

REUSE MANUAL

UPS

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_UPS_PARAMETERS.....	7
5.2 GET_UPS_PARAMETERS	8

5.3 CONVERT_GEODETTIC_TO_UPS	9
5.4 CONVERT_UPS_TO_GEODETTIC	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 UPS 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 UPS is to provide a reusable software component which supports the following coordinate conversions :

- Geodetic coordinates (latitude and longitude in radians) to Universal Polar Stereographic (UPS) projection coordinates (hemisphere, easting in meters, and northing in meters),
- Universal Polar Stereographic (UPS) projection coordinates (hemisphere, easting in meters, 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, and
- 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 (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

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

SECTION 2. INSTALLATION

The following is a list of the compilation files for the UPS component:

Source Code Files:

`ups.c`

Header Files :

`ups.h`

Data Files :

`none`

The compilation instructions for the UPS component are as follows:

DOS Makefile (Uses Microsoft C):

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

UNIX Makefile (Uses gcc compiler):

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

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

2.1 PARTIAL REUSE

The UPS component does not allow for partial reuse.

2.2 MODIFICATIONS

The UPS component does not permit modifications.

SECTION 3. ENVIRONMENT

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

3.1 HARDWARE

3.1.1 DEVELOPMENT

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

- Solaris 2.5
- Windows 95

3.2.2 COMPILERS

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

UPS_NO_ERROR	: No errors occurred in function
UPS_LAT_ERROR	: Latitude outside of valid range (North Pole: 83.5 to 90, South Pole: -79.5 to -90)
UPS_LON_ERROR	: Longitude outside of valid range (-180 to 360 degrees)
UPS_EASTING_ERROR	: Easting outside of valid range, (0 to 4,000,000m)
UPS_NORTHING_ERROR	: Northing outside of valid range, (0 to 4,000,000m)
UPS_HEMISPHERE_ERROR	: Invalid hemisphere ('N' or 'S')
UPS_A_ERROR	: Semi-major axis less than or equal to zero
UPS_B_ERROR	: Semi-minor axis less than or equal to zero
UPS_A_LESS_B_ERROR	: Semi-major axis less than semi-minor axisB

4.3 VARIABLES

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

Ellipsoid Parameters:	
static double UPS_a	: Semi-major axis of ellipsoid in meters
static double UPS_b	: Semi-minor axis of ellipsoid in meters

4.4 INCLUDE FILES

polar.h	: UPS depends on Polar Stereographic
ups.h	: Used for prototype checking

4.5 DEPENDENCIES

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

POLAR STEREOGRAPHIC, since the Universal Polar Stereographic projection is a special case of the Polar Stereographic projection.

SECTION 5. FUNCTIONS

5.1 SET_UPS_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_UPS_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_UPS_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

None.

5.1.5 ERROR HANDLING

This function returns the following status codes:

UPS_NO_ERROR	: No errors occurred in function
UPS_A_ERROR	: Semi-major axis less than or equal to zero
UPS_B_ERROR	: Semi-minor axis less than or equal to zero
UPS_A_LESS_B_ERROR	: Semi-major axis less than semi-minor axis

5.2 GET_UPS_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_UPS_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_GEODETTIC_TO_UPS

5.3.1 DESCRIPTION

This function converts Geodetic coordinates (latitude and longitude in radians) to Universal Polar Stereographic (UPS) projection coordinates (hemisphere, easting in meters, and northing 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_UPS (double Latitude,  
                             double Longitude,  
                             char   *Hemisphere,  
                             double *Easting,  
                             double *Northing);
```

Latitude	Latitude in radians (input),
Longitude	Longitude in radians (input),
Hemisphere	Northern ('N') or Southern ('S') hemisphere (output),
Easting	Easting (X) in meters (output),
Northing	Northing (Y) in meters (output).

Example:

```
status = Convert_Geodetic_To_UPS (Latitude, Longitude, Hemisphere, Easting,  
                                  Northing)
```

Inputs:

Latitude:	87 17 14 S
Longitude:	132 14 52 E

Outputs:

Hemisphere:	“S”
Easting:	2222990
Northing:	1797470

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

Set_Polar_Stereographic_Parameters, in the POLAR STEREOGRAPHIC component – used to set the parameters of the Polar Stereographic projection, and

Convert_Geodetic_To_Polar_Stereographic, in the POLAR STEREOGRAPHIC component – used to convert geodetic (latitude and longitude) coordinates to Polar Stereographic (easting and northing) projection coordinates.

5.3.5 ERROR HANDLING

This function returns the following status codes:

UPS_NO_ERROR	: No errors occurred in function
UPS_LAT_ERROR	: Latitude outside of valid range (North Pole: 83.5 to 90, South Pole: -79.5 to -90)
UPS_LON_ERROR	: Longitude outside of valid range (-180 to 360 degrees)

5.4 CONVERT_UPS_TO_GEODETTIC

5.4.1 DESCRIPTION

This function converts Universal Polar Stereographic (UPS) projection coordinates (hemisphere, easting in meters, and northing in meters) to Geodetic coordinates (latitude and longitude in radians), 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.

