REUSE MANUAL

LOCAL CARTESIAN

10xxxxxx.1

Implementation

TABLE OF CONTENTS

SECTION 1. INTRODUCTION

1.1 PURPOSE OF THE REUSE MANUAL

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

- Geodetic coordinates (latitude, longitude in radians and ellipsoidal height in meters) to Local Cartesian coordinates (X, Y and Z in meters),
- Geocentric coordinates (u, v and w in meters) to Local Cartesian coordinates (X, Y and Z in meters),
- Local Cartesian coordinates (X, Y and Z in meters) to Geocentric coordinates (u, v and w in meters),
- Local Cartesian coordinates (X, Y and Z in meters) to Geodetic coordinates (latitude, longitude in radians and ellipsoidal height in meters).

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 Local Cartesian coordinate system is specified in terms of the following parameters:

- Origin Longitude – Longitude (in radians) at the origin of the Local Cartesian coordinate system,
- Origin Latitude – Latitude (in radians) at the origin of the Local Cartesian coordinate system,
- Origin Height – Ellipsoidal height (in meters) at the origin of the Local Cartesian coordinate system,

- Orientation – Angle (in radians) which gives the direction of the Y axis relative to north.

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 LOCAL CARTESIAN component:

Source Code Files:

```
loccart.c
```

Header Files :

```
loccart.h
```

Data Files :

```
none
```

The compilation instructions for the LOCAL CARTESIAN component are as follows:

DOS Makefile (Uses Microsoft C):

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

UNIX Makefile (Uses gcc compiler):

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

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

2.1 PARTIAL REUSE

The LOCAL CARTESIAN component does not allow for partial reuse.

2.2 MODIFICATIONS

The LOCAL CARTESIAN component does not permit modifications.

SECTION 3. ENVIRONMENT

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

3.1 HARDWARE

3.1.1 DEVELOPMENT

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

- Solaris 2.5
- Windows 95

3.2.2 COMPILERS

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

```
LOCCART_NO_ERROR                : No errors occurred in function
LOCCART_LAT_ERROR               : Latitude outside of valid range
                                  (-90 to 90 degrees)
LOCCART_LON_ERROR               : Longitude outside of valid range
                                  (-180 to 360 degrees)
LOCCART_ORIGIN_LAT_ERROR        : Derived origin latitude outside of valid
                                  range (-90 to 90 degrees except close
                                  to the equator)
LOCCART_ORIGIN_LON_ERROR        : Origin longitude outside of valid range
                                  (-180 to 360 degrees)
LOCCART_A_ERROR                 : Semi-major axis less than or equal to zero
LOCCART_B_ERROR                 : Semi-minor axis less than or equal to zero
LOCCART_A_LESS_B_ERROR          : Semi-major axis less than semi-minor axis
LOCCART_ORIENTATION_ERROR       : Orientation angle outside of valid range
                                  (-360 to 360 degrees)
```

4.3 VARIABLES

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

```
Ellipsoid Parameters:
static double LocalCart_a       : Semi-major axis of ellipsoid in meters
static double LocalCart_b       : Semi-minor axis of ellipsoid in meters

Coordinate System Parameters:
static double LocalCart_Origin_Lat  : Latitude of origin in radians
static double LocalCart_Origin_Long : Longitude of origin in radians
static double LocalCart_Origin_Height : Height of origin in meters
static double LocalCart_Orientation : Orientation of Y axis in radians
```

4.4 INCLUDE FILES

```
math.h                          : Standard C math library
loccart.h                       : Error codes and prototype error checking
geocent.h                       : Converts geodetic coordinates to geocentric
                                  coordinates and geocentric coordinates to
                                  geodetic coordinates
```

4.5 DEPENDENCIES

The following is a list of the software external to the RSC and its descriptions.

GEOCENTRIC, since conversions between Geodetic coordinates and Local Cartesian coordinates go through Geocentric coordinates as an intermediate step.

