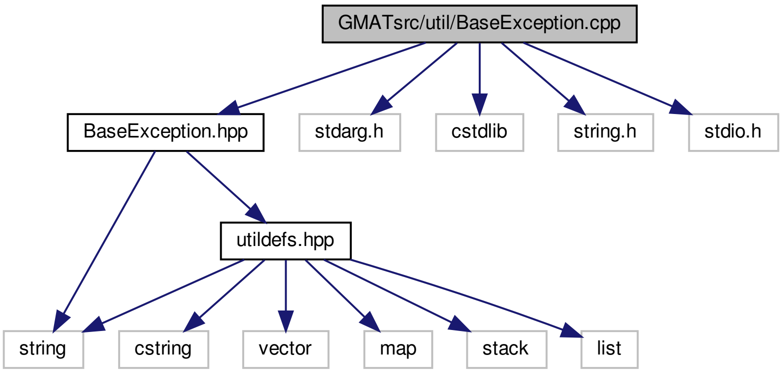
Classes

- class AttitudeConversionUtility

9.17 GMATsrc/util/BaseException.cpp File Reference

```
#include "BaseException.hpp"
#include <stdarg.h>
#include <cstdlib>
#include <string.h>
#include <stdio.h>
```

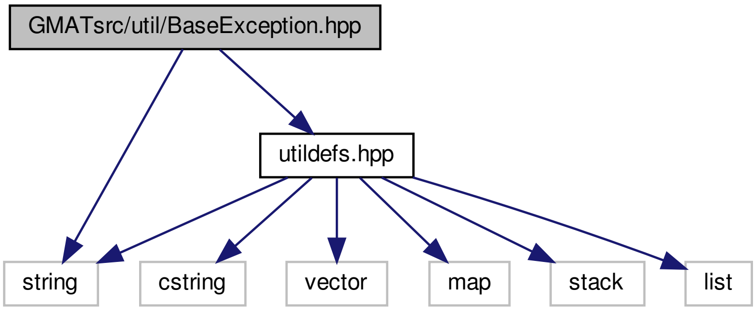
Include dependency graph for BaseException.cpp:



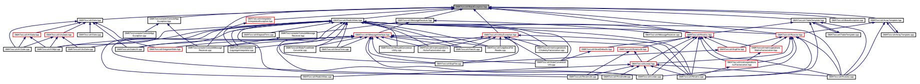
9.18 GMATsrc/util/BaseException.hpp File Reference

```
#include "utildefs.hpp"
#include <string>
```

Include dependency graph for BaseException.hpp:



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



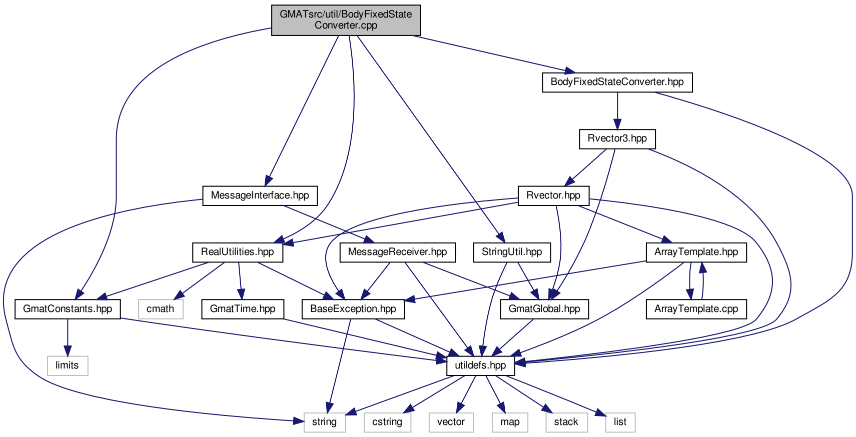
Classes

- 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"
```

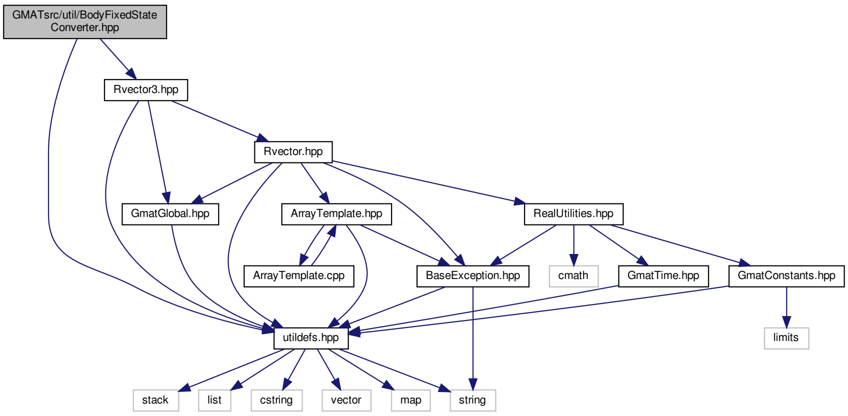
Include dependency graph for BodyFixedStateConverter.cpp:



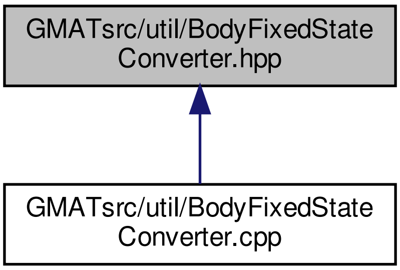
9.20 GMATsrc/util/BodyFixedStateConverter.hpp File Reference

```
#include "utildefs.hpp"
#include "Rvector3.hpp"
```

Include dependency graph for BodyFixedStateConverter.hpp:



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


GMATsrc/util/BodyFixedState Converter.hpp
GMATsrc/util/BodyFixedState Converter.cpp

Classes

- class InvalidStateRepresentationException

Namespaces

- BodyFixedStateConverterUtil

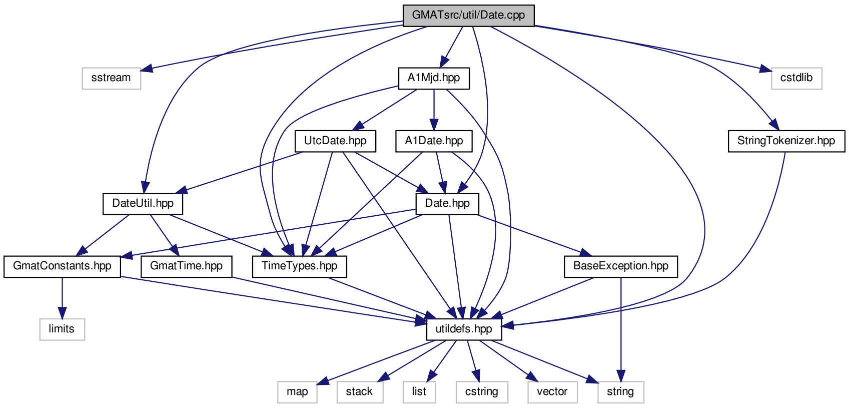
Functions

- 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>
```

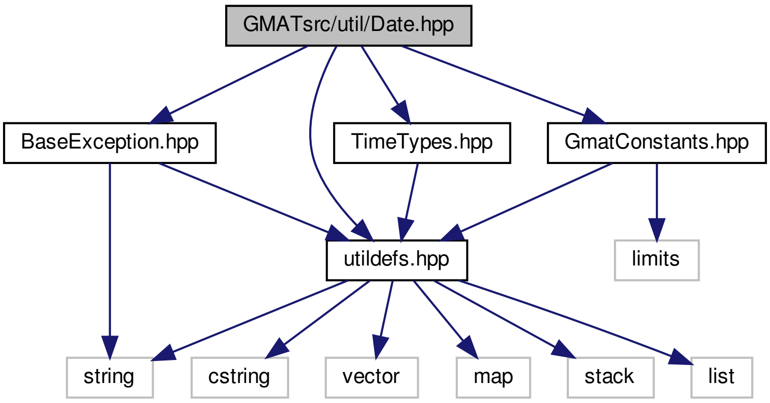
Include dependency graph for Date.cpp:



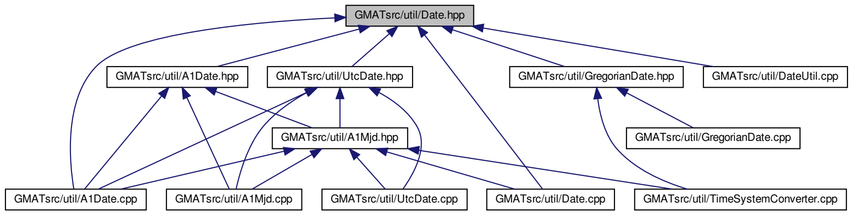
9.22 GMATsrc/util/Date.hpp File Reference

```
#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:



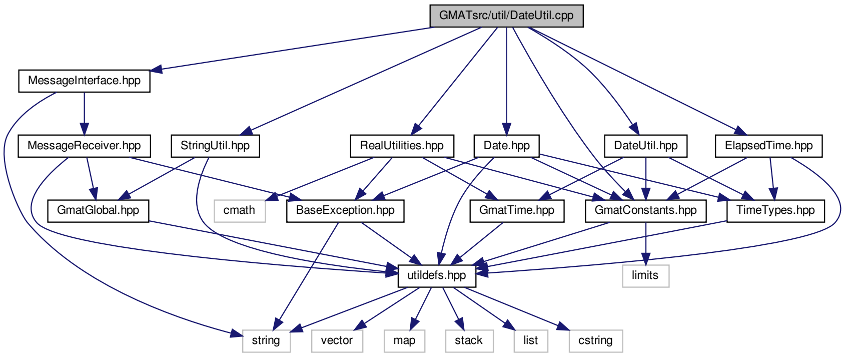
Classes

- 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 Date

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 Date

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.

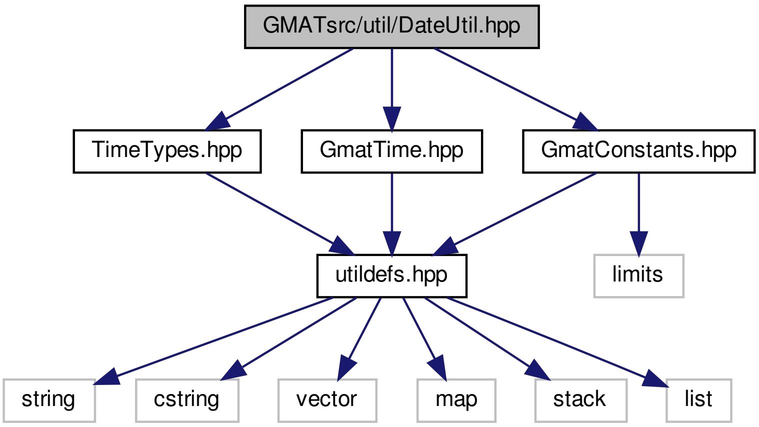
Note

Input time is in hhmmssnnn; added 20 seconds to the input time, to be subtracted later, to avoid gross errors around minute boundaries.

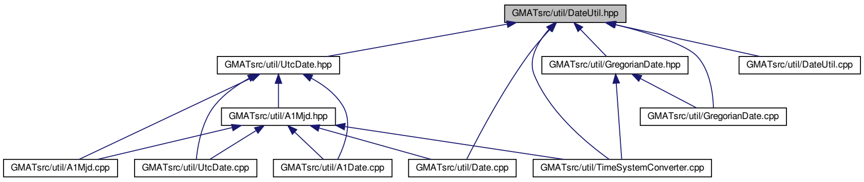
9.24 GMATsrc/util/DateUtil.hpp File Reference

```
#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:


GMATsrc/util/DateUtil.hpp
GMATsrc/util/UtcDate.hpp
GMATsrc/util/GregorianDate.hpp
GMATsrc/util/DateUtil.cpp
GMATsrc/util/A1Mjd.hpp
GMATsrc/util/GregorianDate.cpp
GMATsrc/util/A1Mjd.cpp
GMATsrc/util/UtcDate.cpp
GMATsrc/util/A1Date.cpp
GMATsrc/util/Date.cpp
GMATsrc/util/TimeSystemConverter.cpp

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 Date

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 Date

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.

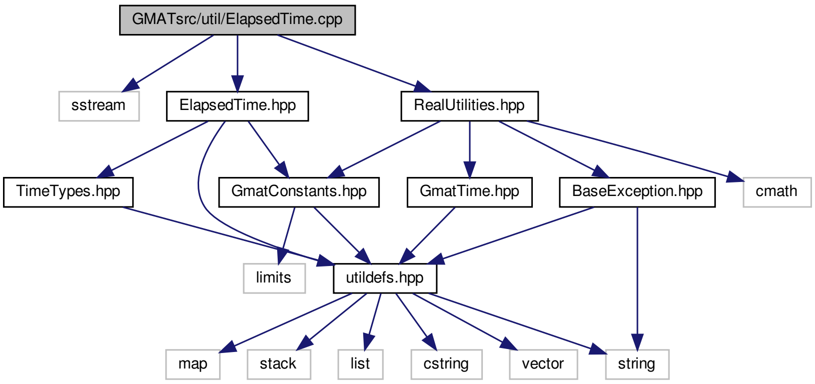
Note

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"
```

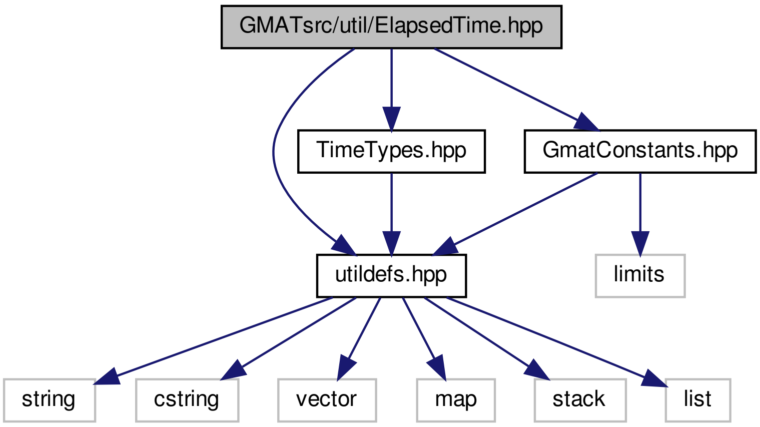
Include dependency graph for ElapsedTime.cpp:



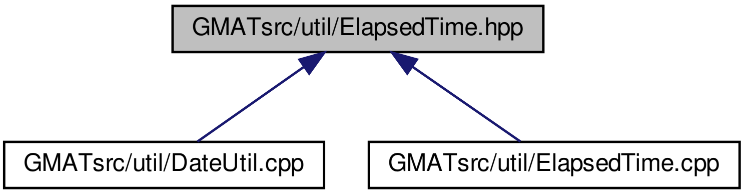
9.26 GMATsrc/util/ElapsedTime.hpp File Reference

```
#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:


GMATsrc/util/ElapsedTime.hpp
GMATsrc/util/DateUtil.cpp
GMATsrc/util/ElapsedTime.cpp

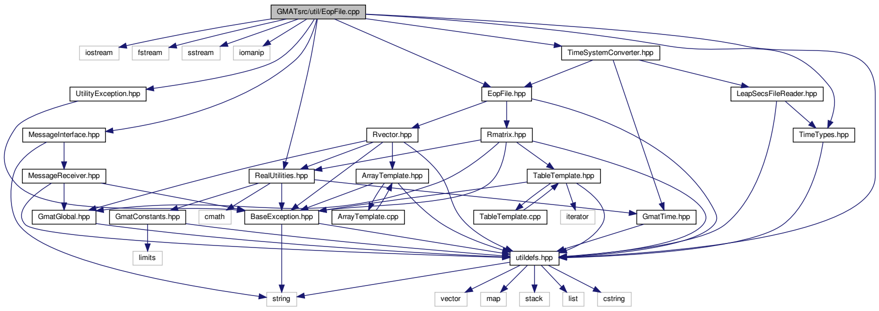
Classes

- 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:

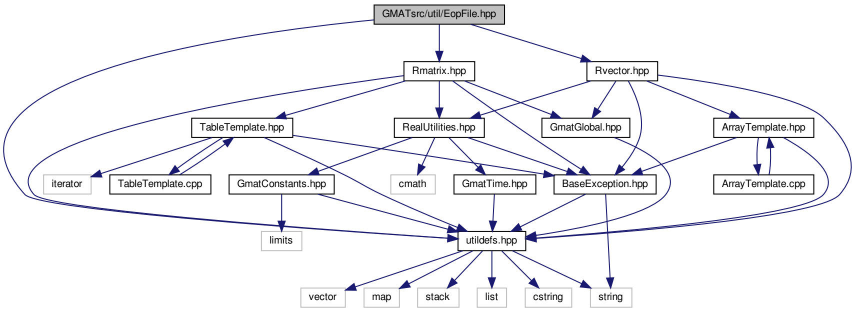


9.28 GMATsrc/util/EopFile.hpp File Reference

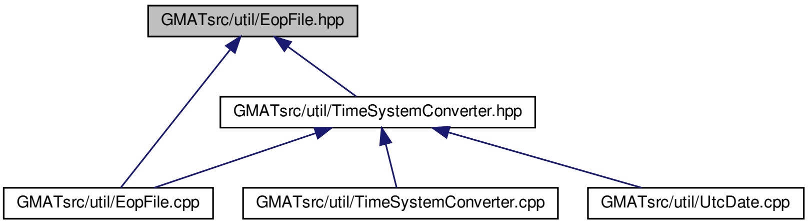
```
#include "utildefs.hpp"
#include "Rmatrix.hpp"
```

```
#include "Rvector.hpp"
```

