REUSE MANUAL

LAMBERT

10xxxxxx.1

Implementation

TABLE OF CONTENTS

SECTION 1. INTRODUCTION

1.1 PURPOSE OF THE REUSE MANUAL

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

- Geodetic coordinates (latitude and longitude in radians) to Lambert Conformal Conic projection coordinates (easting and northing in meters) with two standard parallels,
- Lambert Conformal Conic projection coordinates (easting and northing in meters) with two standard parallels 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, and
- Semi-Minor Axis (b): Radius (in meters) at a pole.

A particular variation of the Lambert Conformal Conic projection with one or two standard parallels is specified in terms of the following parameters:

- Central Meridian – Longitude (in radians) at the origin of the projection coordinate system,
- Origin Latitude – Latitude (in radians) at the origin of the projection coordinate system,
- 1st Standard Parallel – Latitude (in radians) of the first of the two standard parallels, where the point scale factor is 1.0,
- 2nd Standard Parallel – Latitude (in radians) of the second of the two standard parallels,
- False Easting – A coordinate value (in meters) assigned to the origin longitude of the projection to avoid the inconvenience of using negative coordinates, and

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

Note that when the two standard parallels are equal to each other, the result is equivalent to a Lambert Conformal Conic projection with one standard parallel.

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 compilation files for the LAMBERT component:

Source Code Files:

```
lambert.c
```

Header Files :

```
lambert.h
```

Data Files :

```
none
```

The compilation instructions for the LAMBERT component are as follows:

DOS Makefile (Uses Microsoft C):

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

UNIX Makefile (Uses gcc compiler):

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

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

2.1 PARTIAL REUSE

The LAMBERT component does not allow for partial reuse.

2.2 MODIFICATIONS

The LAMBERT component does not permit modifications.

SECTION 3. ENVIRONMENT

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

3.1 HARDWARE

3.1.1 DEVELOPMENT

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

- Solaris 2.5
- Windows 95

3.2.2 COMPILERS

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

```
LAMBERT_NO_ERROR                : No errors occurred in function
LAMBERT_LAT_ERROR               : Latitude outside of valid range
                                   (-90 to 90 degrees)
LAMBERT_LON_ERROR               : Longitude outside of valid range
                                   (-180 to 360 degrees)
LAMBERT_EASTING_ERROR           : Easting outside of valid range
                                   (depends on ellipsoid and projection
                                    parameters)
LAMBERT_NORTHING_ERROR          : Northing outside of valid range
                                   (depends on ellipsoid and projection
                                    parameters)
LAMBERT_FIRST_STDP_ERROR        : First standard parallel outside of valid
                                   range (-89 to 89 degrees)
LAMBERT_SECOND_STDP_ERROR       : Second standard parallel outside of valid
                                   range (-89 to 89 degrees)
LAMBERT_ORIGIN_LAT_ERROR        : Origin latitude outside of valid range
                                   (-90 to 90)
LAMBERT_CENT_MER_ERROR          : Central meridian outside of valid range
                                   (-180 to 360 degrees)
LAMBERT_A_ERROR                 : Semi-major axis less than or equal to zero
LAMBERT_B_ERROR                 : Semi-minor axis less than or equal to zero
LAMBERT_A_LESS_B_ERROR          : Semi-major axis less than semi-minor axis
LAMBERT_HEMISPHERE_ERROR        : Standard parallels cannot be opposite
                                   latitudes
LAMBERT_FIRST_SECOND_ERROR      : The 1st & 2nd standard parallels cannot
                                   both be 0
```

4.3 VARIABLES

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

```
Ellipsoid Parameters:
static double Lambert_a             : Semi-major axis of ellipsoid in meters
static double Lambert_b             : Semi-minor axis of ellipsoid in meters

Projection Parameters:
static double Lambert_Std_Parallel_1 : Lower standard parallel latitude in
radians
static double Lambert_Std_Parallel_2 : Upper standard parallel latitude in
radians
static double Lambert_Origin_Lat     : Latitude of origin in radians
```

```
static double Lambert_Origin_Long    : Longitude of origin in radians
static double Lambert_False_Easting  : False easting in meters
static double Lambert_False_Northing : False northing in meters

Maximum variance for easting and northing values:
static double Lambert_Delta_Easting
static double Lambert_Delta_Northing
```

4.4 INCLUDE FILES

```
math.h                          : Standard C math library
lambert.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_LAMBERT_PARAMETERS

5.1.1 DESCRIPTION

This function sets the ellipsoid and Lambert Conformal Conic 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_Lambert_Parameters (double a,
                             double b,
                             double Origin_Latitude,
                             double Central_Meridian,
                             double Std_Parallel_1,
                             double Std_Parallel_2,
                             double False_Easting,
                             double False_Northing);
```

a	Semi-major axis of ellipsoid in meters (input),
b	Semi-minor axis of ellipsoid in meters (input),
Origin_Latitude	Latitude in radians at the origin of the projection (input),
Central_Meridian	Longitude in radians at the origin of the projection (input),
Std_Parallel_1	Latitude in radians of the lower standard parallel (input),
Std_Parallel_2	Latitude in radians of the upper standard parallel (input),
False_Easting	Coordinate value in meters assigned to the origin longitude (input),
False_Northing	Coordinate value in meters assigned to the origin latitude (input).