SECTION 5. FUNCTIONS

5.1 SET_LOCAL_CARTESIAN_PARAMETERS

5.1.1 DESCRIPTION

This function sets the ellipsoid and Local Cartesian coordinate system 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.

```
int Set_Local_Cartesian_Parameters (double a,
                                    double b,
                                    double Origin_Latitude,
                                    double Origin_Longitude,
                                    double Origin_Height,
                                    double Orientation);
```

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 Local Cartesian coordinate system (input),
Origin_Longitude	Longitude in radians at the origin of the Local Cartesian coordinate system (input),
Origin_Height	Ellipsoidal height in meters at the origin of the Local Cartesian coordinate system (input),
Orientation	Orientation angle of the Local Cartesian coordinate system, in radians (input).

Example:

```
status = Set_Local_Cartesian_Parameters (a, b, Origin_Latitude,
           Origin_Longitude, Origin_Height, Orientation)
```

Inputs:

a	6378137.0
b	6356752.3142

Origin_Latitude 0.0

Origin_Longitude 0.0

Origin_Height 0.0

Orientation 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:

```
LOCCART_NO_ERROR                : No errors occurred in function
LOCCART_ORIGIN_LAT_ERROR        : Derived origin latitude outside of valid
                                    range (-90 to 90 degrees except close
                                    to the equator)
LOCCART_ORIGIN_LON_ERROR        : Origin longitude outside of valid range
                                    (-180 to 360 degrees)
LOCCART_A_ERROR                 : Semi-major axis less than or equal to zero
LOCCART_B_ERROR                 : Semi-minor axis less than or equal to zero
LOCCART_A_LESS_B_ERROR          : Semi-major axis less than semi-minor axis
LOCCART_ORIENTATION_ERROR       : Orientation angle outside of valid range
```

```
                                    (-360 to 360 degrees)
```

5.2 GET_LOCAL_CARTESIAN_PARAMETERS

5.2.1 DESCRIPTION

This function returns the current values of the ellipsoid parameters and Local Cartesian coordinate system parameters.

5.2.2 INTERFACES AND EXAMPLES

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

```
void Get_Local_Cartesian_Parameters (double *a,
                                     double *b,
                                     double *Origin_Latitude,
                                     double *Origin_Longitude,
                                     double *Origin_Height,
                                     double *Orientation);
```

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 Local Cartesian coordinate system (output),
Origin_Longitude	Longitude in radians at the origin of the Local Cartesian coordinate system (output),
Origin_Height	Ellipsoidal height in meters at the origin of the Local Cartesian coordinate system (output),
Orientation	Orientation angle of the Local Cartesian coordinate system, in radians (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_GEOCENTRIC_TO_LOCAL_CARTESIAN

5.3.1 DESCRIPTION

This function converts Geocentric coordinates (u, v and w in meters) to Local Cartesian coordinates (X, Y and Z in meters), using the current ellipsoid and Local Cartesian coordinate system parameters.

5.3.2 INTERFACES AND EXAMPLES

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

```
int Convert_Geocentric_To_Local_Cartesian (double u,
                                           double v,
                                           double w,
                                           double *X,
                                           double *Y,
                                           double *Z);
```

u	Geocentric X coordinate, in meters (input),
v	Geocentric Y coordinate, in meters (input),
w	Geocentric Z coordinate, in meters (input),
X	X coordinate in meters (output),

Y Y coordinate in meters (output),

Z Z coordinate in meters (output).

Example:

```
status = Convert_Geocentric_To_Local_Cartesian (u, v, w, X, Y, Z)
```

Inputs:

u: 0

v: 50000

w 0

Outputs:

X: 50000

Y: 21385

Z: -6367454

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

No errors are reported by this function.

