REUSE MANUAL

MILLER

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 2

1.3.1 POINT OF CONTACT 2

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 7

SECTION 5. FUNCTIONS 8

5.1 SET_MILLER_PARAMETERS 8

5.2 GET_MILLER_PARAMETERS 9

5.3 CONVERT_GEODETIC_TO_MILLER 11
5.4 CONVERT_MILLER_TO_GEODETIC 12

APPENDIX A STRUCTURE/DEPENDENCY DIAGRAMS 15

APPENDIX B DEFINITIONS/GLOSSARY 16

APPENDIX C REFERENCES 18

SECTION 1. INTRODUCTION

1.1 PURPOSE OF THE REUSE MANUAL

This document describes the characteristics of the MILLER 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 MILLER is to provide a reusable software component which supports the following coordinate conversions :

- Geodetic coordinates (latitude and longitude in radians) to Miller Cylindrical projection coordinates (easting and northing in meters), and
- Miller Cylindrical projection coordinates (easting and northing in meters) to Geodetic coordinates (latitude and longitude in radians).

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.

A particular variation of the Miller Cylindrical projection is specified in terms of the following parameters:

- Central Meridian – Longitude (in radians) at the horizontal center of the projection,
- False Easting – A coordinate value (in meters) assigned to the central meridian of the projection to avoid the inconvenience of using negative coordinates,
- False Northing – A coordinate value (in meters) assigned to the origin latitude of the projection to avoid the inconvenience of using negative coordinates.

1.3 GENERAL INFORMATION

1.3.1 POINT OF CONTACT

U.S. Army Topographic Engineering Center (USATEC)
Geospatial Information Division (GID)
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

SECTION 2. INSTALLATION

The following is a list of the files that make up the MILLER component:

Source Code Files:

```
miller.c
```

Header Files :

```
miller.h
```

Data Files :

```
none
```

The compilation instructions for the MILLER component are as follows:

DOS Makefile (Uses Microsoft C):

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

UNIX Makefile (Uses gcc compiler):

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

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

2.1 PARTIAL REUSE

The MILLER component does not allow for partial reuse.

2.2 MODIFICATIONS

The MILLER component does not permit modifications.

SECTION 3. ENVIRONMENT

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

3.1 HARDWARE

3.1.1 DEVELOPMENT

The following is a list of hardware configurations under which MILLER 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 MILLER 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 MILLER was executed and tested.

- Solaris 2.5
- Windows 95

3.2.2 COMPILERS

The following is a list of compilers on which MILLER 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 MILLER with their descriptions.

```
MILL_NO_ERROR                   : No errors occurred in function
MILL_LAT_ERROR                  : Latitude outside of valid range
                                  (-90 to 90 degrees)
MILL_LON_ERROR                  : Longitude outside of valid range
                                  (-180 to 360 degrees)
MILL_EASTING_ERROR              : Easting outside of valid range
                                  (False_Easting +/- ~20,000,000 m,
                                  depending on ellipsoid parameters
                                  and Origin_Latitude)
MILL_NORTHING_ERROR             : Northing outside of valid range
                                  (False_Northing +/- ~14,500,000 m,
                                  depending on ellipsoid parameters
                                  and Origin_Latitude)
MILL_CENT_MER_ERROR             : Central meridian outside of valid range
                                  (-180 to 360 degrees)
MILL_A_ERROR                    : Semi-major axis less than or equal to zero
MILL_B_ERROR                    : Semi-minor axis less than or equal to zero
MILL_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 MILLER with their descriptions.

```
Ellipsoid Parameters:
static double Mill_a            : Semi-major axis of ellipsoid in meters
static double Mill_b            : Semi-minor axis of ellipsoid in meters

Projection Parameters:
static double Mill_Origin_Lat    : Latitude of origin in radians
static double Mill_Origin_Long   : Longitude of origin in radians
static double Mill_False_Easting : False easting in meters
static double Mill_False_Northing   : False northing in meters

Maximum easting and northing values (for default ellipsoid and projection):
static double Mill_Delta_Easting
static double Mill_Delta_Northing
```

4.4 INCLUDE FILES

```
math.h                          : Standard C math library
miller.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_MILLER_PARAMETERS

5.1.1 DESCRIPTION

This function sets the ellipsoid and Miller Cylindrical projection 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_Miller_Parameters (double a,
             double b,
             double Central_Meridian,
             double False_Easting,
             double False_Northing);
```

a	Semi-major axis of ellipsoid in meters (input),
b	Semi-minor axis of ellipsoid in meters (input),
Central_Meridian	Longitude in radians at the center of the projection (input),
False_Easting	Coordinate value in meters assigned to the central meridian (input),
False_Northing	Coordinate value in meters assigned to the origin latitude (input),

Example:

```
status = Set_Miller_Parameters (a, b, Central_Meridian, False_Easting,
            False_Northing)
```

Inputs:

a	6378137.0
b	6356752.3142
Central_Meridian	0.0
False_Easting	0.0
False_Northing	0.0

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

Convert_Miller_To_Geodetic – used to determine maximum valid easting and northing values for current ellipsoid and Miller Cylindrical projection parameters.

5.1.5 ERROR HANDLING

This function returns the following status codes:

```
MILL_NO_ERROR                   : No errors occurred in function
MILL_CENT_MER_ERROR             : Central meridian outside of valid range
                                  (-180 to 360 degrees)
MILL_A_ERROR                    : Semi-major axis less than or equal to zero
MILL_B_ERROR                    : Semi-minor axis less than or equal to zero
MILL_A_LESS_B_ERROR             : Semi-major axis less than semi-minor axis
```

5.2 GET_MILLER_PARAMETERS

5.2.1 DESCRIPTION

This function returns the current values of the ellipsoid parameters, and Miller Cylindrical projection parameters.

5.2.2 INTERFACES AND EXAMPLES

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

```
void Get_Miller_Parameters (double *a,
             double *b,
             double *Central_Meridian,
             double *False_Easting,
             double *False_Northing);
```

a	Semi-major axis of ellipsoid in meters (output),
b	Semi-minor axis of ellipsoid in meters (output),
Central_Meridian	Longitude in radians at the center of the projection (output),
False_Easting	Coordinate value in meters assigned to the central meridian (output),
False_Northing	Coordinate value in meters assigned to the origin latitude (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_MILLER

5.3.1 DESCRIPTION

This function converts Geodetic coordinates (latitude and longitude in radians) to Miller Cylindrical projection coordinates (easting and northing in meters), using the current ellipsoid and Miller Cylindrical projection 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_Miller (double Latitude,
             double Longitude,
             double *Easting,
             double *Northing);
```

Latitude	Latitude in radians (input),
Longitude	Longitude in radians (input),
Easting	Easting (X) in meters (output),
Northing	Northing (Y) in meters (output).

Example:

```
status = Convert_Geodetic_To_Miller (Latitude, Longitude, Easting, Northing)
```

Inputs:

Latitude:	-35.0
Longitude:	75.0

Outputs:

Easting:	-2335096.09
Northing:	-4056677.39

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:

```
MILL_NO_ERROR                   : No errors occurred in function
MILL_LAT_ERROR                  : Latitude outside of valid range
                                  (-90 to 90 degrees)
MILL_LON_ERROR                  : Longitude outside of valid range
                                  (-180 to 360 degrees)
```

5.4 CONVERT_MILLER_TO_GEODETIC

5.4.1 DESCRIPTION

This function converts Miller Cylindrical projection coordinates (easting and northing in meters) to Geodetic coordinates (latitude and longitude in radians), using the current ellipsoid and Miller Cylindrical projection parameters.

5.4.2 INTERFACES AND EXAMPLES

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

```
long Convert_Miller_To_Geodetic (double Easting,
             double Northing,
             double *Latitude,
             double *Longitude);
```

Easting	Easting (X) in meters (input),
Northing	Northing (Y) in meters (input),
Latitude	Latitude in radians (output),
Longitude	Longitude in radians (output).

Example:

```
status = Convert_Miller_To_Geodetic (Easting, Northing, Latitude, Longitude)
```

Inputs:

Easting:	-2335096.09
Northing:	-4056677.39

Outputs:

Latitude:	-35.0
Longitude:	75.0

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

This function returns the following status codes:

```
MILL_NO_ERROR                   : No errors occurred in function
MILL_EASTING_ERROR              : Easting outside of valid range
                                  (False_Easting +/- ~20,000,000 m,
                                  depending on ellipsoid parameters
                                  and Origin_Latitude)
MILL_NORTHING_ERROR             : Northing outside of valid range
                                  (False_Northing +/- ~14,000,000 m,
                                  depending on ellipsoid parameters
                                  and Origin_Latitude)
```

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 or spherical coordinates.

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

False Easting – A coordinate value (in meters) assigned to the central meridian of the projection to avoid the inconvenience of using negative coordinates.

False Northing – A coordinate value (in meters) assigned to the origin latitude of the projection to avoid the inconvenience of using negative coordinates.

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.

Map Projection – A function relating coordinates of points on a curved surface (usually an ellipsoid or sphere) to coordinates of points on a plane. A map projection may be established by analytical computation or, less commonly, may be constructed geometrically.

Map Scale – The ratio between a distance on a map and the corresponding actual distance on the earth's surface.

Meridian – A north-south reference line, particularly a great circle through the geographical poles of the earth, from which longitudes and azimuths are determined; or the intersection of a plane forming a great circle that contains both geographic poles of the earth, and the ellipsoid.

Miller Projection – A cylindrical projection used only in *spherical* (not ellipsoidal) form. Meridians and parallels are straight lines intersecting at right angles, where meridians are equidistant and parallels are spaced farther apart away from the Equator.

Origin Latitude – Latitude at which the scale factor of the projection is 1.0.

Parallel – A line on the earth, or a representation thereof, that represents the same latitude at every point.

Scale Factor (Projection) – A multiplier for reducing a distance in a map projection to the actual distance on the chosen reference ellipsoid.

APPENDIX C REFERENCES

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

(2) Snyder, J. P., **Geological Survey Professional Paper 1395 Map Projections - A Working Manual**, 1987.