Example:

```
status = Set_Lambert_Parameters (a, b, Origin_Latitude, Central_Meridian,
           Std_Parallel_1, Std_Parallel_2, False_Easting, False_Northing)
```

Inputs:

a	6378137.0

b	6356752.3142
Origin_Latitude	45.0
Central_Meridian	0.0
Std_Parallel_1	40.0
Std_Parallel_2	50.0
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

None.

5.1.5 ERROR HANDLING

This function returns the following status codes:

```
LAMBERT_NO_ERROR                : No errors occurred in function
LAMBERT_FIRST_STDP_ERROR        : First standard parallel outside of valid
```

```
                                   range (-89 to 89 degrees)
LAMBERT_SECOND_STDP_ERROR        : Second standard parallel outside of valid
                                   range (-89 to 89 degrees)
LAMBERT_ORIGIN_LAT_ERROR         : Origin latitude outside of valid range
                                   (-90 to 90 degrees)
LAMBERT_CENT_MER_ERROR           : Origin longitude outside of valid range
                                   (-180 to 360 degrees)
LAMBERT_A_ERROR                  : Semi-major axis less than or equal to zero
LAMBERT_B_ERROR                  : Semi-minor axis less than or equal to zero
LAMBERT_A_LESS_B_ERROR           : Semi-major axis less than semi-minor axis
LAMBERT_HEMISPHERE_ERROR         : Standard parallels cannot be opposite
                                   Latitudes
LAMBERT_FIRST_SECOND_ERROR       : The 1st & 2nd standard parallels cannot
                                   both be 0
```

5.2 GET_LAMBERT_PARAMETERS

5.2.1 DESCRIPTION

This function returns the current values of the ellipsoid parameters and Lambert Conformal Conic projection parameters.

5.2.2 INTERFACES AND EXAMPLES

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

```
void Get_Lambert_Parameters (double *a,
                             double *b,
                             double *Origin_Latitude,
                             double *Central_Meridian,
                             double *Std_Parallel_1,
                             double *Std_Parallel_2,
                             double *False_Easting,
                             double *False_Northing);
```

a	Semi-major axis of ellipsoid in meters (output),
b	Semi-minor axis of ellipsoid in meters (output),
Origin_Latitude	Latitude in radians at the origin of the projection (output),
Central_Meridian	Longitude in radians at the origin of the projection (output),
Std_Parallel_1	Latitude in radians of the lower standard parallel (output),
Std_Parallel_2	Latitude in radians of the upper standard parallel (output),
False_Easting	Coordinate value in meters assigned to the origin longitude (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_LAMBERT

5.3.1 DESCRIPTION

This function converts Geodetic coordinates (latitude and longitude in radians) to Lambert Conformal Conic projection coordinates (easting and northing in meters), using the current ellipsoid and Lambert Conformal Conic 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_Lambert (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_Lambert (Latitude, Longitude, Easting, Northing)
```

Inputs:

Latitude: 35.0

Longitude: 75.0

Outputs:

Easting: 5972201

Northing: 1872576

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:

```
LAMBERT_NO_ERROR                : No errors occurred in function
LAMBERT_LAT_ERROR               : Latitude outside of valid range
                                    (-90 to 90 degrees)
LAMBERT_LON_ERROR               : Longitude outside of valid range
                                    (-180 to 360 degrees)
```

5.4 CONVERT_LAMBERT_TO_GEODETIC

5.4.1 DESCRIPTION

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

5.4.2 INTERFACES AND EXAMPLES

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

```
long Convert_Lambert_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_Lambert_To_Geodetic (Easting, Northing, Latitude, Longitude)
```

Inputs:

Easting:	5972201
Northing:	1872576

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:

```
LAMBERT_NO_ERROR                : No errors occurred in function
LAMBERT_EASTING_ERROR           : Easting outside of valid range
                                    (depends on ellipsoid and projection
                                     parameters)
LAMBERT_NORTHING_ERROR          : Northing outside of valid range
                                    (depends on ellipsoid and projection
                                     parameters)
```

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.

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 origin longitude 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.

Lambert Conformal Conic Projection – A conformal projection in which the projected parallels are arcs of concentric circles centered at the pole. The projected meridians are radii of concentric circles that meet at the pole. There are one or two parallels, called standard parallels, along which the point scale factor is one.

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 geo–graphical 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.

Origin Latitude – Latitude of the origin of the projected coordinate system.

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

Scale Factor (Projection) – A multplier 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.