Include dependency graph for EopFile.hpp:



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


GMATsrc/util/EopFile.hpp
GMATsrc/util/TimeSystemConverter.hpp
GMATsrc/util/EopFile.cpp
GMATsrc/util/TimeSystemConverter.cpp
GMATsrc/util/UtcDate.cpp

Classes

- class EopFile

Namespaces

- GmatEop

Enumerations

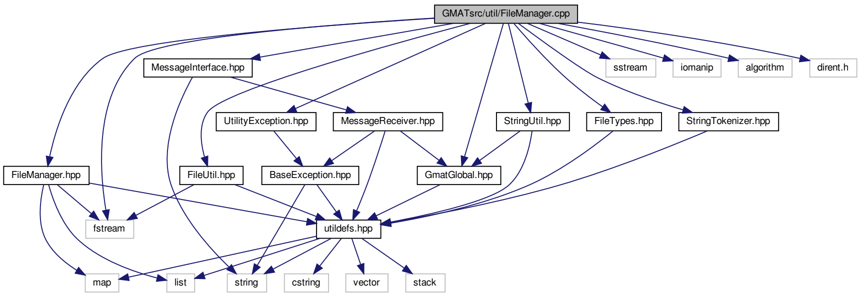
- enum GmatEop::EopFileType { GmatEop::EOP_C04, GmatEop::FINALS }

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>
```

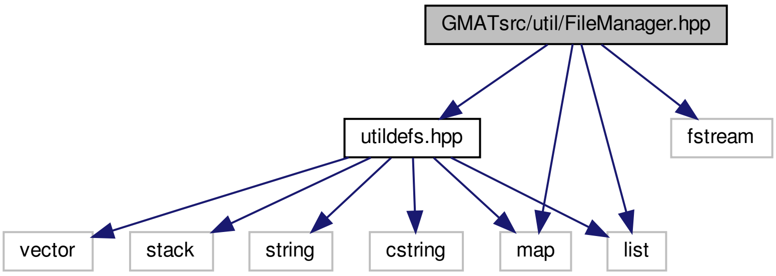
Include dependency graph for FileManager.cpp:



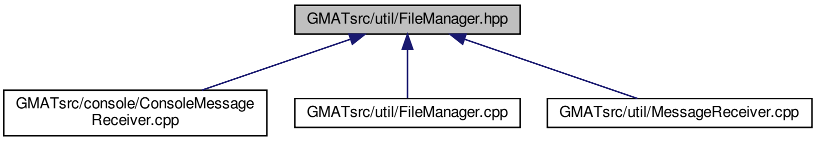
9.30 GMATsrc/util/FileManager.hpp File Reference

```
#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:


GMATsrc/util/FileManager.hpp
GMATsrc/console/ConsoleMessage Receiver.cpp
GMATsrc/util/FileManager.cpp
GMATsrc/util/MessageReceiver.cpp

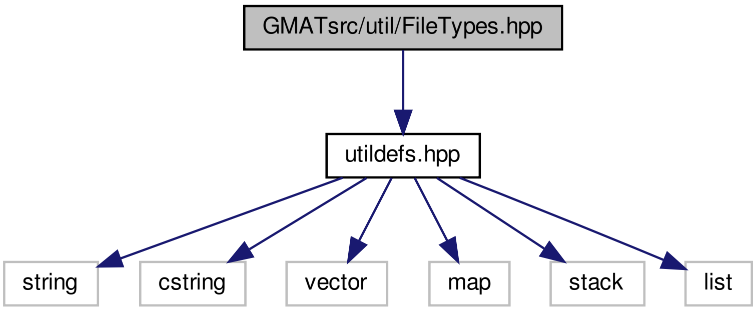
Classes

- class FileManager

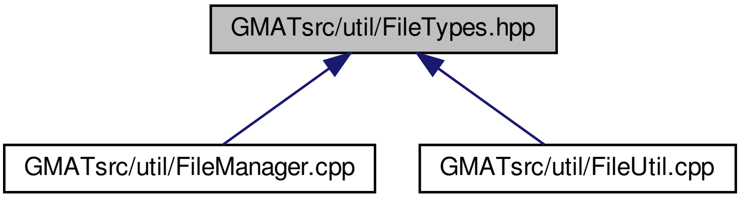
9.31 GMATsrc/util/FileTypes.hpp File Reference

```
#include "utildefs.hpp"
```

Include dependency graph for FileTypes.hpp:



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


GMATsrc/util/FileTypes.hpp
GMATsrc/util/FileManager.cpp
GMATsrc/util/FileUtil.cpp

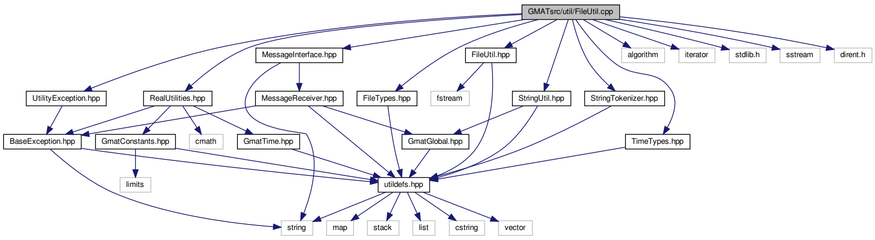
Namespaces

- 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>
```

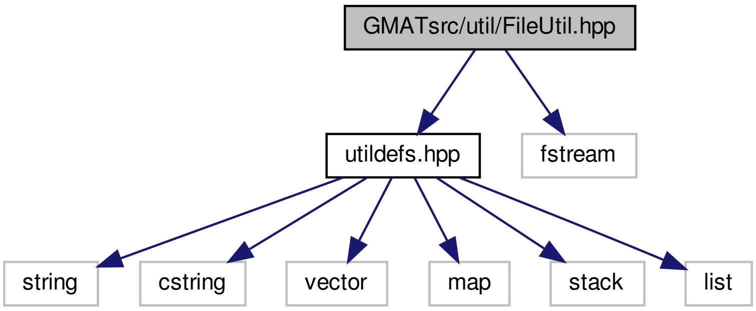
Include dependency graph for FileUtil.cpp:



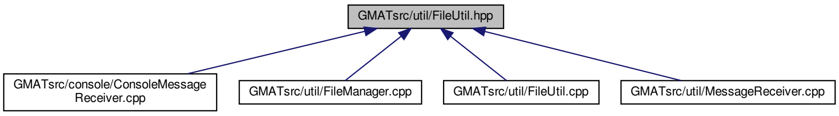
9.33 GMATsrc/util/FileUtil.hpp File Reference

```
#include "utildefs.hpp"
#include <fstream>
```

Include dependency graph for FileUtil.hpp:



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


GMATsrc/util/FileUtil.hpp
GMATsrc/console/ConsoleMessage Receiver.cpp
GMATsrc/util/FileManager.cpp
GMATsrc/util/FileUtil.cpp
GMATsrc/util/MessageReceiver.cpp

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::IsAppInstalled (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=COM↩ PARE_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)

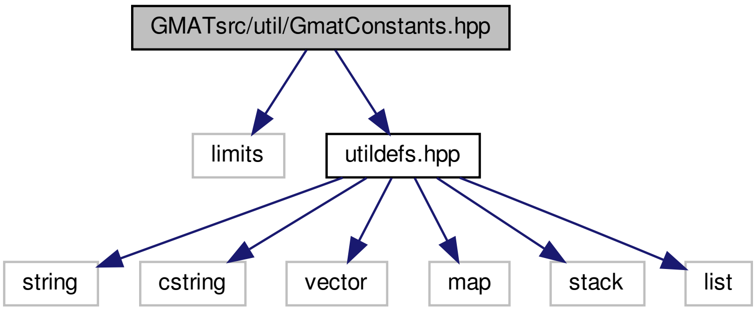
Variables

- const Integer GmatFileUtil::BUFFER_SIZE = 4096

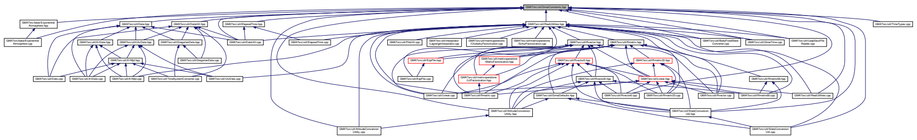
9.34 GMATsrc/util/GmatConstants.hpp File Reference

```
#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

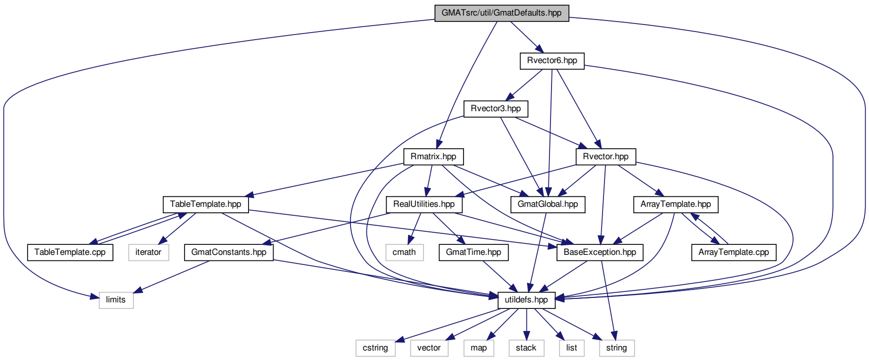
- const Real GmatRealConstants::REAL_TOL = 1.0e-15
- const Real GmatRealConstants::REAL_UNDEFINED = -987654321.0123e-45
- const Real GmatRealConstants::REAL_UNDEFINED_LARGE = -9876543210.1234
- const Integer GmatRealConstants::INTEGER_UNDEFINED = -987654321
- const Integer GmatRealConstants::INTEGER_MAX = 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

- const Real GmatPhysicalConstants::ASTRONOMICAL_UNIT = 1.49597870e8
- const Real GmatPhysicalConstants::ABSOLUTE_ZERO_K = 0.0
- const Real GmatPhysicalConstants::ABSOLUTE_ZERO_C = -273.15
- const Integer GmatIntegerConstants::INTEGER_UNDEFINED = -987654321
- const Real GmatTimeConstants::SECS_PER_DAY = 86400.0
- const Real GmatTimeConstants::SECS_PER_HOUR = 3600.0
- const Real GmatTimeConstants::SECS_PER_MINUTE = 60.0
- const Real GmatTimeConstants::DAYS_PER_YEAR = 365.25
- const Real GmatTimeConstants::DAYS_PER_JULIAN_CENTURY = 36525.00
- const Real GmatTimeConstants::DAYS_PER_SEC = 1.1574074074074074074074074074074e-5
- const Real GmatTimeConstants::TIME_OF_J2000 = 883655990.850000
- const Real GmatTimeConstants::JD_OF_J2000 = 2451545.0
- const Real GmatTimeConstants::MJD_OF_J2000 = 21545.00000000
- const Real GmatTimeConstants::A1MJD_OF_J2000 = 21545.00000000
- const Real GmatTimeConstants::JD_MJD_OFFSET = 2400000.5
- const Real GmatTimeConstants::TT_TAI_OFFSET = 32.184
- const Real GmatTimeConstants::A1_TAI_OFFSET = 0.0343817
- const Real GmatTimeConstants::JD_JAN_5_1941 = 2430000.0
- const Real GmatTimeConstants::JD_NOV_17_1858 = 2400000.5
- const Integer GmatTimeConstants::DAYS_BEFORE_MONTH [12]
- const Integer GmatTimeConstants::LEAP_YEAR_DAYS_BEFORE_MONTH [12]
- const Integer GmatTimeConstants::DAYS_IN_MONTH [12]
- const Integer GmatTimeConstants::LEAP_YEAR_DAYS_IN_MONTH [12]
- const Integer GmatTimeConstants::JULIAN_DATE_OF_010541 = 2430000
- const Real GmatTimeConstants::MJD_EPOCH_PRECISION = 7.27e-12
- const Real GmatAttitudeConstants::QUAT_MIN_MAG = 1.0e-10
- const Real GmatAttitudeConstants::EULER_ANGLE_TOLERANCE = 1.0E-10
- const Real GmatAttitudeConstants::DCM_ORTHONORMALITY_TOLERANCE = 1.0e-14

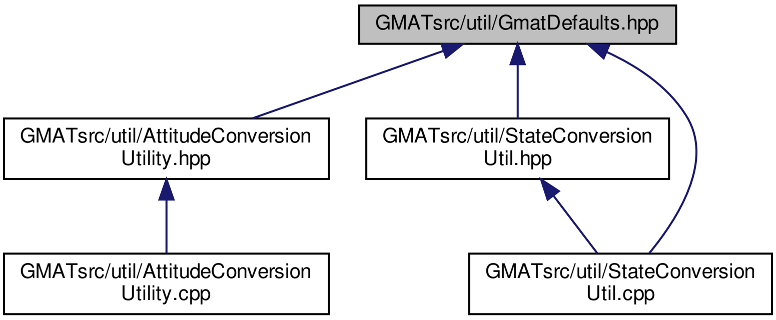
9.35 GMATsrc/util/GmatDefaults.hpp File Reference

```
#include <limits>
#include "utildefs.hpp"
#include "Rmatrix.hpp"
#include "Rvector6.hpp"
```

Include dependency graph for GmatDefaults.hpp:



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


GMATsrc/util/GmatDefaults.hpp
GMATsrc/util/AttitudeConversion Utility.hpp
GMATsrc/util/StateConversion Util.hpp
GMATsrc/util/AttitudeConversion Utility.cpp
GMATsrc/util/StateConversion Util.cpp

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::ADRASTEA_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 = "Promethus"
- 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.

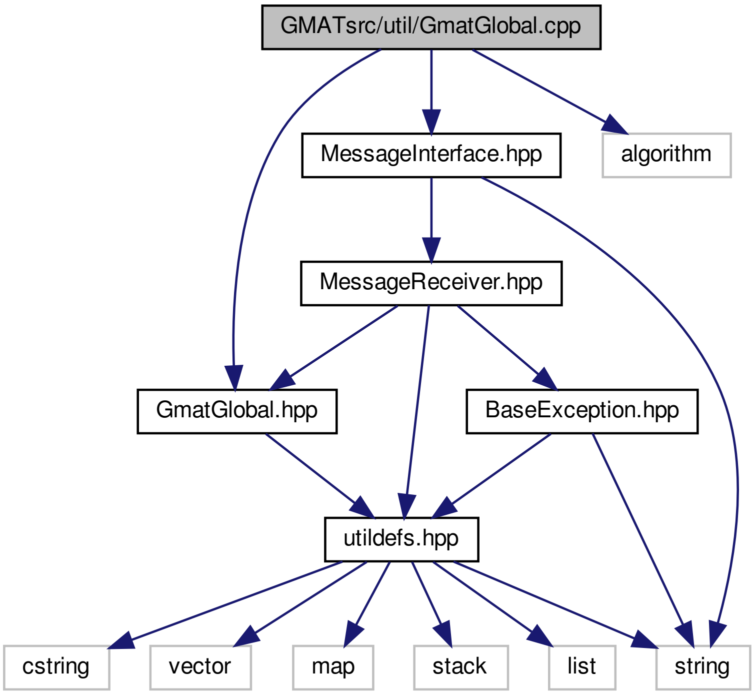
- const Real GmatSolarSystemDefaults::PLANET_EQUATORIAL_RADIUS [NumberOfDefaultPlanets]
- const Real GmatSolarSystemDefaults::PLANET_FLATTENING [NumberOfDefaultPlanets]
- const Real GmatSolarSystemDefaults::PLANET_MU [NumberOfDefaultPlanets]
- const Real GmatSolarSystemDefaults::PLANET_TWO_BODY_EPOCH [NumberOfDefaultPlanets]
- const Rvector6 GmatSolarSystemDefaults::PLANET_TWO_BODY_ELEMENTS [NumberOfDefaultPlanets]
- const Rvector6 GmatSolarSystemDefaults::PLANET_ORIENTATION_PARAMETERS [NumberOfDefault←Planets]
- 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 [NumberOfDefault←Moons]
- 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::AST←RONOMICAL_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>
```

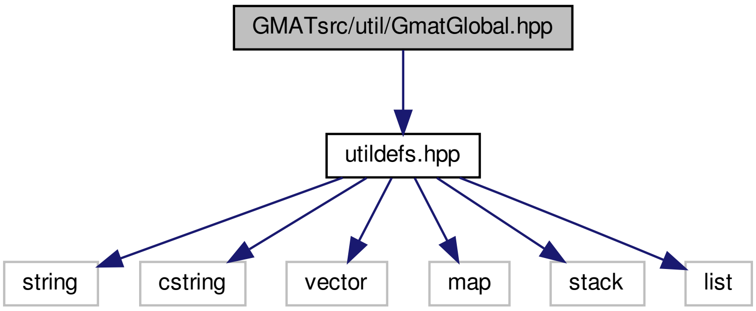
Include dependency graph for GmatGlobal.cpp:



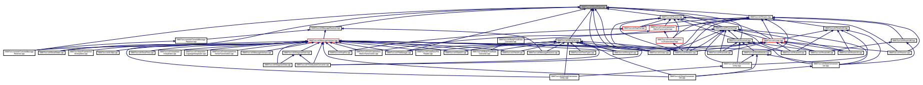
9.37 GMATsrc/util/GmatGlobal.hpp File Reference

