REUSE MANUAL

GEOCENTRIC

10xxxxxx.1

Implementation

TABLE OF CONTENTS

TABLE OF CONTENTS I

SECTION 1. INTRODUCTION 1

1.1 PURPOSE OF THE REUSE MANUAL 1

1.2 PURPOSE OF THE REUSABLE SOFTWARE COMPONENT 1

1.3 GENERAL INFORMATION 1

1.3.1 POINT OF CONTACT 1

1.3.2 CERTIFICATION LEVEL 2

1.3.3 LEGAL RESTRICTIONS 2

SECTION 2. INSTALLATION 3

2.1 PARTIAL REUSE 3

2.2 MODIFICATIONS 3

SECTION 3. ENVIRONMENT 4

3.1 HARDWARE 4

3.1.1 DEVELOPMENT 4

3.1.2 TARGET 4

3.2 SOFTWARE 4

3.2.1 OPERATING SYSTEM 4

3.2.2 COMPILERS 4

3.3 ASSUMPTIONS AND PERFORMANCE LIMITATIONS 5

SECTION 4. GLOBAL RSC ENVIRONMENT 6

4.1 TYPES 6

4.2 CONSTANTS 6

4.3 VARIABLES 6

4.4 INCLUDE FILES 6

4.5 DEPENDENCIES 6

SECTION 5. FUNCTIONS 7

5.1 SET_GEOCENTRIC_PARAMETERS 7

5.2 GET_GEOCENTRIC_PARAMETERS 8

5.3 CONVERT_GEODETIC_TO_GEOCENTRIC 9
5.4 CONVERT_GEOCENTRIC_TO_GEODETIC 11

APPENDIX A STRUCTURE/DEPENDENCY DIAGRAMS 13

APPENDIX B DEFINITIONS/GLOSSARY 14

APPENDIX C REFERENCES 15

SECTION 1. INTRODUCTION

1.1 PURPOSE OF THE REUSE MANUAL

This document describes the characteristics of the GEOCENTRIC reusable software component and provides instructions on its installation and operation. The manual is a self-contained reference for the software engineer intending to incorporate the component in another software system. This manual was written with the assumption that the user has a basic working knowledge of C and is familiar with fundamental C concepts and terminology.

1.2 PURPOSE OF THE REUSABLE SOFTWARE COMPONENT

The purpose of GEOCENTRIC is to provide a reusable software component which supports the following coordinate conversions :

- Geodetic coordinates (latitude, longitude in radians and height in meters) to Geocentric coordinates (X, Y, Z) in meters, and
- Geocentric coordinates (X, Y, Z) in meters to Geodetic coordinates (latitude, longitude in radians and height in meters).

A particular ellipsoid is specified in terms of the following parameters:

- Semi-Major Axis (a) – Radius (in meters) at the equator.
- Semi-Minor Axis (b) – Radius (in meters) at a pole.

1.3 GENERAL INFORMATION

1.3.1 POINT OF CONTACT

U.S. Army Topographic Engineering Center (USATEC)
Geospatial Information Division
ATTN: CETEC-GD-A (Dan Specht)
7701 Telegraph Road
Alexandria, VA 22315-3864
Dan Specht (703) 428 - 6761 Project Manager

1.3.2 CERTIFICATION LEVEL

This RSC has been certified at level 4. A level 4 component satisfies the criteria for reliability, testing, and documentation for the Army Reuse Center (ARC). The component comes with test materials and a Reuse Manual that aids in integrating the component into a software system.

1.3.3 LEGAL RESTRICTIONS

This Reusable Software Component (RSC) contains data with Unlimited Government Rights.

SECTION 2. INSTALLATION

The following is a list of the files which make up the GEOCENTRIC component:

Source Code Files:

```
geocent.c
```

Header Files :

```
geocent.h
```

Data Files :

```
none
```

The compilation instructions for the GEOCENTRIC component are as follows:

DOS Makefile (Uses Microsoft C):

```
cl /nologo /W3 /FR /G2 /DNDEBUG /Gs /Ox /AM /D_DOS /c geocent.c
```

UNIX Makefile (Uses gcc compiler):

```
cc -g -O -ansi -Wall -c geocent.c
```

The compilation order of the GEOCENTRIC component relative to other components is unconstrained.

2.1 PARTIAL REUSE

The GEOCENTRIC component does not allow for partial reuse.

2.2 MODIFICATIONS

The GEOCENTRIC component does not permit modifications.

SECTION 3. ENVIRONMENT

This section provides details on the environment under which GEOCENTRIC was developed, tested, and executed.