```
int Convert_UPS_To_Geodetic (char Hemisphere,  
                             double Easting,  
                             double Northing,  
                             double *Latitude,  
                             double *Longitude);
```

Hemisphere	Northern ('N') or Southern ('S') hemisphere (input),
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_UPS_To_Geodetic (Hemisphere, Easting, Northing, Latitude,  
                                  Longitude)
```

Inputs:

Hemisphere: "S"

Easting: 2222990

Northing: 1797470

Outputs:

Latitude: 87 17 14 S

Longitude: 132 14 52 E

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_Polar_Stereographic_Parameters, in the POLAR STEREOGRAPHIC component – used to set the parameters of the Polar Stereographic projection, and

Convert_Polar_Stereographic_To_Geodetic, in the POLAR STEREOGRAPHIC component – used to convert Polar Stereographic (easting and northing) projection coordinates to geodetic (latitude and longitude) coordinates.

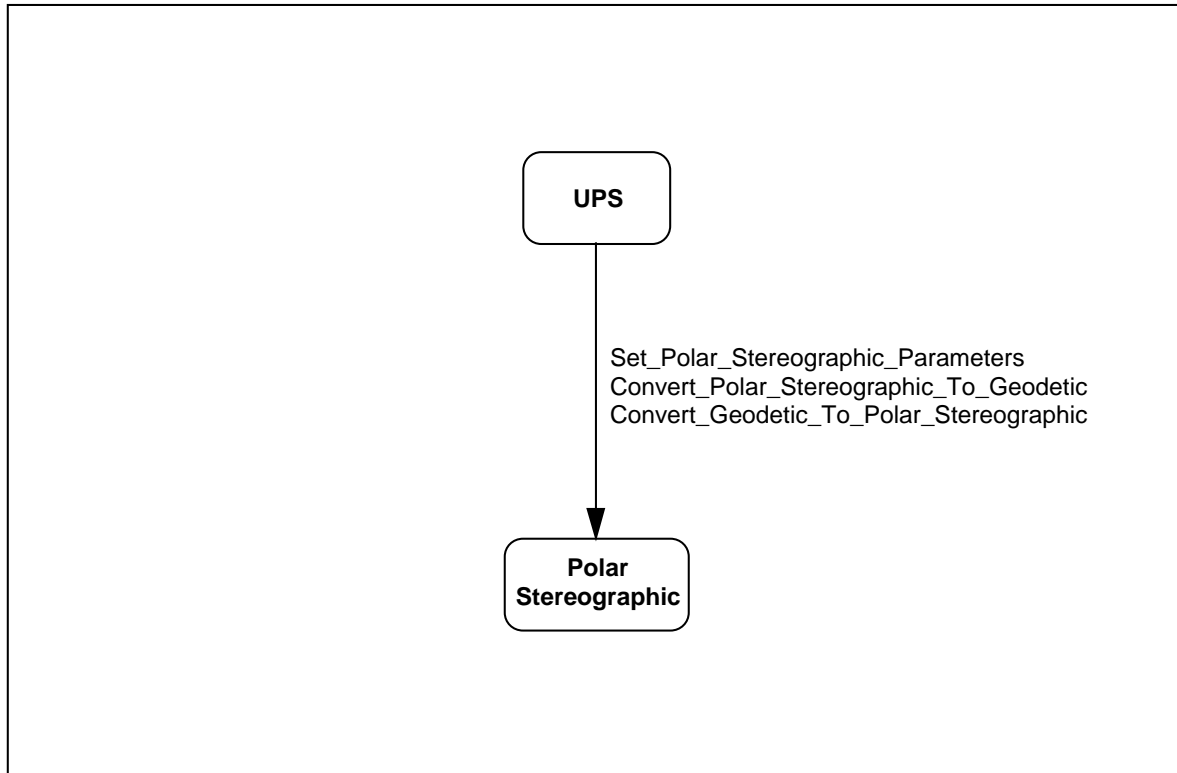
5.4.5 ERROR HANDLING

This function returns the following status codes:

UPS_NO_ERROR	: No errors occurred in function
UPS_HEMISPHERE_ERROR	: Invalid hemisphere ('N' or 'S')
UPS_EASTING_ERROR	: Easting outside of valid range (0 to 4,000,000m)
UPS_NORTHING_ERROR	: Northing outside of valid range (0 to 4,000,000m)

APPENDIX A STRUCTURE/DEPENDENCY DIAGRAMS

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

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.

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.

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

Polar Stereographic Projection – A conformal projection is a limiting case of the Lambert conformal conic projection when the one standard parallel approaches a pole. In this conformal projection, meridians are straight lines, parallels are concentric circles, and the point scale factor is one at the pole.

Universal Polar Stereographic (UPS) Projection – A family of two projections based on the Polar Stereographic projection, one for each of the poles. Both the X value, called the easting, and the Y value, called the northing, have values of 2,000,000m at the poles. The point scale factor at each pole 0.9994.

APPENDIX C REFERENCES

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