```
#include "utildefs.hpp"
```

Include dependency graph for GmatGlobal.hpp:



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



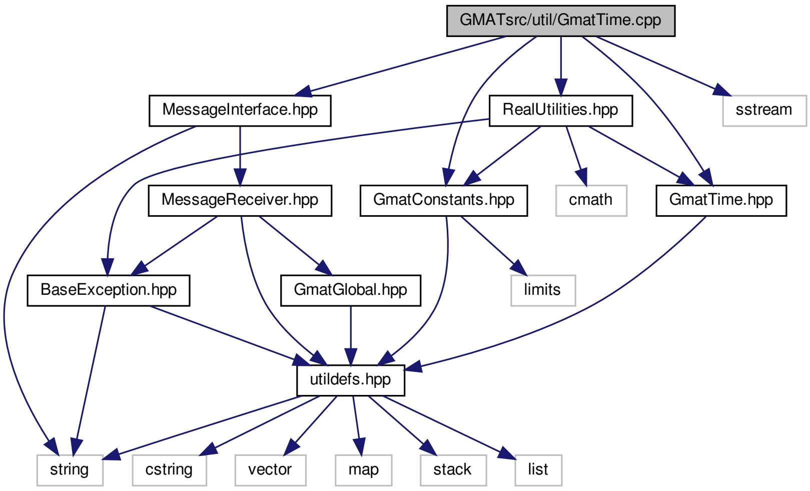
Classes

- class GmatGlobal

9.38 GMATsrc/util/GmatTime.cpp File Reference

```
#include "GmatTime.hpp"
#include "GmatConstants.hpp"
#include "RealUtilities.hpp"
#include "MessageInterface.hpp"
#include <sstream>
```

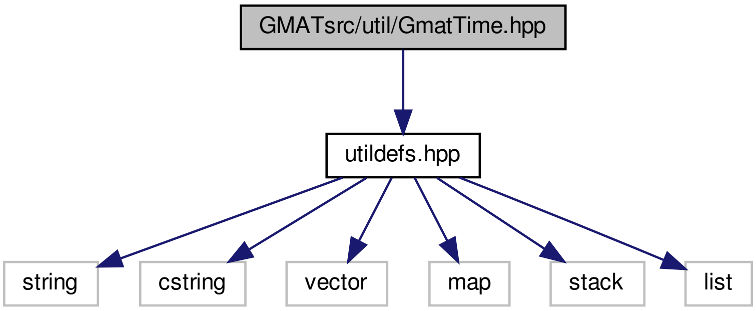
Include dependency graph for GmatTime.cpp:



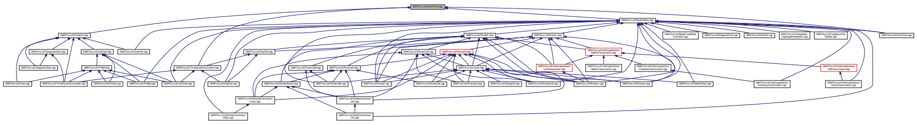
9.39 GMATsrc/util/GmatTime.hpp File Reference

```
#include "utildefs.hpp"
```

Include dependency graph for GmatTime.hpp:



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



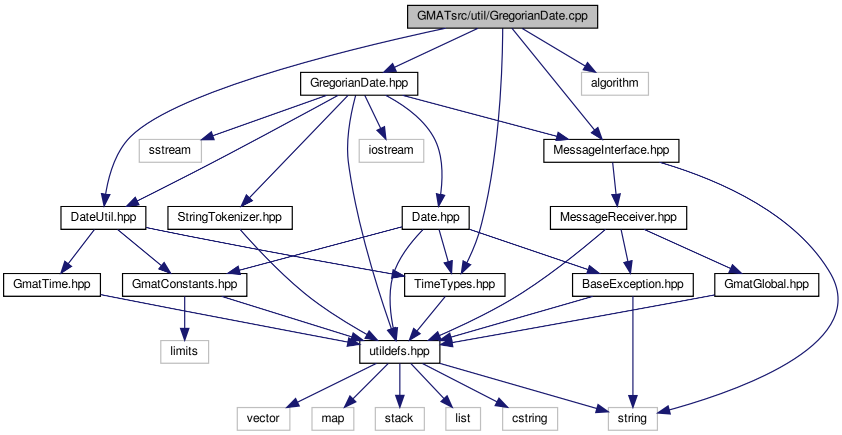
Classes

- class GmatTime

9.40 GMATsrc/util/GregorianDate.cpp File Reference

```
#include "GregorianDate.hpp"
#include "TimeTypes.hpp"
#include "DateUtil.hpp"
#include "MessageInterface.hpp"
#include <algorithm>
```

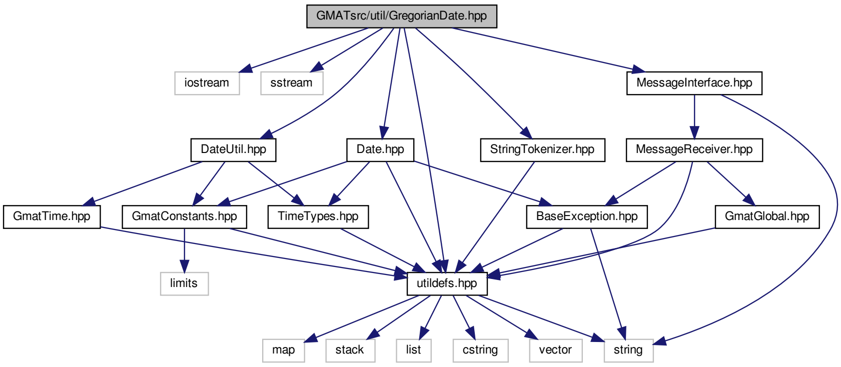
Include dependency graph for GregorianDate.cpp:



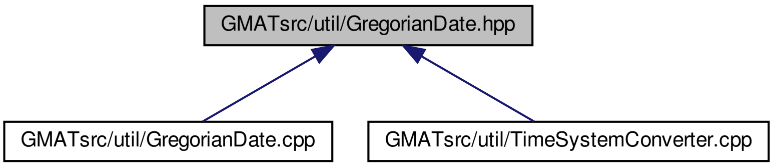
9.41 GMATsrc/util/GregorianDate.hpp File Reference

```
#include <iostream>
#include <sstream>
#include "utildefs.hpp"
#include "Date.hpp"
#include "DateUtil.hpp"
#include "StringTokenizer.hpp"
#include "MessageInterface.hpp"
```

Include dependency graph for GregorianDate.hpp:



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


GMATsrc/util/GregorianDate.hpp
GMATsrc/util/GregorianDate.cpp
GMATsrc/util/TimeSystemConverter.cpp

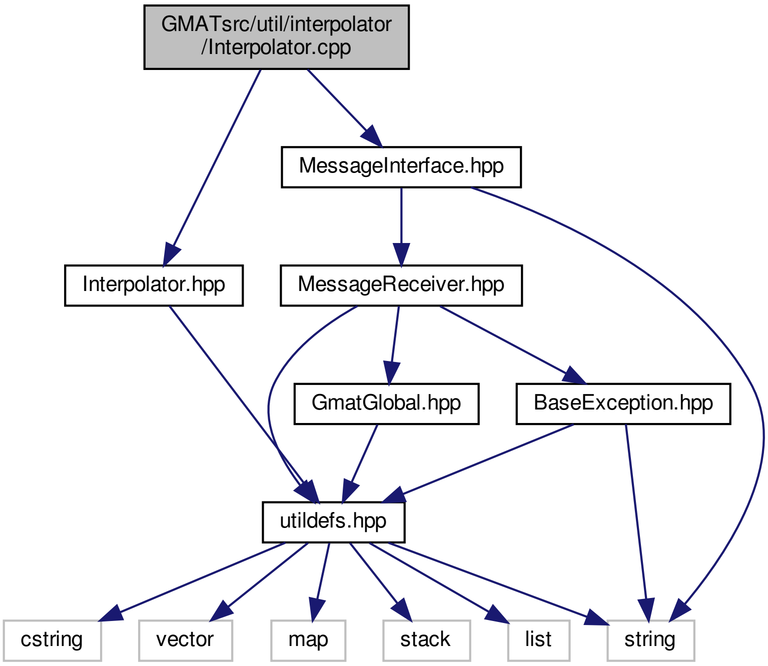
Classes

- class GregorianDate
- class GregorianDate::GregorianDateException

9.42 GMATsrc/util/interpolator/Interpolator.cpp File Reference

```
#include "Interpolator.hpp"
#include "MessageInterface.hpp"
```

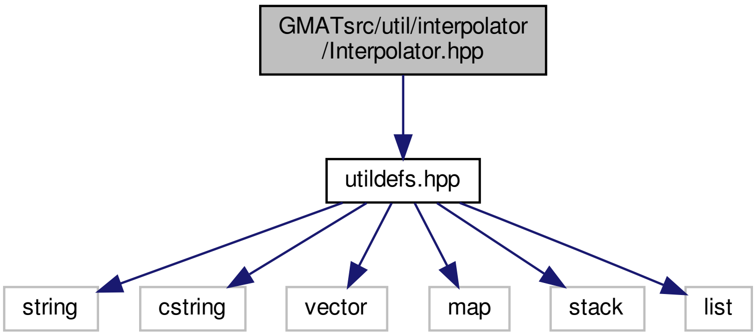
Include dependency graph for Interpolator.cpp:



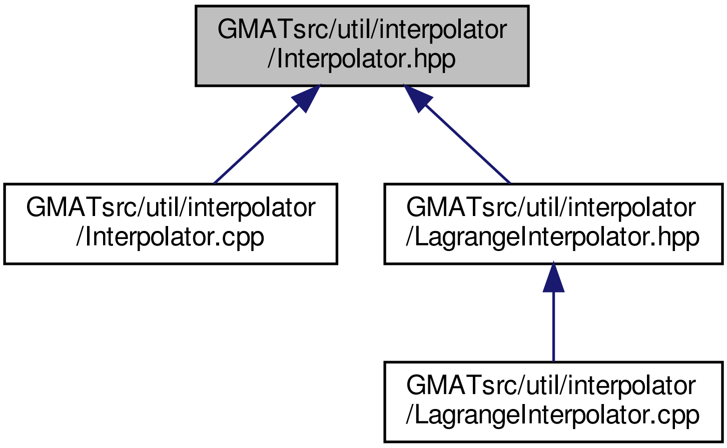
9.43 GMATsrc/util/interpolator/Interpolator.hpp File Reference

```
#include "utildefs.hpp"
```

Include dependency graph for Interpolator.hpp:



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


GMATsrc/util/interpolator
/Interpolator.hpp
GMATsrc/util/interpolator
/Interpolator.cpp
GMATsrc/util/interpolator
/LagrangeInterpolator.hpp
GMATsrc/util/interpolator
/LagrangeInterpolator.cpp

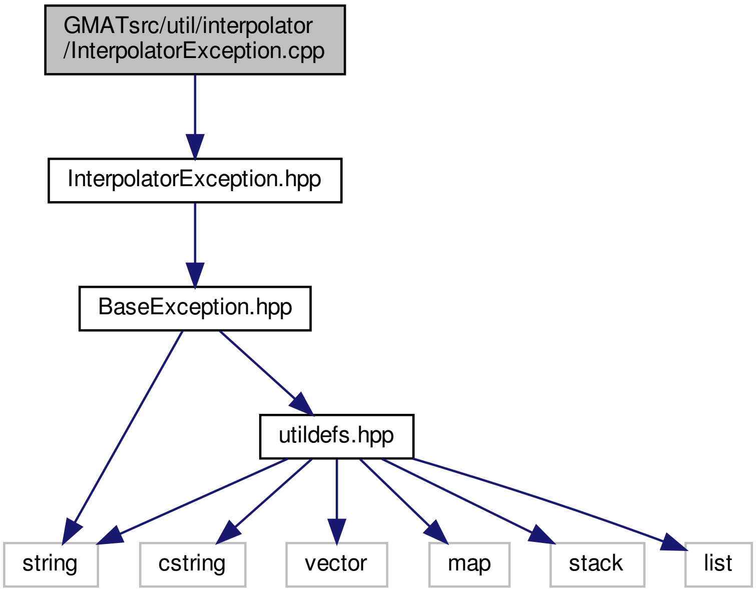
Classes

- class Interpolator

9.44 GMATsrc/util/interpolator/InterpolatorException.cpp File Reference

```
#include "InterpolatorException.hpp"
```

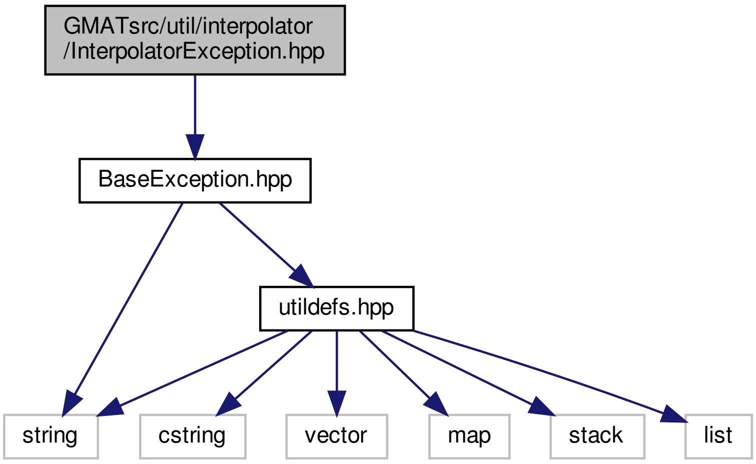
Include dependency graph for InterpolatorException.cpp:


GMATsrc/util/interpolator
/InterpolatorException.cpp
InterpolatorException.hpp
BaseException.hpp
utildefs.hpp
string
cstring
vector
map
stack
list

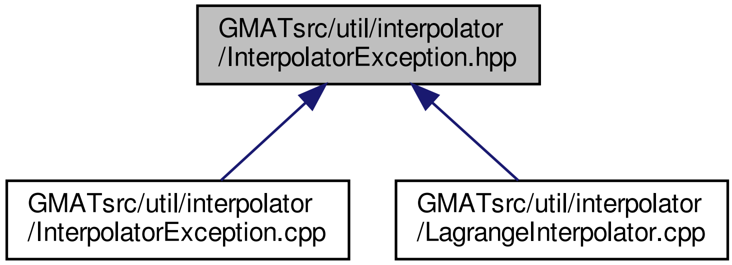
9.45 GMATsrc/util/interpolator/InterpolatorException.hpp File Reference

```
#include "BaseException.hpp"
```

Include dependency graph for InterpolatorException.hpp:



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


GMATsrc/util/interpolator
/InterpolatorException.hpp
GMATsrc/util/interpolator
/InterpolatorException.cpp
GMATsrc/util/interpolator
/LagrangeInterpolator.cpp

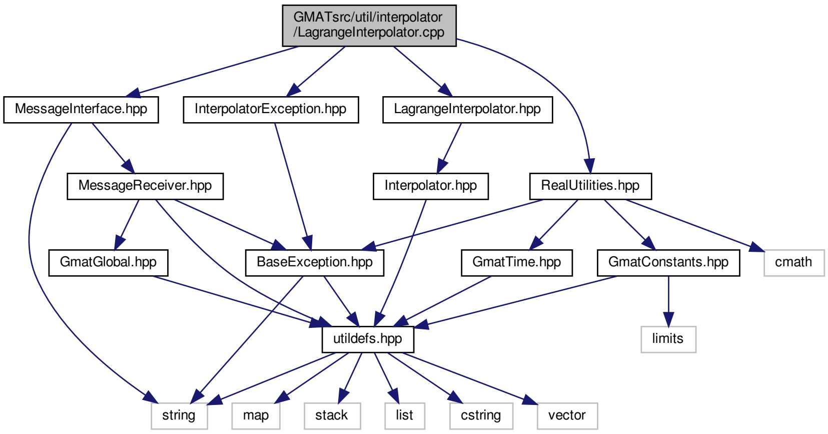
Classes

- class InterpolatorException

9.46 GMATsrc/util/interpolator/LagrangeInterpolator.cpp File Reference

