

LOCAL CARTESIAN

10xxxxxx.1

LEGAL RESTRICTIONS:

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

DESCRIPTION:

LOCAL CARTESIAN is a C language code component that provides conversions between Geodetic coordinates (latitude, longitude and height) and Local Cartesian coordinates (X, Y and Z).

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 LOCAL CARTESIAN is to provide a reusable component which supports the following coordinate conversions:

- Geodetic coordinates (latitude, longitude in radians and 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 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 variation of Local Cartesian is specified in terms of the following parameters:

- Origin Longitude – Longitude (in radians) at the origin of the coordinate system,
- Origin Latitude – Latitude (in radians) at the origin of the coordinate system,
- Origin Height – Height (in meters) at the origin of the coordinate system,
- Orientation – Angle (in radians) which gives the direction of the Y axis relative to north.

HARDWARE/ENVIRONMENT CONSTRAINTS:

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

FUNCTIONS:

Set_Local_Cartesian_Parameters – This function sets the ellipsoid parameters and Local Cartesian parameters for the particular variation of Local Cartesian that is to be used in subsequent coordinate conversion operations.

Get_Local_Cartesian_Parameters – This function returns the current values of the ellipsoid parameters and Local Cartesian parameters.

Convert_Geocentric_To_Local_Cartesian – This function converts the specified geocentric coordinates (u, v and w in meters) to Local Cartesian coordinates (X, Y and Z in meters) using the current ellipsoid parameters and Local Cartesian parameters.

Convert_Geodetic_To_Local_Cartesian – This function converts the specified geodetic coordinates (latitude, longitude in radians and height in meters) to geocentric coordinates (U, V and W in meters) by calling the **Convert_Geodetic_To_Geocentric** function. The geocentric coordinates are then

converted to Local Cartesian coordinates (X, Y and Z in meters) using the current ellipsoid parameters and Local Cartesian parameters.

Convert_Local_Cartesian_To_Geocentric – This function converts the specified Local Cartesian coordinates (X, Y and Z in meters) to geocentric coordinates (u, v, and w in meters), using the current ellipsoid parameters and Local Cartesian parameters.

Convert_Local_Cartesian_To_Geodetic – This function converts the specified Local Cartesian coordinates (X, Y and Z in meters) to geocentric coordinates (U, V, and W in meters), using the current ellipsoid parameters and Local Cartesian parameters. The geocentric coordinates are then converted to geodetic coordinates (latitude, longitude in radians and height in meters) by calling the **Convert_Geocentric_To_Geodetic** function.

EXAMPLE APPLICATIONS:

The following example illustrates how LOCAL CARTESIAN can be used to convert Geodetic coordinates to Local Cartesian coordinates and back again:

Function Call:

```
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

Function Call:

```
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

Function Call:

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

Function Call:

status = Convert_Local_Cartesian_To_Geocentric (X, Y, Z, u, v, w)

Inputs:

X:	50000
Y:	21385

Z: -6367454

Outputs:

u: 0

v: 50000

w: 0

Function Call:

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