5.4 CONVERT_GEODETIC_TO_LOCAL_CARTESIAN

5.4.1 DESCRIPTION

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

5.4.2 INTERFACES AND EXAMPLES

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

```
int Convert_Geodetic_To_Local_Cartesian (double Latitude,
                                         double Longitude,
                                         double Height,
                                         double *X,
                                         double *Y,
                                         double *Z);
```

Latitude	Latitude in radians (input),
Longitude	Longitude in radians (input),
Height	Ellipsoidal 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_Local_Cartesian (Latitude, Longitude, Height, X,
           Y, Z)
```

Inputs:

Latitude:	45.0
Longitude:	-75.0

Height 700

Outputs:

X: -4364136

Y: 4487843

Z: -5208770

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

Set_Geocentric_Parameters, in the GEOCENTRIC component – used to set the parameters of the Geocentric coordinate system, and

Convert_Geodetic_To_Geocentric, in the GEOCENTRIC component – used to convert Geodetic coordinates (latitude, longitude, and ellipsoidal height) to Geocentric (X, Y, Z) coordinates.

5.4.5 ERROR HANDLING

This function returns the following status codes:

```
LOCCART_NO_ERROR                : No errors occurred in function
LOCCART_LAT_ERROR               : Latitude outside of valid range
                                    (-90 to 90 degrees)
LOCCART_LON_ERROR               : Longitude outside of valid range
                                    (-180 to 360 degrees)
```

5.5 CONVERT_LOCAL_CARTESIAN_TO_GEOCENTRIC

5.5.1 DESCRIPTION

This function converts Local Cartesian coordinates (X, Y and Z in meters) to Geocentric coordinates (u, v and w in meters), using the current ellipsoid and Local Cartesian coordinate system parameters .

5.5.2 INTERFACES AND EXAMPLES

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

```
int Convert_Local_Cartesian_To_Geocentric (double X,
                                           double Y,
                                           double Z,
                                           double *u,
                                           double *v,
                                           double *w);
```

X	X coordinate in meters (input),
Y	Y coordinate in meters (input),
Z	Z coordinate in meters (input),
u	Geocentric X coordinate, in meters (output),
v	Geocentric Y coordinate, in meters (output),
w	Geocentric Z coordinate, in meters (output),

Example:

```
status = Convert_Local_Cartesian_To_Geodcentric (X, Y, Z, u, v, w)
```

Inputs:

X:	50000
Y:	21385
Z:	-6367454

Outputs:

u:	0

v: 50000

w: 0

5.5.3 DECLARATIONS

5.5.3.1 TYPES

Not applicable.

5.5.3.2 CONSTANTS

Not applicable.

5.5.3.3 VARIABLES

Not applicable.

5.5.4 DEPENDENCIES

None.

5.5.5 ERROR HANDLING

No errors are reported by this function

5.6 CONVERT_LOCAL_CARTESIAN_TO_GEODETIC

5.6.1 DESCRIPTION

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

5.6.2 INTERFACES AND EXAMPLES

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