```
#include "LagrangeInterpolator.hpp"
#include "InterpolatorException.hpp"
#include "RealUtilities.hpp"
#include "MessageInterface.hpp"
```

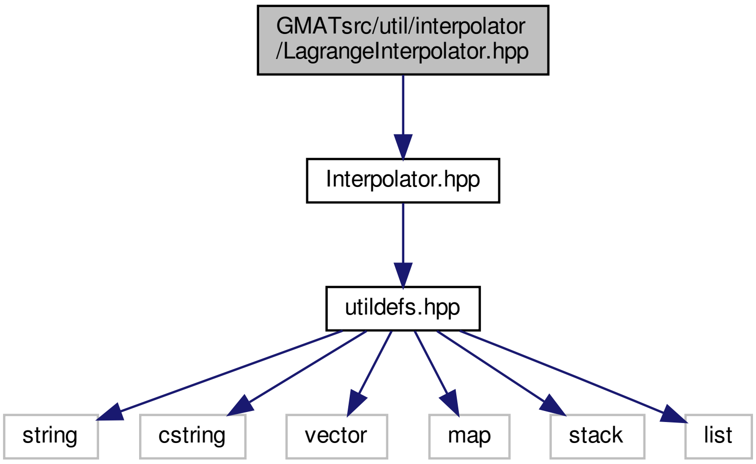
Include dependency graph for LagrangeInterpolator.cpp:



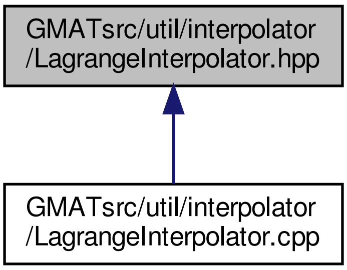
9.47 GMATsrc/util/interpolator/LagrangeInterpolator.hpp File Reference

```
#include "Interpolator.hpp"
```

Include dependency graph for LagrangeInterpolator.hpp:



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


GMATsrc/util/interpolator
/LagrangeInterpolator.hpp
GMATsrc/util/interpolator
/LagrangeInterpolator.cpp

Classes

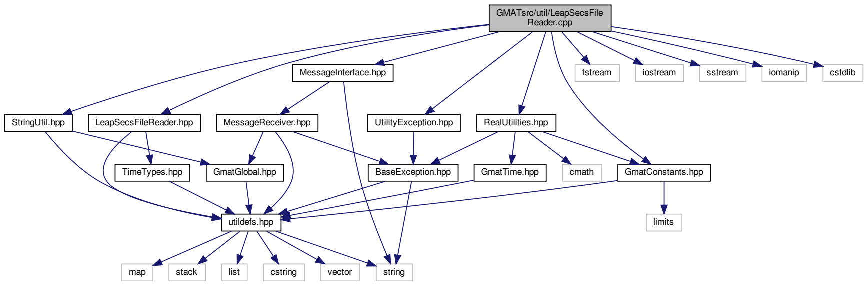
- class LagrangeInterpolator

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>
```

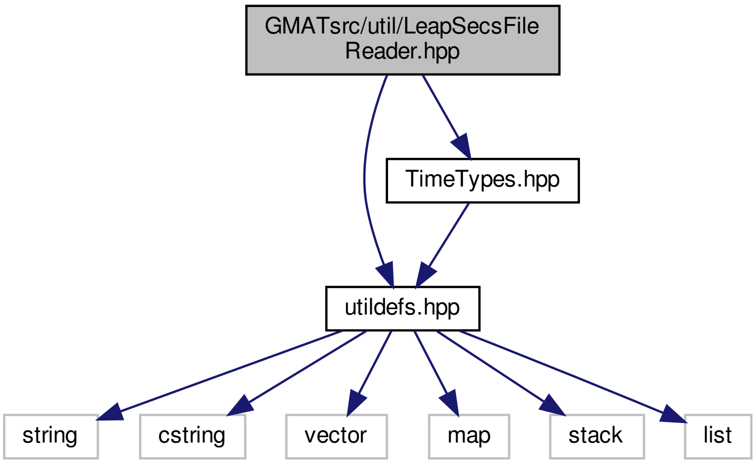
Include dependency graph for LeapSecsFileReader.cpp:



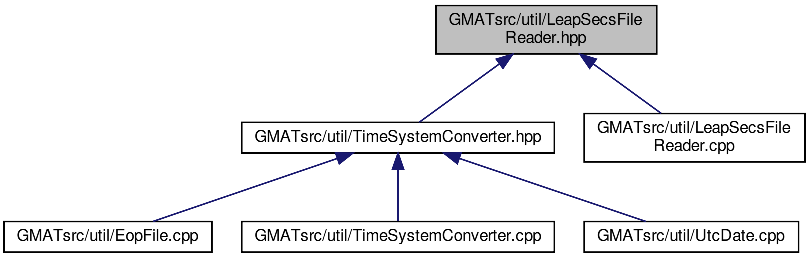
9.49 GMATsrc/util/LeapSecsFileReader.hpp File Reference

```
#include "utildefs.hpp"
#include "TimeTypes.hpp"
```

Include dependency graph for LeapSecsFileReader.hpp:



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


GMATsrc/util/LeapSecsFile Reader.hpp
GMATsrc/util/TimeSystemConverter.hpp
GMATsrc/util/LeapSecsFile Reader.cpp
GMATsrc/util/EopFile.cpp
GMATsrc/util/TimeSystemConverter.cpp
GMATsrc/util/UtcDate.cpp

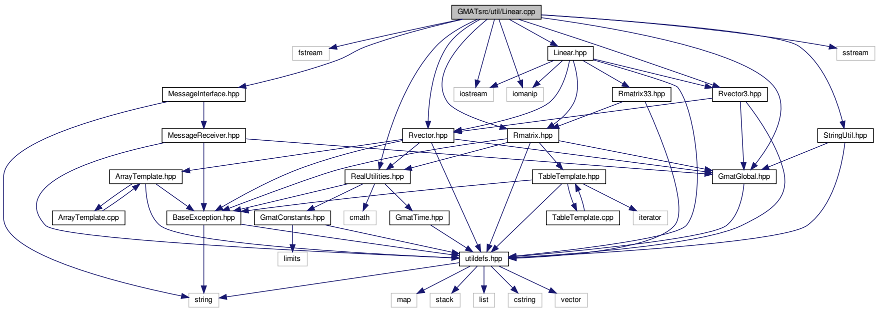
Classes

- 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"
```

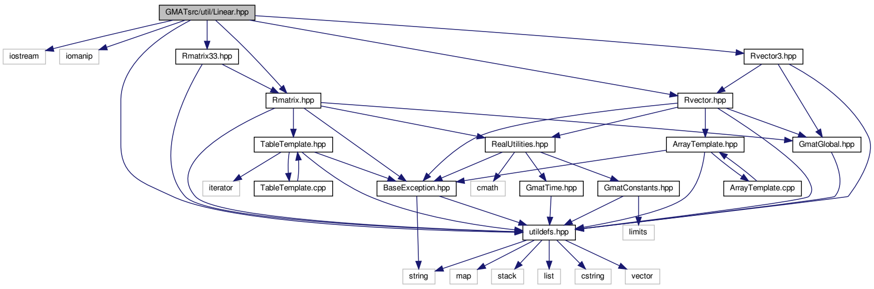
Include dependency graph for Linear.cpp:



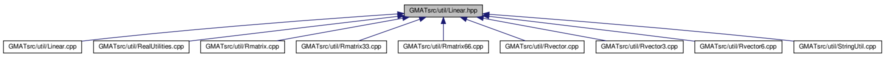
9.51 GMATsrc/util/Linear.hpp File Reference

```
#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:


GMATsrc/util/Linear.hpp
GMATsrc/util/Linear.cpp
GMATsrc/util/RealUtilities.cpp
GMATsrc/util/Rmatrix.cpp
GMATsrc/util/Rmatrix33.cpp
GMATsrc/util/Rmatrix66.cpp
GMATsrc/util/Rvector.cpp
GMATsrc/util/Rvector3.cpp
GMATsrc/util/Rvector6.cpp
GMATsrc/util/StringUtil.cpp

Classes

- struct GmatRealUtil::RaCodec
- struct GmatRealUtil::RaDec

Namespaces

- GmatRealUtil

Functions

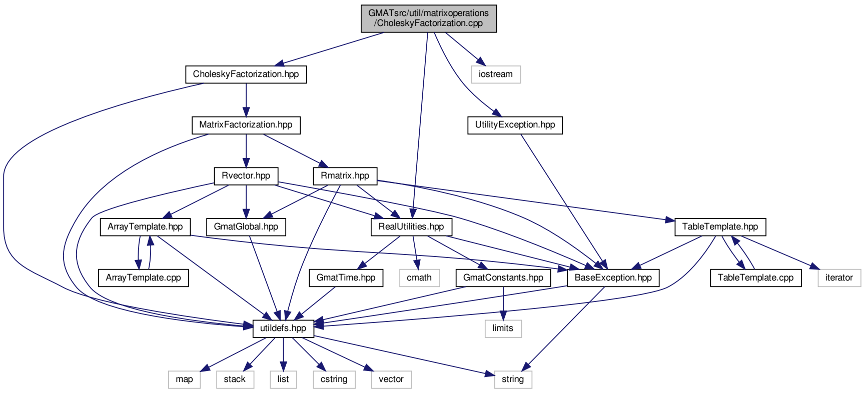
- GMATUTIL_API RaCodec GmatRealUtil::CartesianToRaCodec (const Rvector3 &r)
- GMATUTIL_API RaDec GmatRealUtil::CartesianToRaDec (const Rvector3 &r)
- GMATUTIL_API Rvector3 GmatRealUtil::RaCodecToCartesian (const RaCodec &r)
- GMATUTIL_API RaDec GmatRealUtil::RaCodecToRaDec (const RaCodec &r)
- GMATUTIL_API 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>
```

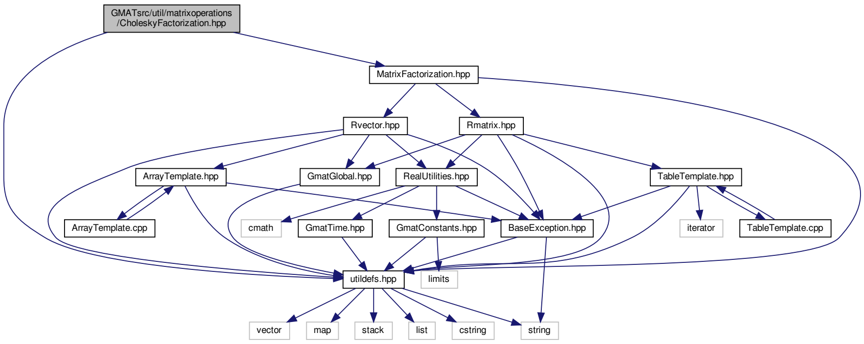
Include dependency graph for CholeskyFactorization.cpp:



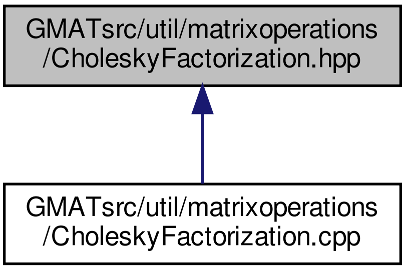
9.53 GMATsrc/util/matrixoperations/CholeskyFactorization.hpp File Reference

```
#include "utildefs.hpp"
#include "MatrixFactorization.hpp"
```

Include dependency graph for CholeskyFactorization.hpp:



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


GMATsrc/util/matrixoperations
/CholeskyFactorization.hpp
GMATsrc/util/matrixoperations
/CholeskyFactorization.cpp

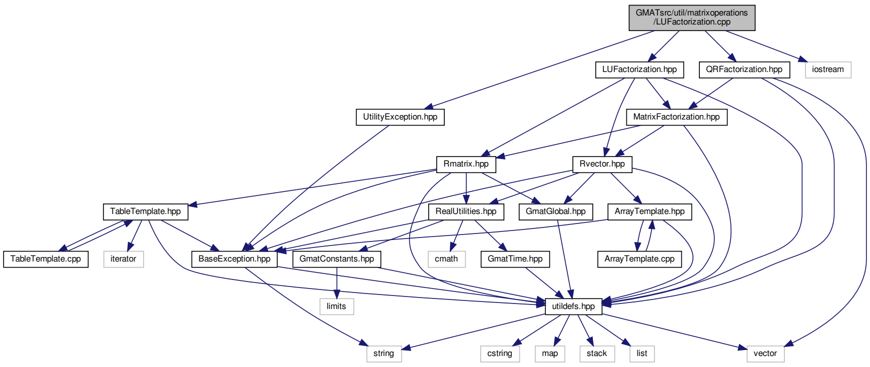
Classes

- 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:

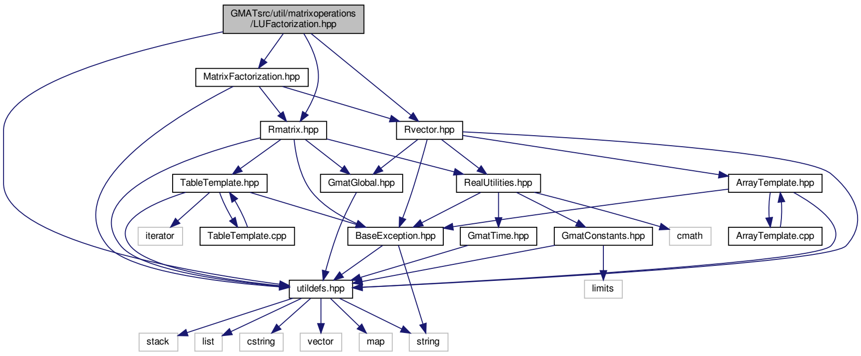


9.55 GMATsrc/util/matrixoperations/LUFactorization.hpp File Reference

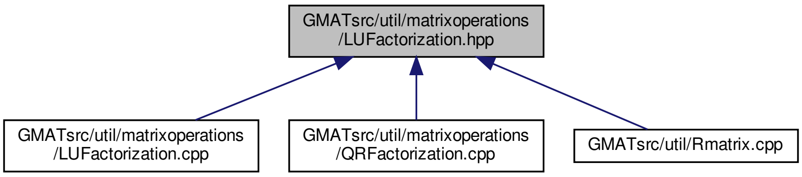
```
#include "Rmatrix.hpp"
#include "Rvector.hpp"
#include "utildefs.hpp"
```

```
#include "MatrixFactorization.hpp"
```

Include dependency graph for LUFactorization.hpp:



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


GMATsrc/util/matrixoperations
/LUFactorization.hpp
GMATsrc/util/matrixoperations
/LUFactorization.cpp
GMATsrc/util/matrixoperations
/QRFactorization.cpp
GMATsrc/util/Rmatrix.cpp

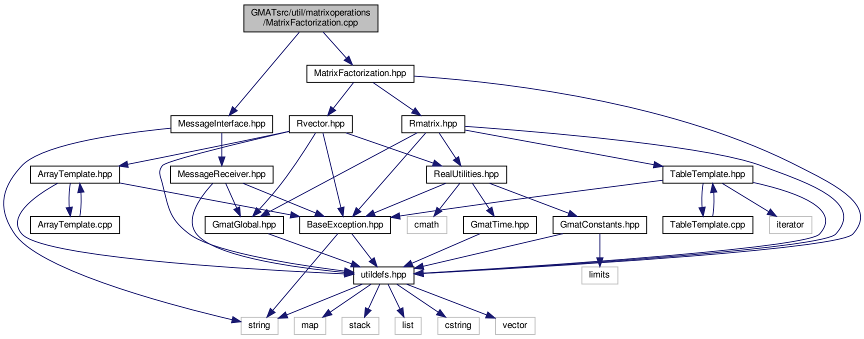
Classes

- class LUFactorization

9.56 GMATsrc/util/matrixoperations/MatrixFactorization.cpp File Reference

```
#include "MatrixFactorization.hpp"
#include "MessageInterface.hpp"
```

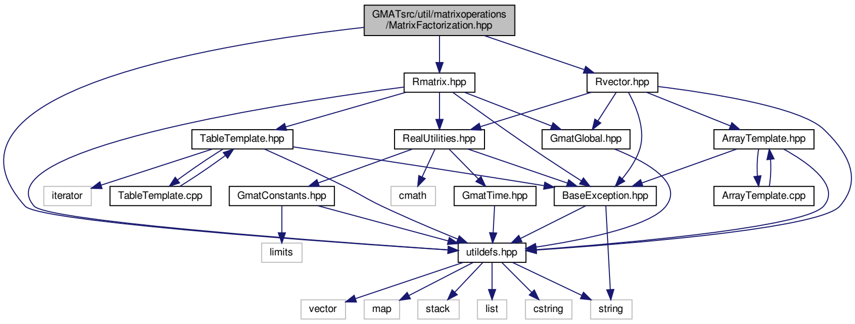
Include dependency graph for MatrixFactorization.cpp:



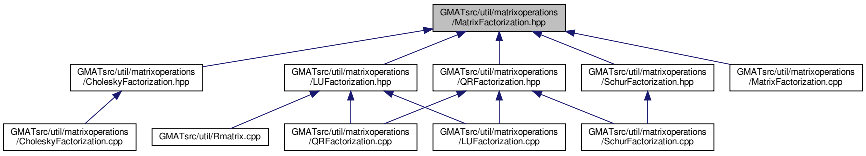
9.57 GMATsrc/util/matrixoperations/MatrixFactorization.hpp File Reference

