

ALBERS

10xxxxxx.1

LEGAL RESTRICTIONS:

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

DESCRIPTION:

ALBERS is a C language code component that provides conversions between Geodetic coordinates (latitude and longitude) and Albers Equal Area Conic projection coordinates (easting and northing) with two standard parallels.

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.

LEVEL OF TESTING/ACCEPTANCE:

Unit and integration testing have been performed for the functions contained in this component.

PURPOSE/INTENDED USE:

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

- Geodetic coordinates (latitude and longitude in radians) to Albers Equal Area Conic projection coordinates (easting and northing in meters) with two standard parallels,
- Albers Equal Area 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 Albers Equal Area Conic projection with 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.

HARDWARE/ENVIRONMENT CONSTRAINTS:

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

FUNCTIONS:

Set_Albers_Parameters – This function sets the ellipsoid parameters and Albers Equal Area Conic projection parameters for the particular variation of the Albers Equal Area Conic projection that is to be used in subsequent coordinate conversion operations.

Get_Albers_Parameters – This function returns the current values of the ellipsoid parameters and Albers Equal Area Conic projection parameters.

Convert_Geodetic_To_Albers – This function converts the specified geodetic coordinates (latitude and longitude in radians) to Albers Equal Area Conic projection coordinates (easting and northing in meters) using the current ellipsoid parameters and Albers Equal Area Conic projection parameters.

Convert_Albers_To_Geodetic – This function converts the specified Albers Equal Area Conic projection coordinates (easting and northing in meters) to geodetic coordinates (latitude and longitude in radians) using the current ellipsoid parameters and Albers Equal Area Conic projection parameters.

EXAMPLE APPLICATIONS:

The following example illustrates how ALBERS can be used to convert Geodetic coordinates to Albers Equal Area Conic projection coordinates and back again:

Function Call:

```
status = Set_Albers_Parameters (a, b, Standard_Parallel_1, Standard_Parallel_2,  
                                Origin_Latitude, Central_Meridian, False_Easting, False_Northing)
```

Inputs:

a	6378137.0
b	6356752.3142
Standard_Parallel_1	40.0
Standard_Parallel_2	50.0
Origin_Latitude	45.0
Central_Meridian	0.0
False_Easting	0.0
False_Northing	0.0

Function Call:

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

Inputs:

Latitude:	-35.0
Longitude:	75.0

Outputs:

Easting: 10926833

Northing: -1895253

Function Call:

status = Convert_Albers_To_Geodetic (Easting, Northing, Latitude, Longitude)

Inputs:

Easting: 10926833

Northing: -1895253

Outputs:

Latitude: -35.0

Longitude: 75.0