3.1 HARDWARE

3.1.1 DEVELOPMENT

The following is a list of hardware configurations under which GEOCENTRIC was developed and tested.

- SUN SparcStation 20
- IBM compatible Pentium PC

3.1.2 TARGET

The following is a list of hardware configurations under which GEOCENTRIC was executed.

- SUN SparcStation 20
- IBM compatible Pentium PC

3.2 SOFTWARE

3.2.1 OPERATING SYSTEM

The following is a list of operating systems under which GEOCENTRIC was executed and tested.

- Solaris 2.5
- Windows 95

3.2.2 COMPILERS

The following is a list of compilers on which GEOCENTRIC was compiled successfully.

- GCC version 2.8.1
- Microsoft Visual C++ version 6

3.3 ASSUMPTIONS AND PERFORMANCE LIMITATIONS

There are no hardware or environment constraints. There are no limitations.

This RSC is written in ANSI C.

SECTION 4. GLOBAL RSC ENVIRONMENT

4.1 TYPES

Not applicable.

4.2 CONSTANTS

The following is a list of significant visible constants declared globally in GEOCENTRIC with their descriptions.

```
GEOCENT_NO_ERROR                : No errors occurred in function
GEOCENT_LAT_ERROR               : Latitude outside of valid range
                                    (-90 to 90 degrees)
GEOCENT_LON_ERROR               : Longitude outside of valid range
                                    (-180 to 360 degrees)
GEOCENT_A_ERROR                 : Semi-major axis less than or equal to zero
GEOCENT_B_ERROR                 : Semi-minor axis less than or equal to zero
GEOCENT_A_LESS_B_ERROR          : Semi-major axis less than semi-minor axis
```

4.3 VARIABLES

The following is a list of significant global variables declared in GEOCENTRIC with their descriptions.

```
Ellipsoid Parameters:
double Geocent_a                : Semi-major axis of ellipsoid in meters
double Geocent_b                : Semi-minor axis of ellipsoid in meters

Optimization variables:
double Geocent_a2               : Semi-major axis squared
double Geocent_b2               : Semi-minor axis squared
double Geocent_e2               : Eccentricity of ellipsoid squared
double Geocent_ep2              : Eccentricity prime squared
```

4.4 INCLUDE FILES

```
math.h                          : Used to call standard math functions
geocent.h                       : Error codes and prototype error checking
```

4.5 DEPENDENCIES

None, other than the standard ANSI C math library.

SECTION 5. FUNCTIONS

5.1 SET_GEOCENTRIC_PARAMETERS

5.1.1 DESCRIPTION

This function sets the ellipsoid parameters to the specified values.

5.1.2 INTERFACES AND EXAMPLES

The following is a list of the formal arguments required to use this function.

```
long Set_Geocentric_Parameters (double a,
             double b);
```

a Semi-major axis of ellipsoid in meters (input),

b Semi-minor axis of ellipsoid in meters (input),

Example:

```
status = Set_Geocentric_Parameters (a, b)
```

Inputs:

a 6378137.0

b 6356752.3142

Outputs:

none

5.1.3 DECLARATIONS

5.1.3.1 TYPES

Not applicable.

5.1.3.2 CONSTANTS

Not applicable.

5.1.3.3 VARIABLES

Not applicable.

5.1.4 DEPENDENCIES

Not applicable.

5.1.5 ERROR HANDLING

This function returns the following status codes:

```
GEOCENT_NO_ERROR                : No errors occurred in function
GEOCENT_A_ERROR                 : Semi-major axis less than or equal to zero
GEOCENT_B_ERROR                 : Semi-minor axis less than or equal to zero
GEOCENT_A_LESS_B_ERROR          : Semi-major axis less than semi-minor axis
```

5.2 GET_GEOCENTRIC_PARAMETERS

5.2.1 DESCRIPTION

This function returns the current values of the ellipsoid parameters.

5.2.2 INTERFACES AND EXAMPLES

The following is a list of the formal arguments required to use this function.

```
void Get_Geocentric_Parameters (double *a,
             double *b);
```

a Semi-major axis of ellipsoid in meters (output),

b Semi-minor axis of ellipsoid in meters (output),

5.2.3 DECLARATIONS

5.2.3.1 TYPES

Not applicable.

5.2.3.2 CONSTANTS

Not applicable.

5.2.3.3 VARIABLES

Not applicable.

5.2.4 DEPENDENCIES

None.

5.2.5 ERROR HANDLING

No errors are reported by this function.

5.3 CONVERT_GEODETIC_TO_GEOCENTRIC

5.3.1 DESCRIPTION

This function converts Geodetic coordinates (latitude, longitude in radians and height in meters) to Geocentric coordinates (X, Y, Z in meters), using the current ellipsoid parameters.