```
#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:


GMATsrc/util/matrixoperations
/MatrixFactorization.hpp
GMATsrc/util/matrixoperations
/CholeskyFactorization.hpp
GMATsrc/util/matrixoperations
/LUFactorization.hpp
GMATsrc/util/matrixoperations
/QRFactorization.hpp
GMATsrc/util/matrixoperations
/SchurFactorization.hpp
GMATsrc/util/matrixoperations
/MatrixFactorization.cpp
GMATsrc/util/matrixoperations
/CholeskyFactorization.cpp
GMATsrc/util/Rmatrix.cpp
GMATsrc/util/matrixoperations
/QRFactorization.cpp
GMATsrc/util/matrixoperations
/LUFactorization.cpp
GMATsrc/util/matrixoperations
/SchurFactorization.cpp

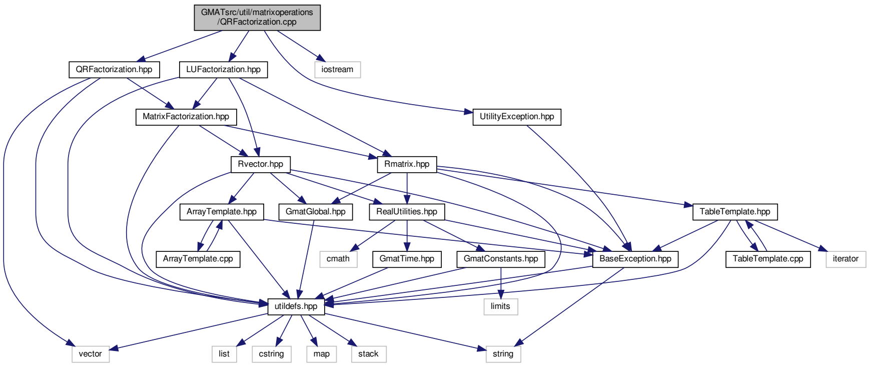
Classes

- class MatrixFactorization

9.58 GMATsrc/util/matrixoperations/QRFactorization.cpp File Reference

```
#include "QRFactorization.hpp"
#include "UtilityException.hpp"
#include "LUFactorization.hpp"
#include <iostream>
```

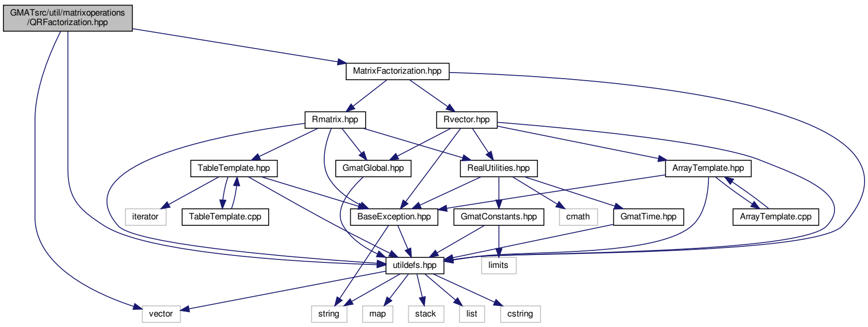
Include dependency graph for QRFactorization.cpp:



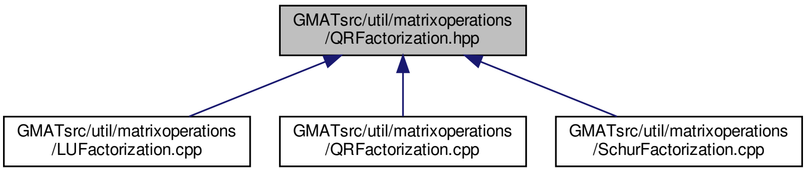
9.59 GMATsrc/util/matrixoperations/QRFactorization.hpp File Reference

```
#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:


GMATsrc/util/matrixoperations
/QRFactorization.hpp
GMATsrc/util/matrixoperations
/LUFactorization.cpp
GMATsrc/util/matrixoperations
/QRFactorization.cpp
GMATsrc/util/matrixoperations
/SchurFactorization.cpp

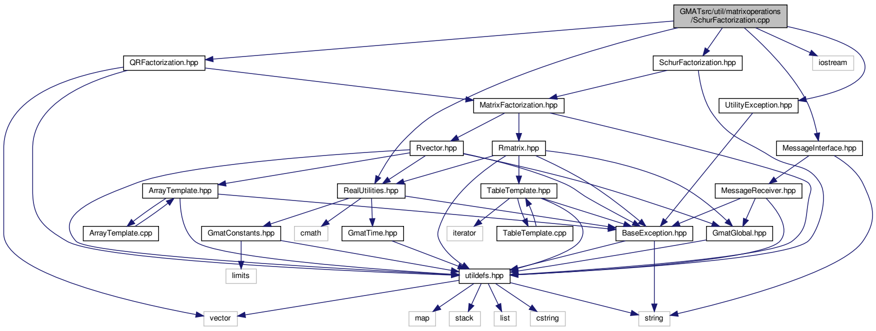
Classes

- class QRFactorization

9.60 GMATsrc/util/matrixoperations/SchurFactorization.cpp File Reference

```
#include "SchurFactorization.hpp"
#include "QRFactorization.hpp"
#include "RealUtilities.hpp"
#include "UtilityException.hpp"
#include <iostream>
#include "MessageInterface.hpp"
```

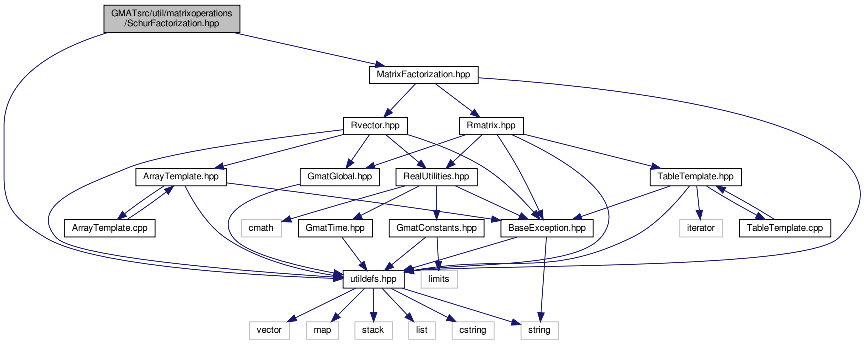
Include dependency graph for SchurFactorization.cpp:



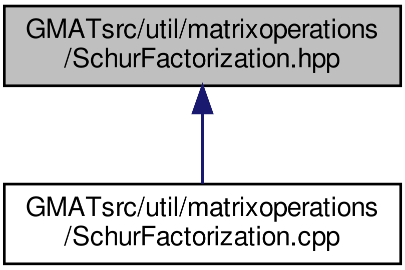
9.61 GMATsrc/util/matrixoperations/SchurFactorization.hpp File Reference

```
#include "utildefs.hpp"
#include "MatrixFactorization.hpp"
```

Include dependency graph for SchurFactorization.hpp:



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



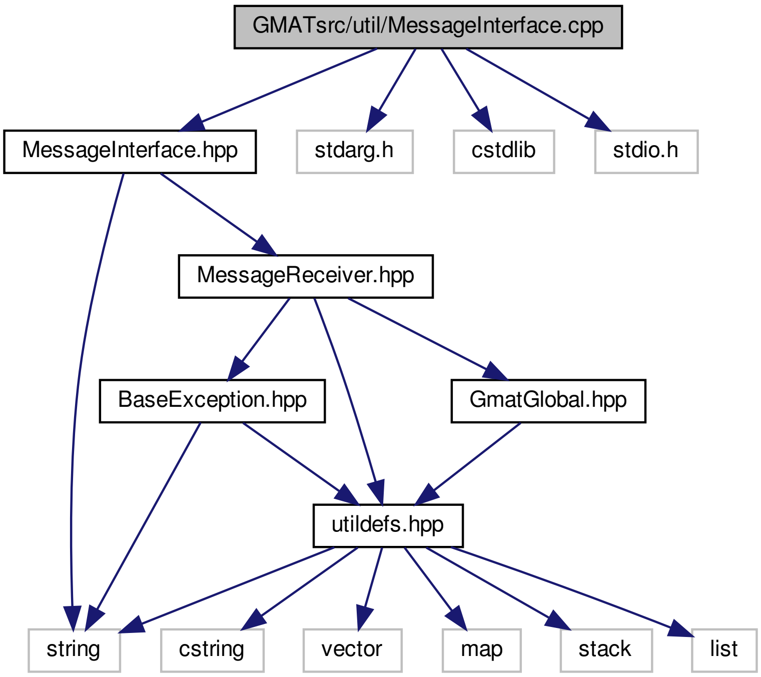
Classes

- class SchurFactorization

9.62 GMATsrc/util/MessageInterface.cpp File Reference

```
#include "MessageInterface.hpp"
#include <stdarg.h>
#include <cstdlib>
#include <stdio.h>
```

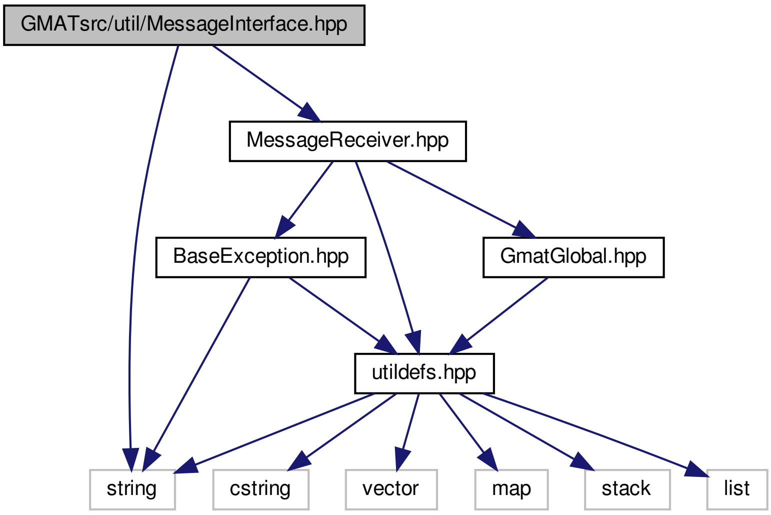
Include dependency graph for MessageInterface.cpp:



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:

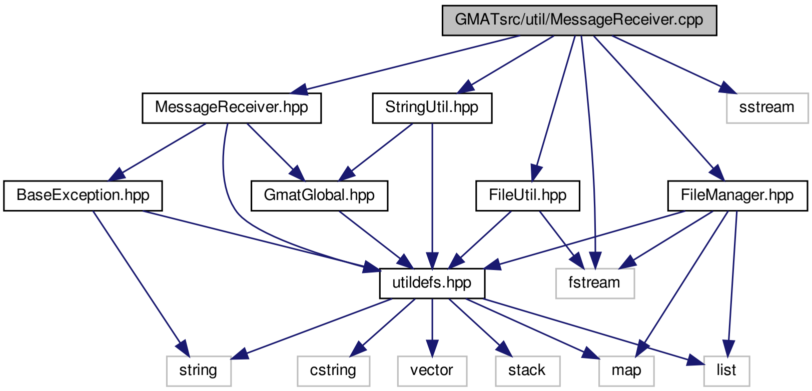
Classes

- 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>
```

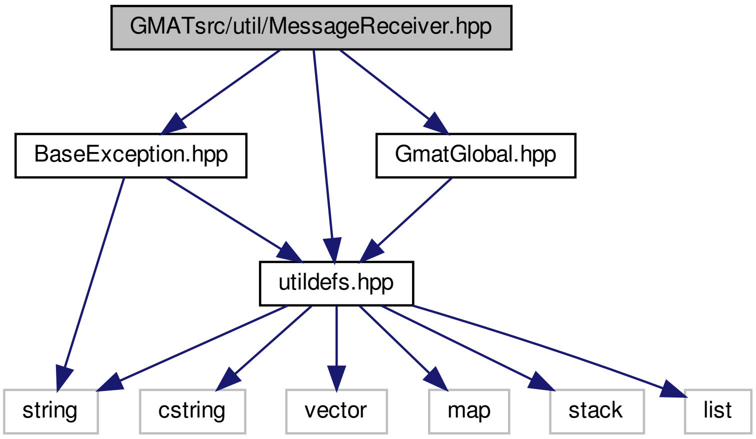
Include dependency graph for MessageReceiver.cpp:



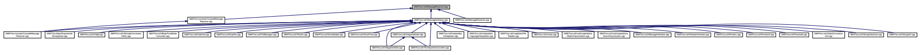
9.65 GMATsrc/util/MessageReceiver.hpp File Reference

```
#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:



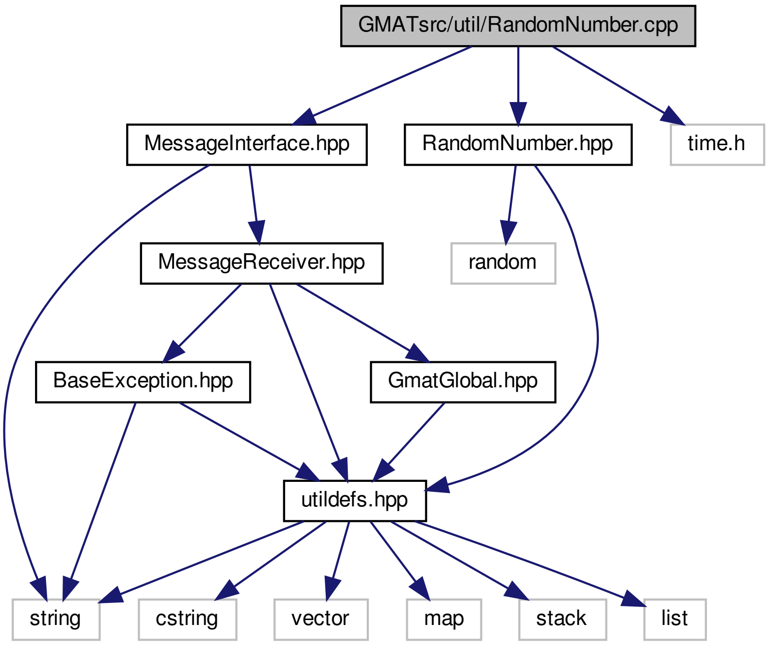
Classes

- class MessageReceiver

9.66 GMATsrc/util/RandomNumber.cpp File Reference

```
#include "RandomNumber.hpp"
#include "MessageInterface.hpp"
#include <time.h>
```

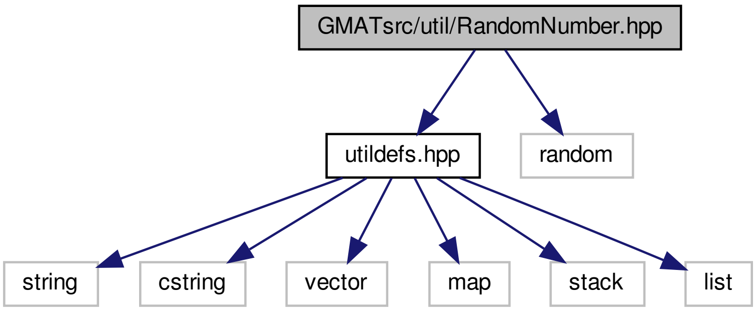
Include dependency graph for RandomNumber.cpp:



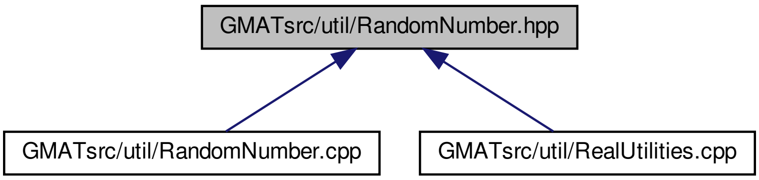
9.67 GMATsrc/util/RandomNumber.hpp File Reference

```
#include "utildefs.hpp"
#include <random>
```

Include dependency graph for RandomNumber.hpp:



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


GMATsrc/util/RandomNumber.hpp
GMATsrc/util/RandomNumber.cpp
GMATsrc/util/RealUtilities.cpp

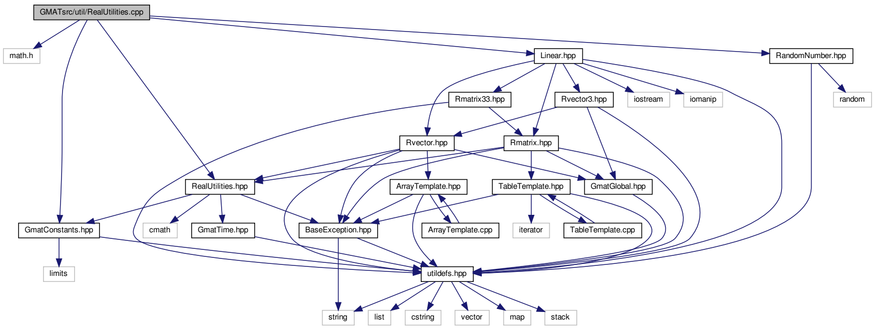
Classes

- 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"
```