```
int Convert_Local_Cartesian_To_Geodetic (double X,
```

```
                                        double Y,
                                        double Z,
                                        double *Latitude,
                                        double *Longitude,
                                        double *Height);
```

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

Example:

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

Inputs:

X:	-4364136
Y:	4487843
Z:	-5208770

Outputs:

Latitude:	45.0
Longitude:	-75.0
Height:	700

5.6.3 DECLARATIONS

5.6.3.1 TYPES

Not applicable.

5.6.3.2 CONSTANTS

Not applicable.

5.6.3.3 VARIABLES

Not applicable.

5.6.4 DEPENDENCIES

Set_Geocentric_Parameters, in the GEOCENTRIC component – used to set the parameters of the Geocentric coordinate system, and

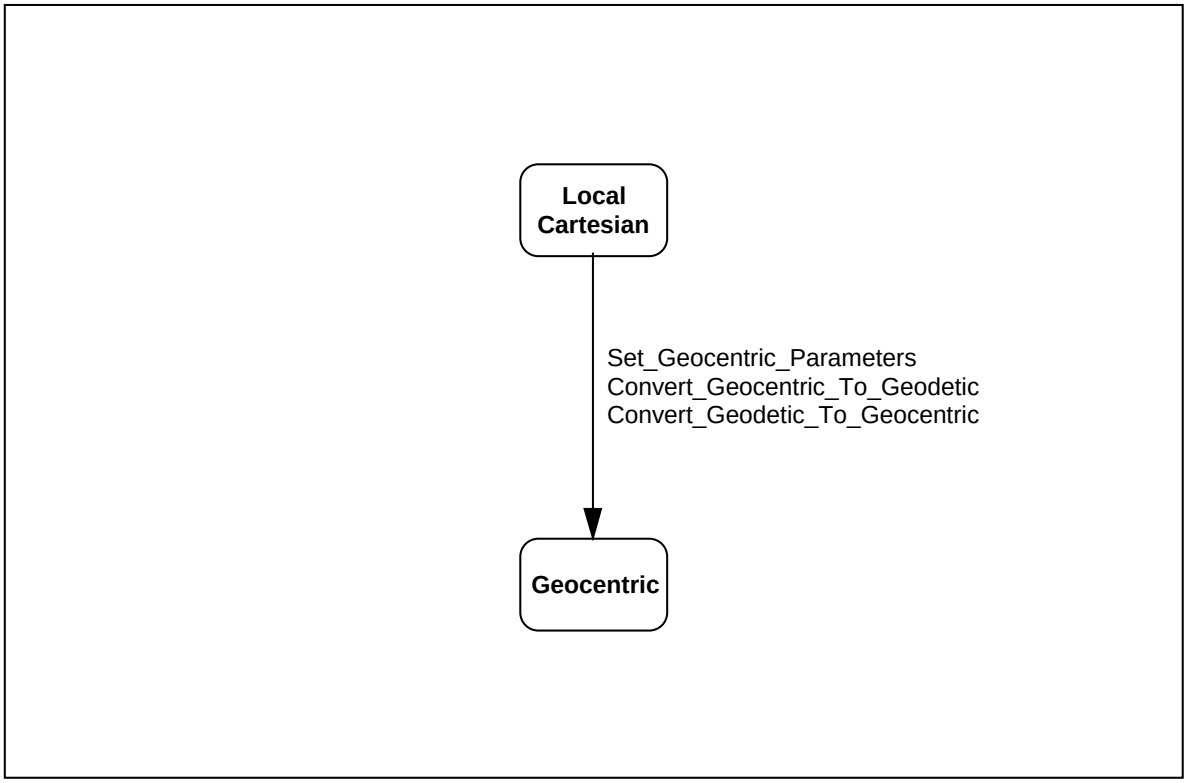
Convert_Geocentric_To_Geodetic, in the GEOCENTRIC component – used to convert Geocentric coordinates (X, Y, Z) to geodetic (latitude, longitude, and ellipsoidal height) coordinates.

5.6.5 ERROR HANDLING

No errors are reported by this function.

APPENDIX A STRUCTURE/DEPENDENCY DIAGRAMS

This component consists of a single compilation unit and depends on the GEOCENTRIC component.



APPENDIX B DEFINITIONS/GLOSSARY

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.

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.

Geodetic Height – The perpendicular distance from the reference sphere or ellipsoid to the earth's surface.

Geocentric Coordinates – Cartesian coordinates (U, V, W) that define the position of a point with respect to the center of mass of the earth.

Local Cartesian – The local rectangular coordinates X,Y and Z that define the position of a point on the surface of the earth with respect to an origin located on, or near, the surface of the reference ellipsoid.

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.

Orientation – Azimuthal rotation which gives the direction of the Y axis relative to north. The angle is given in radians and is measured clockwise from 0 to 2Π.

Origin Height – Ellipsoidal height of the origin of the Local Cartesian coordinate system.

Origin Latitude – Latitude of the origin of the Local Cartesian coordinate system.

Origin Longitude – Longitude of the origin of the Local Cartesian coordinate system.

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

APPENDIX C REFERENCES

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