5.3.2 INTERFACES AND EXAMPLES

The following is a list of the formal arguments required to use this function.

```
long Convert_Geodetic_To_Geocentric (double Latitude,
              double Longitude,
              double Height,
              double *X,
              double *Y,
              double *Z);
```

Latitude	Latitude in radians (input),
Longitude	Longitude in radians (input),
Height	Ellipsoid height in meters (input),
X	X coordinate in meters (output),
Y	Y coordinate in meters (output),

Z Z coordinate in meters (output).

Example:

```
status = Convert_Geodetic_To_Geocentric (Latitude, Longitude, Height, X, Y, Z)
```

Inputs:

Latitude: 45.0

Longitude: -75.0

Height: 700

Outputs:

X: 1169366.666323

Y: -4364135.811395

Z: 4487843.383565

5.3.3 DECLARATIONS

5.3.3.1 TYPES

Not applicable.

5.3.3.2 CONSTANTS

Not applicable.

5.3.3.3 VARIABLES

Not applicable.

5.3.4 DEPENDENCIES

None.

5.3.5 ERROR HANDLING

This function returns the following status codes:

```
GEOCENT_NO_ERROR                : No errors occurred in function
GEOCENT_LAT_ERROR               : Latitude outside of valid range
                                    (-90 to 90 degrees)
GEOCENT_LON_ERROR               : Longitude outside of valid range
                                    (-180 to 360 degrees)
```

5.4 CONVERT_GEOCENTRIC_TO_GEODETIC

5.4.1 DESCRIPTION

This function converts Geocentric coordinates (X, Y, Z in meters) to Geodetic coordinates (latitude, longitude in radians and height in meters), using the current ellipsoid parameters.

5.4.2 INTERFACES AND EXAMPLES

The following is a list of the formal arguments required to use this function.

```
void Convert_Geocentric_To_Geodetic (double X,
             double Y,
             double Z,
             double *Latitude,
             double *Longitude,
             double *Height);
```

X	X geocentric coordinate in meters (input),
Y	Y geocentric coordinate in meters (input),
Z	Z geocentric coordinate in meters (input),
Latitude	Latitude in radians (output),
Longitude	Longitude in radians (output),
Height	Ellipsoid height in meters (output).

Example:

```
status = Convert_Geocentric_To_Geodetic (X, Y, Z, Latitude, Longitude, Height)
```

Inputs:

X:	1169366.666323
Y:	-4364135.811395
Z:	4487843.383565

Outputs:

Latitude:	45.0
Longitude:	-75.0
Height:	700.004645

5.4.3 DECLARATIONS

5.4.3.1 TYPES

Not applicable.

5.4.3.2 CONSTANTS

Not applicable.

5.4.3.3 VARIABLES

Not applicable.

5.4.4 DEPENDENCIES

None.

5.4.5 ERROR HANDLING

No errors are reported by this function.

APPENDIX A STRUCTURE/DEPENDENCY DIAGRAMS

This component consists of a single compilation unit and depends only on the ANSI C standard math library.

APPENDIX B DEFINITIONS/GLOSSARY

Central Meridian – Longitude at the horizontal center of a projection; Origin Longitude.

Coordinate – Linear or angular quantities that designate the position that a point occupies in a given reference frame or system. Also used as a general term to designate the particular kind of reference frame or system, such as Cartesian coordinates or spherical coordinates.

Ellipsoid – The surface generated by an ellipse rotating about one of its axes.

Geocentric Coordinates – Cartesian coordinates (X, Y, Z) that define the position of a point with respect to the center of mass of the earth.

Geodetic Coordinates – The quantities of latitude and longitude that define the position of a point on the surface of the earth with respect to the reference ellipsoid. Also, imprecisely called geographic coordinates.

Geodetic Latitude – The angle between the plane of the equator and the normal to the ellipsoid through the computation point. Geodetic latitude is positive north of the equator and negative south of the equator.

Geodetic Longitude – The angle between the plane of a meridian and the plane of the prime meridian. A longitude can be measured from the angle formed between the local and prime meridians at the pole of rotation of the reference ellipsoid, or by the arc along the equator intercepted by these meridians.

APPENDIX C REFERENCES

(1) Topographic Engineering Center, TEC-SR-7, **Handbook for transformation of DATUMS, PROJECTIONS, GRIDS, AND COMMON COORDINATE SYSTEMS**, January 1996.

(2) An Improved Algorithm for Geocentric to Geodetic Coordinate Conversion. Ralph Toms, February 1996 UCRL-JC-123138.