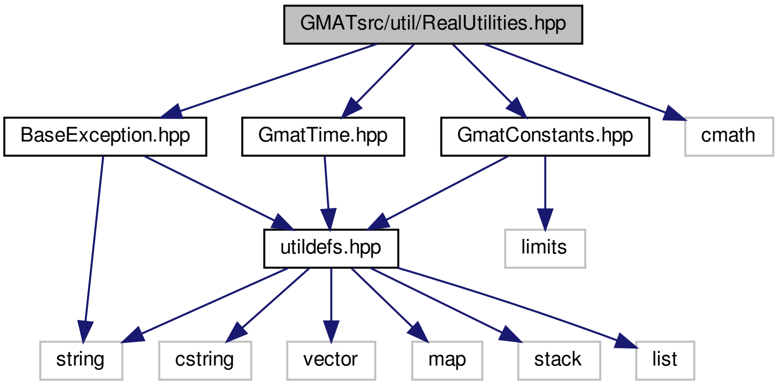
Include dependency graph for RealUtilities.cpp:



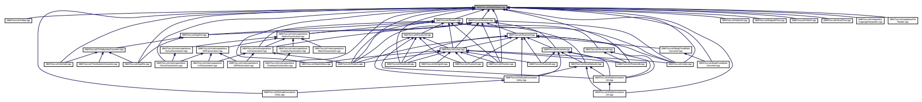
9.69 GMATsrc/util/RealUtilities.hpp File Reference

```
#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

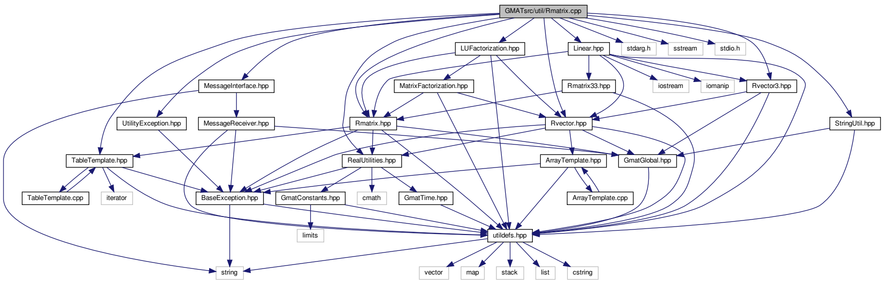
- Integer GMATUTIL_API GmatMathUtil::Abs (Integer theNumber)
- Real GMATUTIL_API GmatMathUtil::Abs (Real theNumber)
- Real GMATUTIL_API GmatMathUtil::NearestInt (Real theNumber)
- Real GMATUTIL_API GmatMathUtil::Round (Real theNumber)
- Real GMATUTIL_API GmatMathUtil::Floor (Real theNumber)
- Real GMATUTIL_API 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

output	Output stream
a	Rmatrix 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 )
```

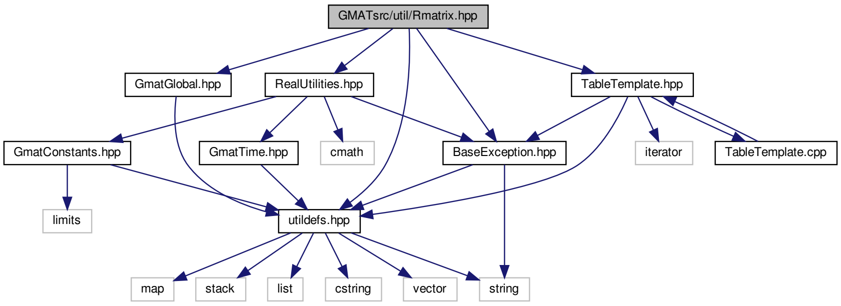
9.70.1.10 TransposeTimesTranspose()

```
Rmatrix TransposeTimesTranspose (
             const Rmatrix & m1,
             const Rmatrix & m2 )
```

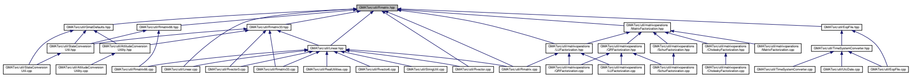
9.71 GMATsrc/util/Rmatrix.hpp File Reference

```
#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

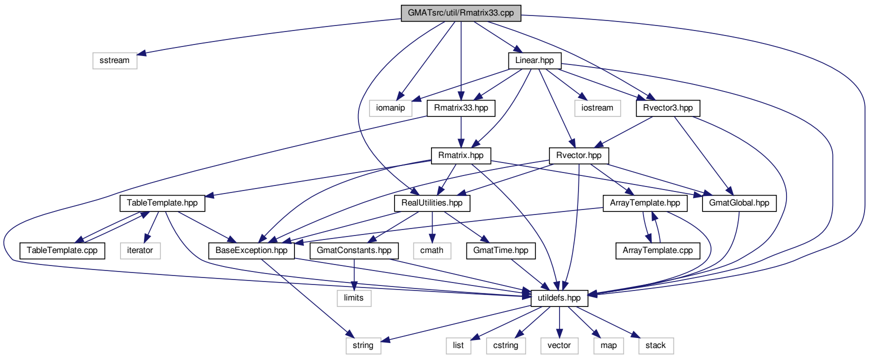
- class Rmatrix
- class Rmatrix::NotSquare
- class Rmatrix::IsSingular
- class Rmatrix::DivideByZero

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 )
```

9.72.1.3 SkewSymmetric()

```
Rmatrix33 SkewSymmetric (
             const Rvector3 & v )
```

9.72.1.4 TransposeTimesRmatrix()

```
Rmatrix33 TransposeTimesRmatrix (
             const Rmatrix33 & m1,
             const Rmatrix33 & m2 )
```

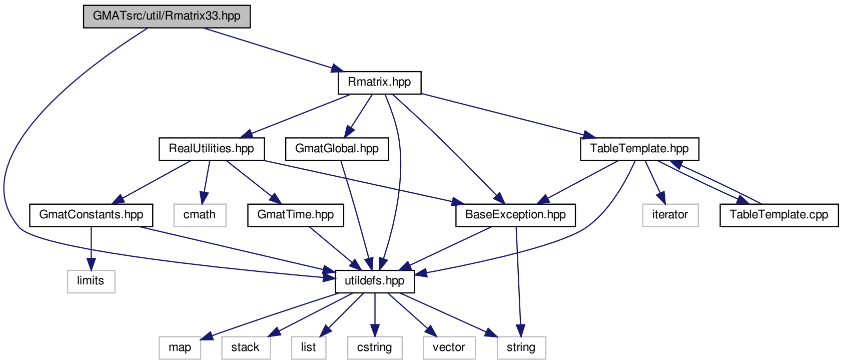
9.72.1.5 TransposeTimesTranspose()

```
Rmatrix33 TransposeTimesTranspose (
             const Rmatrix33 & m1,
             const Rmatrix33 & m2 )
```

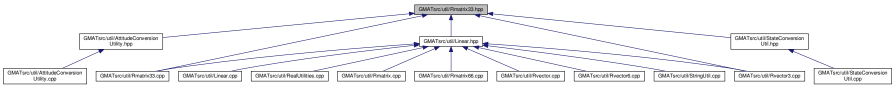
9.73 GMATsrc/util/Rmatrix33.hpp File Reference

```
#include "utildefs.hpp"
#include "Rmatrix.hpp"
```

Include dependency graph for Rmatrix33.hpp:



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


GMATsrc/util/Rmatrix33.hpp
GMATsrc/util/AttitudeConversion Utility.hpp
GMATsrc/util/Linear.hpp
GMATsrc/util/StateConversion Util.hpp
GMATsrc/util/AttitudeConversion Utility.cpp
GMATsrc/util/Rmatrix33.cpp
GMATsrc/util/Linear.cpp
GMATsrc/util/RealUtilities.cpp
GMATsrc/util/Rmatrix.cpp
GMATsrc/util/Rmatrix66.cpp
GMATsrc/util/Rvector.cpp
GMATsrc/util/Rvector6.cpp
GMATsrc/util/StringUtil.cpp
GMATsrc/util/Rvector3.cpp
GMATsrc/util/StateConversion Util.cpp

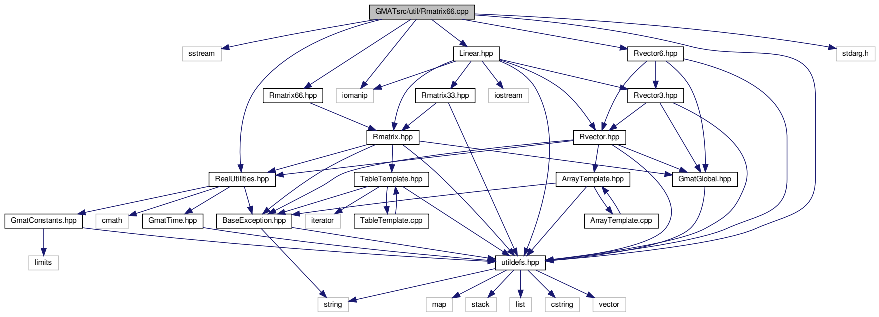
Classes

- class Rmatrix33

9.74 GMATsrc/util/Rmatrix66.cpp File Reference

```
#include <sstream>
#include <iomanip>
#include "Rmatrix66.hpp"
#include "utildefs.hpp"
#include "Rvector6.hpp"
#include "RealUtilities.hpp"
#include "Linear.hpp"
#include <stdarg.h>
```

Include dependency graph for Rmatrix66.cpp:



Functions

- 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)

9.74.1 Function Documentation

9.74.1.1 MatrixTimesTranspose()

```
Rmatrix66 MatrixTimesTranspose (
            const Rmatrix66 & m1,
            const Rmatrix66 & m2 )
```

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 )
```

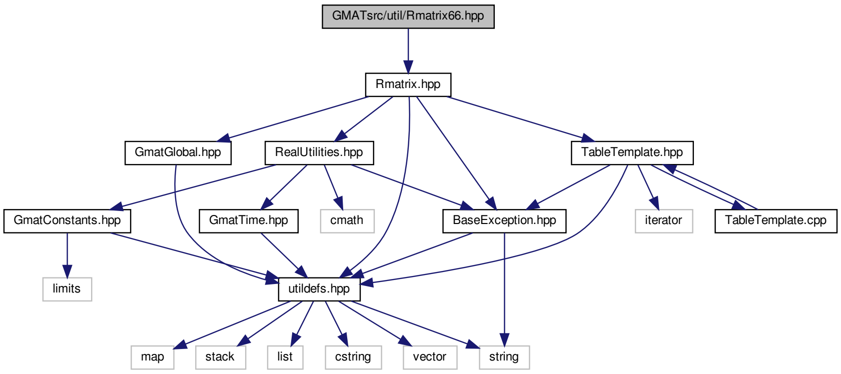
9.74.1.5 TransposeTimesTranspose()

```
Rmatrix66 TransposeTimesTranspose (
             const Rmatrix66 & m1,
             const Rmatrix66 & m2 )
```

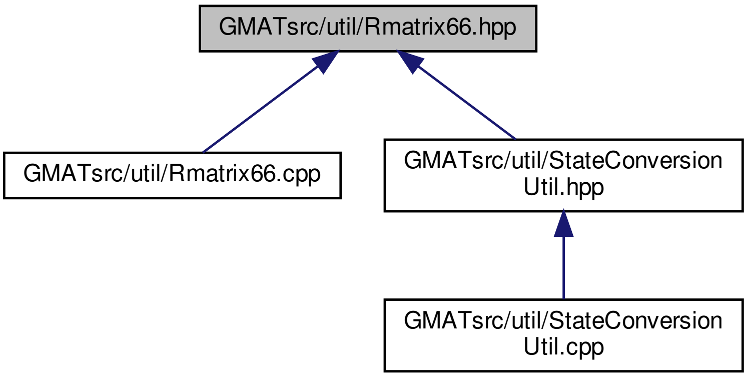
9.75 GMATsrc/util/Rmatrix66.hpp File Reference

```
#include "Rmatrix.hpp"
```

Include dependency graph for Rmatrix66.hpp:



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


GMATsrc/util/Rmatrix66.hpp
GMATsrc/util/Rmatrix66.cpp
GMATsrc/util/StateConversion Util.hpp
GMATsrc/util/StateConversion Util.cpp

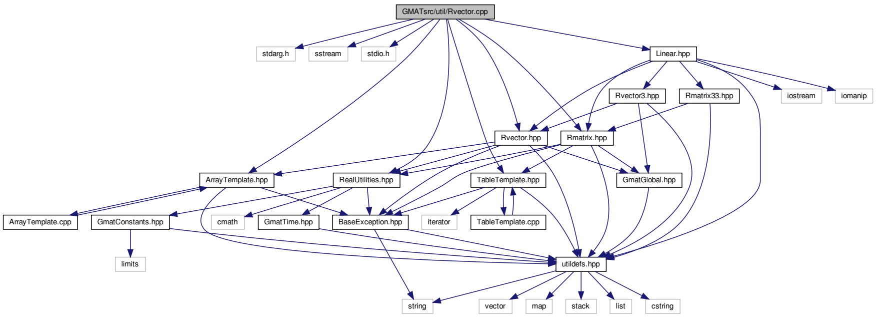
Classes

- 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

e2	right hand Element 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

output	Output stream
a	Rvector 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 )
```

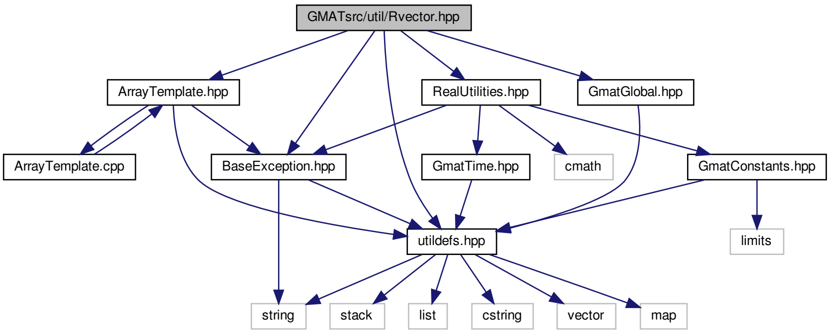
Note

Rmatrix multiplication, with first vector considered a Nx1 matrix and a second vector considered a 1xM matrix.

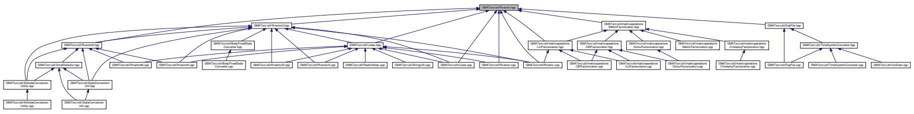
9.77 GMATsrc/util/Rvector.hpp File Reference

```
#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 Element in test

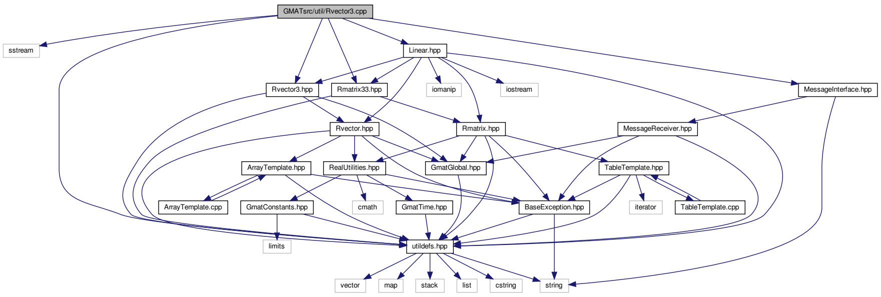
Returns

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"
```

Include dependency graph for Rvector3.cpp:



Functions

- Rvector3 operator∗ (Real s, const Rvector3 &v)
- Rvector3 Cross (const Rvector3 &v1, const Rvector3 &v2)
- Rmatrix33 Outerproduct (const Rvector3 &v1, const Rvector3 &v2)
- std::istream & operator>> (std::istream &input, Rvector3 &a)
- std::ostream & operator<< (std::ostream &output, const Rvector3 &a)

9.78.1 Function Documentation

9.78.1.1 Cross()

```
Rvector3 Cross (
            const Rvector3 & v1,
            const Rvector3 & v2 )
```

9.78.1.2 operator∗()

```
Rvector3 operator* (
            Real s,
            const Rvector3 & v )
```

9.78.1.3 operator<<()

```
std::ostream& operator<< (
            std::ostream & output,
            const Rvector3 & a )
```

9.78.1.4 operator>>()

```
std::istream& operator>> (
            std::istream & input,
            Rvector3 & a )
```

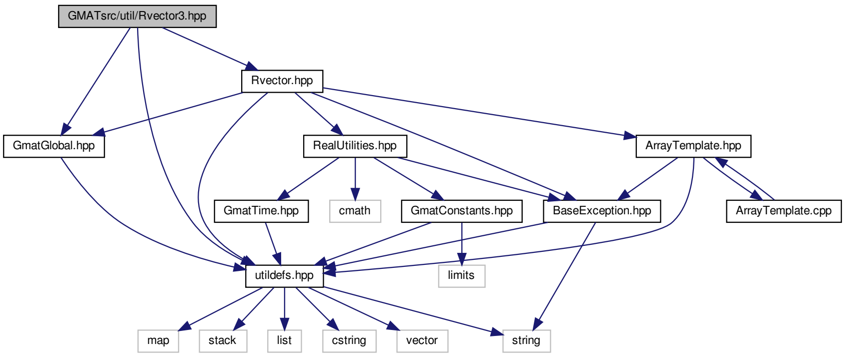
9.78.1.5 Outerproduct()

```
Rmatrix33 Outerproduct (
            const Rvector3 & v1,
            const Rvector3 & v2 )
```

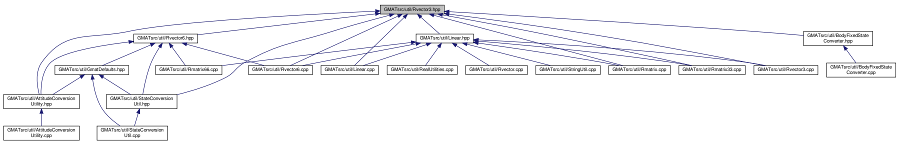
9.79 GMATsrc/util/Rvector3.hpp File Reference

```
#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:


GMATsrc/util/Rvector3.hpp
GMATsrc/util/Rvector6.hpp
GMATsrc/util/Linear.hpp
GMATsrc/util/BodyFixedState Converter.hpp
GMATsrc/util/GmatDefaults.hpp
GMATsrc/util/Rmatrix66.cpp
GMATsrc/util/Rvector6.cpp
GMATsrc/util/Linear.cpp
GMATsrc/util/RealUtilities.cpp
GMATsrc/util/Rvector.cpp
GMATsrc/util/StringUtil.cpp
GMATsrc/util/Rmatrix.cpp
GMATsrc/util/Rmatrix33.cpp
GMATsrc/util/Rvector3.cpp
GMATsrc/util/BodyFixedState Converter.cpp
GMATsrc/util/AttitudeConversion Utility.hpp
GMATsrc/util/StateConversion Util.hpp
GMATsrc/util/AttitudeConversion Utility.cpp
GMATsrc/util/StateConversion Util.cpp

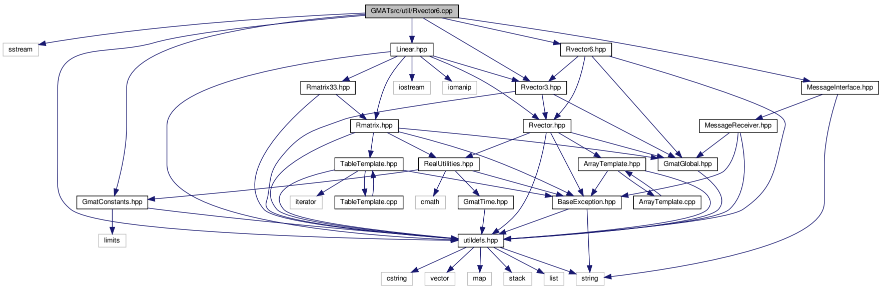
Classes

- 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"
```

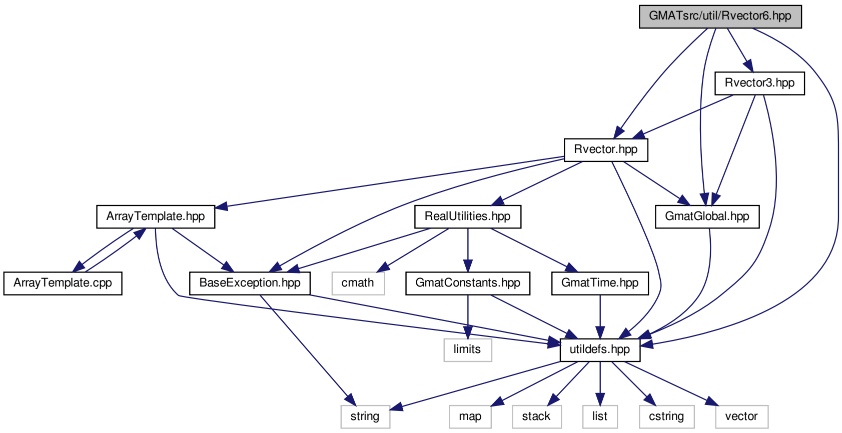
Include dependency graph for Rvector6.cpp:



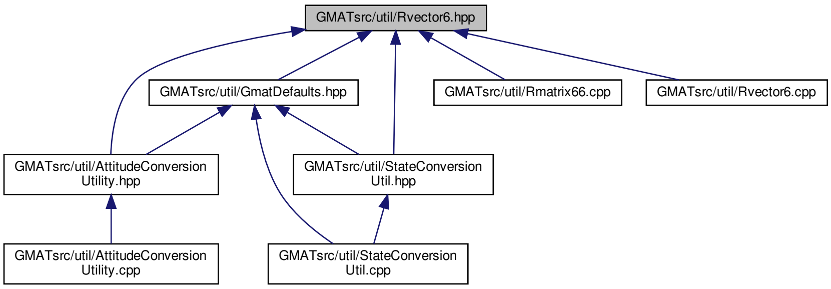
9.81 GMATsrc/util/Rvector6.hpp File Reference

```
#include "utildefs.hpp"
#include "Rvector.hpp"
#include "Rvector3.hpp"
#include "GmatGlobal.hpp"
```

Include dependency graph for Rvector6.hpp:



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


GMATsrc/util/Rvector6.hpp
GMATsrc/util/GmatDefaults.hpp
GMATsrc/util/Rmatrix66.cpp
GMATsrc/util/Rvector6.cpp
GMATsrc/util/AttitudeConversionUtility.hpp
GMATsrc/util/StateConversionUtil.hpp
GMATsrc/util/AttitudeConversionUtility.cpp
GMATsrc/util/StateConversionUtil.cpp

Classes

- class Rvector6

Macros

- #define NUM_DATA_INIT 6

9.81.1 Macro Definition Documentation

9.81.1.1 NUM_DATA_INIT

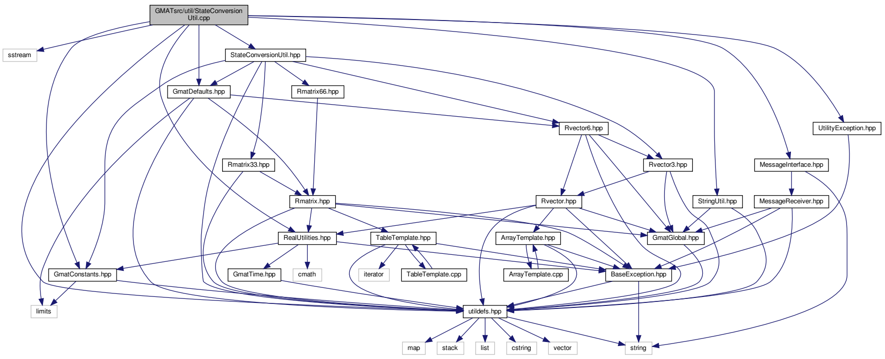
```
#define NUM_DATA_INIT 6
```

9.82 GMATsrc/util/StateConversionUtil.cpp File Reference

```
#include <sstream>
#include "utildefs.hpp"
#include "StateConversionUtil.hpp"
#include "GmatDefaults.hpp"
#include "GmatConstants.hpp"
#include "RealUtilities.hpp"
#include "MessageInterface.hpp"
#include "UtilityException.hpp"
```

```
#include "StringUtil.hpp"
```

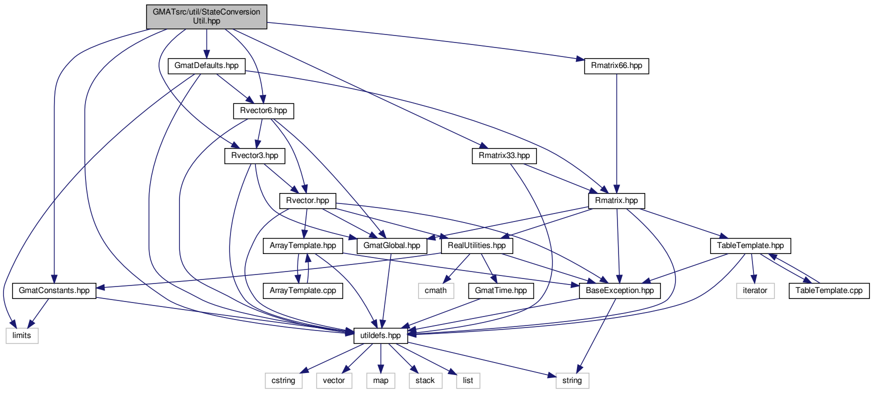
Include dependency graph for StateConversionUtil.cpp:



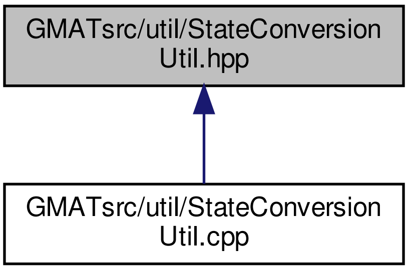
9.83 GMATsrc/util/StateConversionUtil.hpp File Reference

```
#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:


GMATsrc/util/StateConversion Util.hpp
GMATsrc/util/StateConversion Util.cpp

Classes

- class StateConversionUtil

Macros

- #define EARTH_MU GmatSolarSystemDefaults::PLANET_MU[GmatSolarSystemDefaults::EARTH]
- #define EARTH_FLATTENING GmatSolarSystemDefaults::PLANET_FLATTENING[GmatSolarSystem↩ Defaults::EARTH]
- #define EARTH_EQ_RADIUS GmatSolarSystemDefaults::PLANET_EQUATORIAL_RADIUS[GmatSolar↩ SystemDefaults::EARTH]

9.83.1 Macro Definition Documentation

9.83.1.1 EARTH_EQ_RADIUS

```
#define EARTH_EQ_RADIUS GmatSolarSystemDefaults::PLANET_EQUATORIAL_RADIUS[GmatSolarSystem↩
Defaults::EARTH]
```

9.83.1.2 EARTH_FLATTENING

```
#define EARTH_FLATTENING GmatSolarSystemDefaults::PLANET_FLATTENING[GmatSolarSystemDefaults↩
::EARTH]
```

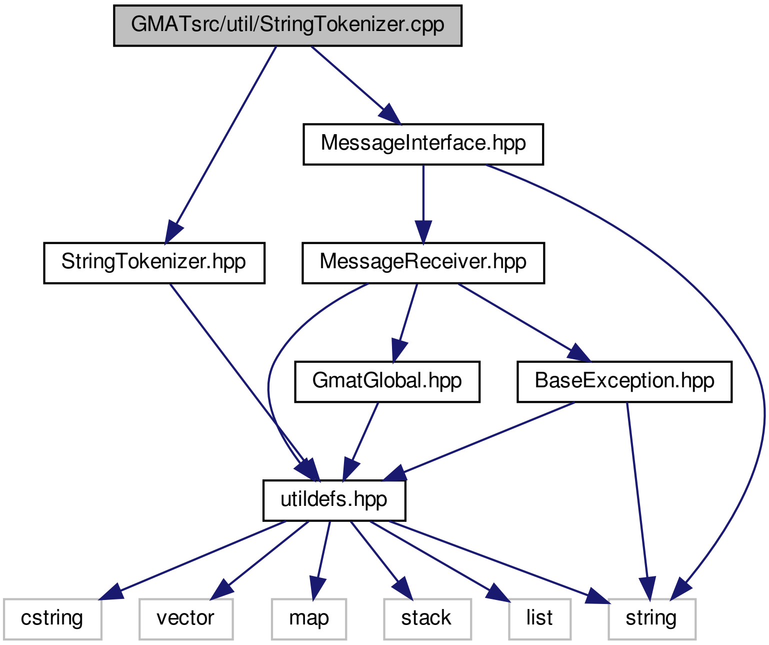
9.83.1.3 EARTH_MU

```
#define EARTH_MU GmatSolarSystemDefaults::PLANET_MU[GmatSolarSystemDefaults::EARTH]
```

9.84 GMATsrc/util/StringTokenizer.cpp File Reference

```
#include "StringTokenizer.hpp"
#include "MessageInterface.hpp"
```

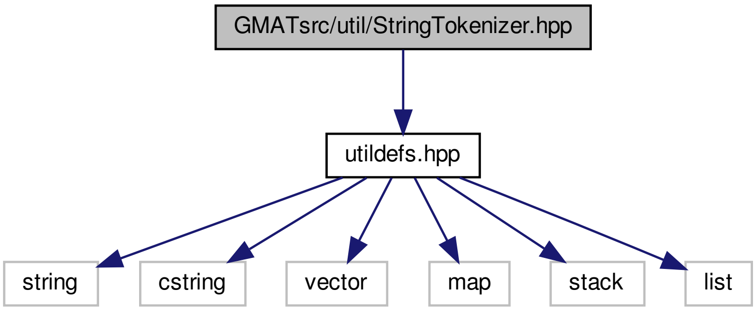
Include dependency graph for StringTokenizer.cpp:



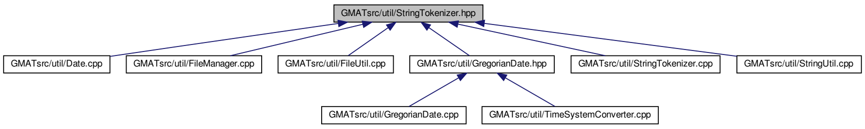
9.85 GMATsrc/util/StringTokenizer.hpp File Reference

```
#include "utildefs.hpp"
```

Include dependency graph for StringTokenizer.hpp:



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


GMATsrc/util/StringTokenizer.hpp
GMATsrc/util/Date.cpp
GMATsrc/util/FileManager.cpp
GMATsrc/util/FileUtil.cpp
GMATsrc/util/GregorianDate.hpp
GMATsrc/util/StringTokenizer.cpp
GMATsrc/util/StringUtil.cpp
GMATsrc/util/GregorianDate.cpp
GMATsrc/util/TimeSystemConverter.cpp

Classes

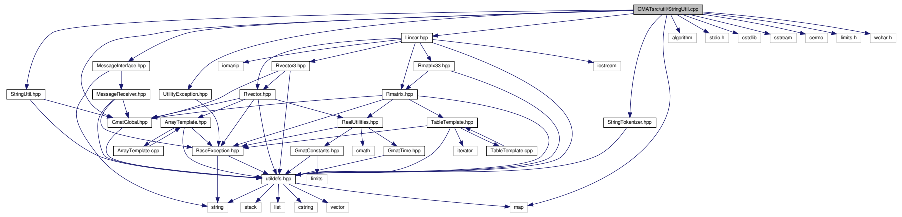
- class StringTokenizer

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 <wchar.h>
```

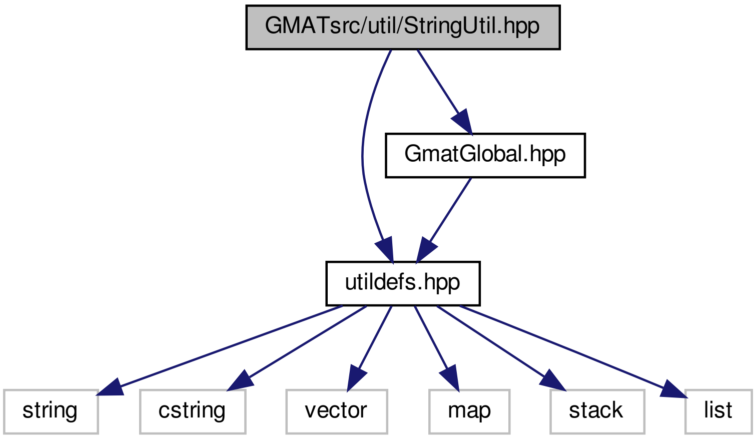
Include dependency graph for StringUtil.cpp:



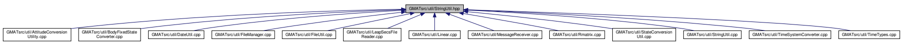
9.87 GMATsrc/util/StringUtil.hpp File Reference

```
#include "utildefs.hpp"
#include "GmatGlobal.hpp"
```

Include dependency graph for StringUtil.hpp:



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


GMATsrc/util/StringUtil.hpp
GMATsrc/util/AttitudeConversion Utility.cpp
GMATsrc/util/BodyFixedState Converter.cpp
GMATsrc/util/DateUtil.cpp
GMATsrc/util/FileManager.cpp
GMATsrc/util/FileUtil.cpp
GMATsrc/util/LeapSecsFile Reader.cpp
GMATsrc/util/Linear.cpp
GMATsrc/util/MessageReceiver.cpp
GMATsrc/util/Rmatrix.cpp
GMATsrc/util/StateConversion Util.cpp
GMATsrc/util/StringUtil.cpp
GMATsrc/util/TimeSystemConverter.cpp
GMATsrc/util/TimeTypes.cpp

Namespaces

- 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 startIndex=0)
- GMATUTIL_API std::string GmatStringUtil::Replace (const std::string &str, const std::string &from, const std::string &to, std::string::size_type startIndex=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 ignoreComma=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 &enStr)
- 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 checkSingleQuote=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 trimParens=false, bool allowOverflow=true)
- GMATUTIL_API bool GmatStringUtil::ToUnsignedInt (const std::string &str, UnsignedInt &value, bool trimParens=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 trimParens=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)

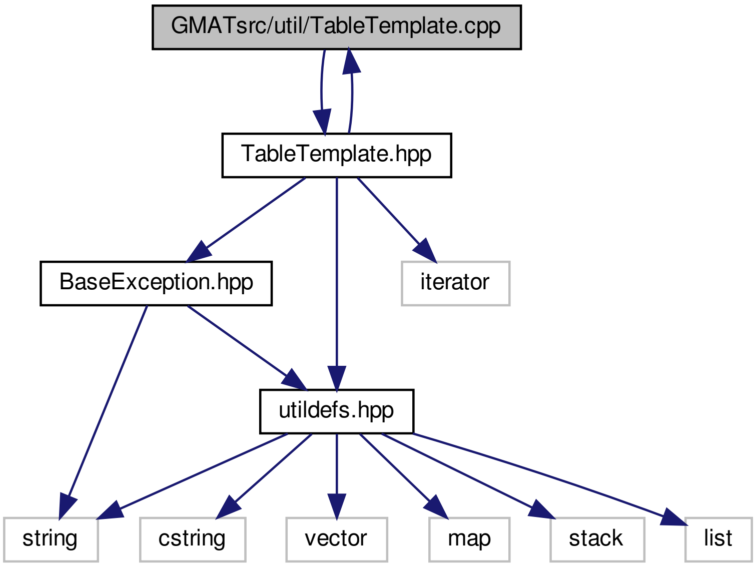
- GMATUTIL_API IntegerArray GmatStringUtil::ToIntegerArray (const std::string &str, bool allowOverflow=true)
- GMATUTIL_API UnsignedIntArray GmatStringUtil::ToUnsignedIntArray (const std::string &str, bool allow↩ Overflow=true)
- GMATUTIL_API StringArray GmatStringUtil::ToStringArray (const std::string &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 &bracket↩ Pair="()")
- 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 &bracket↩ Pair)
- 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 &all↩ Pairs)
- GMATUTIL_API bool GmatStringUtil::AreAllNamesValid (const std::string &str, bool blankNameIsOk=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 parensForArrays↩ Allowed=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 blank↩NameIsOk=false)
- GMATUTIL_API bool GmatStringUtil::IsValidName (const std::string &str, bool ignoreBracket=false, bool blankNameIsOk=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 blankNameIsOk=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)

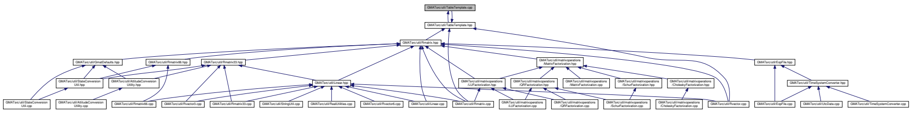
9.88 GMATsrc/util/TableTemplate.cpp File Reference

```
#include "TableTemplate.hpp"
```

Include dependency graph for TableTemplate.cpp:



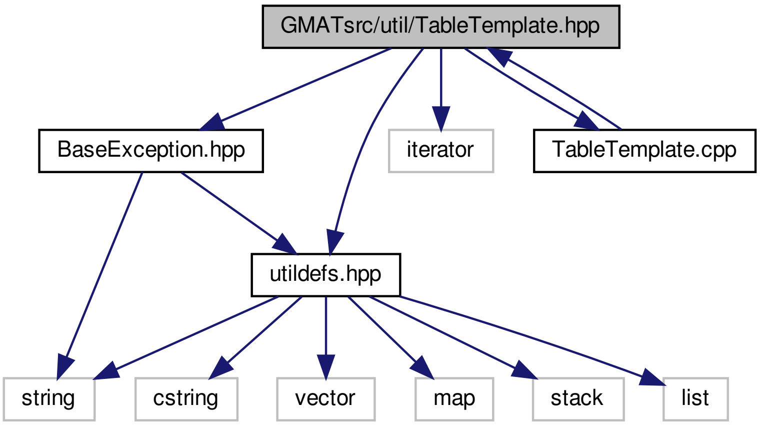
This graph shows which files directly or indirectly include this file:



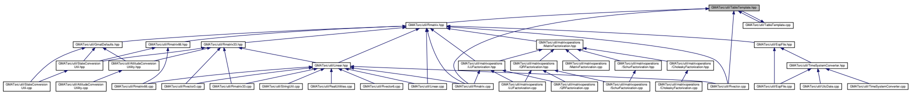
9.89 GMATsrc/util/TableTemplate.hpp File Reference

```
#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

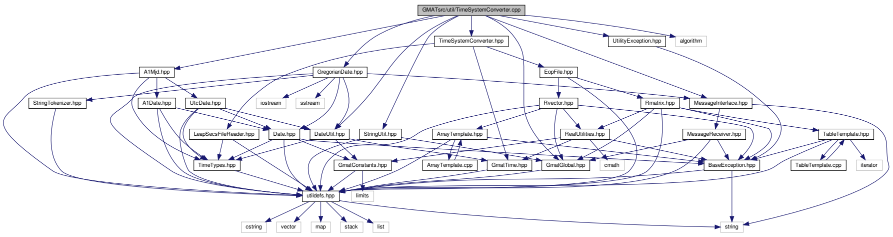
- class TableTemplateExceptions
- class TableTemplateExceptions::DimensionError
- class TableTemplateExceptions::OutOfBounds
- class TableTemplateExceptions::UnsizedTable
- class TableTemplateExceptions::TableAlreadySized
- class TableTemplateExceptions::IllegalSize
- class TableTemplate< T >

9.90 GMATsrc/util/TimeSystemConverter.cpp File Reference

```
#include "TimeSystemConverter.hpp"
#include "A1Mjd.hpp"
#include "GregorianDate.hpp"
#include "DateUtil.hpp"
#include "StringUtil.hpp"
#include "GmatGlobal.hpp"
#include "MessageInterface.hpp"
#include "UtilityException.hpp"
```

```
#include <algorithm>
```

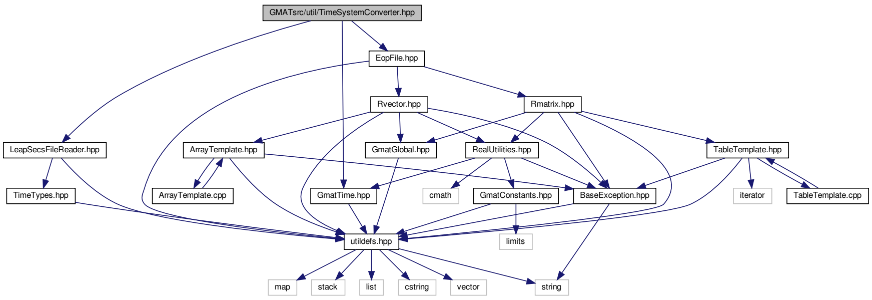
Include dependency graph for TimeSystemConverter.cpp:



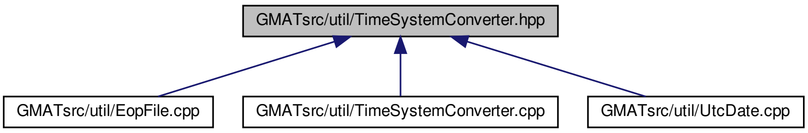
9.91 GMATsrc/util/TimeSystemConverter.hpp File Reference

```
#include "EopFile.hpp"
#include "LeapSecsFileReader.hpp"
#include "GmatTime.hpp"
```

Include dependency graph for TimeSystemConverter.hpp:



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


GMATsrc/util/TimeSystemConverter.hpp
GMATsrc/util/EopFile.cpp
GMATsrc/util/TimeSystemConverter.cpp
GMATsrc/util/UtcDate.cpp

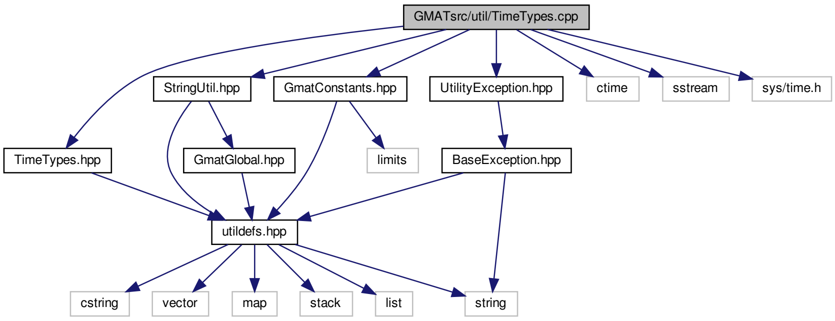
Classes

- 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>
```

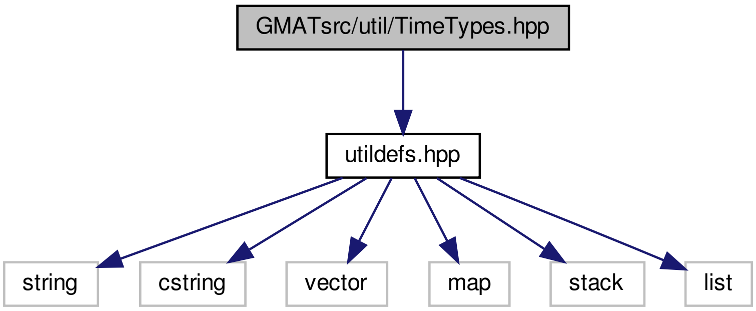
Include dependency graph for TimeTypes.cpp:



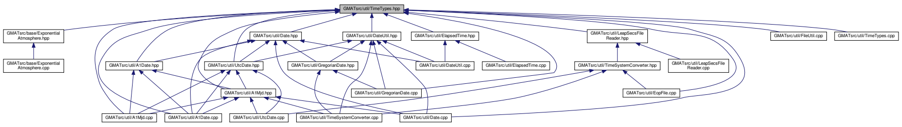
9.93 GMATsrc/util/TimeTypes.hpp File Reference

```
#include "utildefs.hpp"
```

Include dependency graph for TimeTypes.hpp:



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


GMATsrc/util/TimeTypes.hpp
GMATsrc/base/Exponential Atmosphere.hpp
GMATsrc/util/Date.hpp
GMATsrc/util/DateUtil.hpp
GMATsrc/util/ElapsedTime.hpp
GMATsrc/util/LeapSecsFile Reader.hpp
GMATsrc/util/FileUtil.cpp
GMATsrc/util/TimeTypes.cpp
GMATsrc/base/Exponential Atmosphere.cpp
GMATsrc/util/A1Date.hpp
GMATsrc/util/UtcDate.hpp
GMATsrc/util/GregorianDate.hpp
GMATsrc/util/DateUtil.cpp
GMATsrc/util/ElapsedTime.cpp
GMATsrc/util/TimeSystemConverter.hpp
GMATsrc/util/LeapSecsFile Reader.cpp
GMATsrc/util/A1Mjd.hpp
GMATsrc/util/GregorianDate.cpp
GMATsrc/util/EopFile.cpp
GMATsrc/util/A1Mjd.cpp
GMATsrc/util/A1Date.cpp
GMATsrc/util/UtcDate.cpp
GMATsrc/util/TimeSystemConverter.cpp
GMATsrc/util/Date.cpp

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.

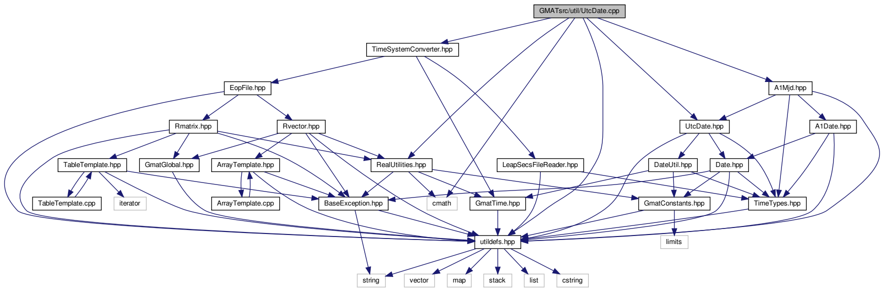
9.93.1.8 YearNumber

```
typedef Integer YearNumber
```

9.94 GMATsrc/util/UtcDate.cpp File Reference

```
#include <cmath>
#include "utildefs.hpp"
#include "RealUtilities.hpp"
#include "A1Mjd.hpp"
#include "UtcDate.hpp"
#include "TimeSystemConverter.hpp"
```

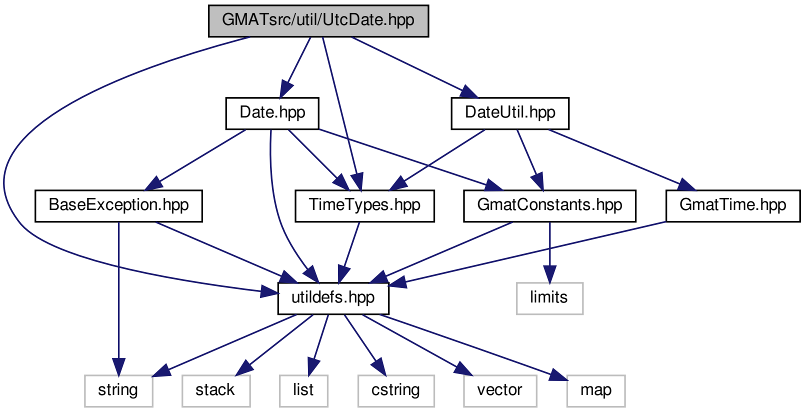
Include dependency graph for UtcDate.cpp:



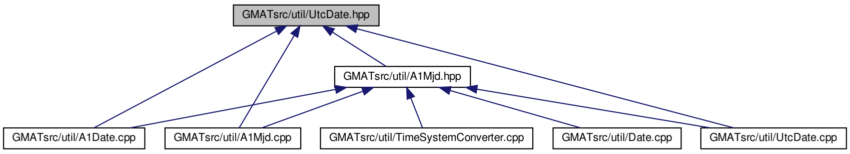
9.95 GMATsrc/util/UtcDate.hpp File Reference

```
#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:



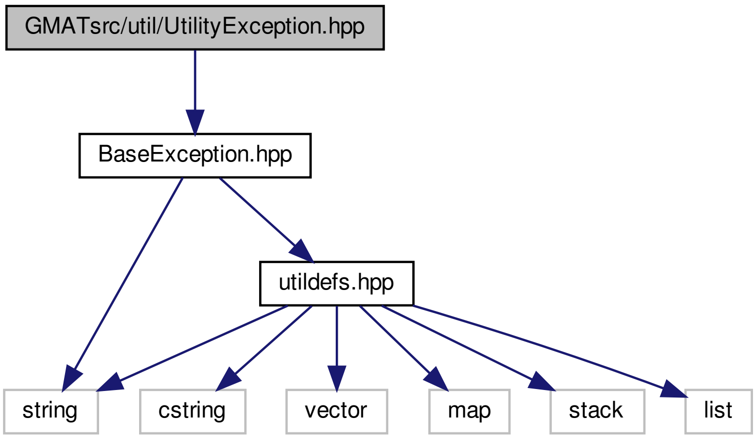
Classes

- class UtcDate

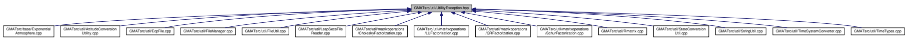
9.96 GMATsrc/util/UtilityException.hpp File Reference

```
#include "BaseException.hpp"
```

Include dependency graph for UtilityException.hpp:



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


GMATsrc/util/UtilityException.hpp
GMATsrc/base/Exponential Atmosphere.cpp
GMATsrc/util/AttitudeConversion Utility.cpp
GMATsrc/util/EopFile.cpp
GMATsrc/util/FileManager.cpp
GMATsrc/util/FileUtil.cpp
GMATsrc/util/LeapSecsFile Reader.cpp
GMATsrc/util/matrixoperations /CholeskyFactorization.cpp
GMATsrc/util/matrixoperations /LUFactorization.cpp
GMATsrc/util/matrixoperations /QRFactorization.cpp
GMATsrc/util/matrixoperations /SchurFactorization.cpp
GMATsrc/util/Rmatrix.cpp
GMATsrc/util/StateConversion Util.cpp
GMATsrc/util/StringUtil.cpp
GMATsrc/util/TimeSystemConverter.cpp
GMATsrc/util/TimeTypes.cpp

Classes

- class UtilityException
- class GravityFileException
- class TimeException

Index