Coordinate Transformation Custom Derived Function

January 2013

SYMVIONICS Document SSD-IADS-052, Rev 8.0

Created October 20, 2010

Table of Contents

Table of Figures

1. Introduction

This document describes the IADS Coordinate Transformation Custom Derived Function developed to assist flight test engineers with position, distance and bearing calculations. The WGS84 Ellipsoid is used for these functions.

1.1. Overview

The IADS Coordinate Transformation custom derived function is provided as a Dynamic Link Library (DLL) which contains the following internal functions:

Convert ECEF coordinates to Geodetic coordinates. Provided as three separate functions that return Latitude, Longitude, and Altitude MSL given X, Y, Z coordinates.

Convert Geodetic coordinates to ECEF coordinates. Provided as three separate functions that return X, Y, and Z given Latitude, Longitude, and Altitude MSL coordinates.

Calculate look angles using two positions' ECEF or geodetic coordinates. Provided as three separate functions that return Azimuth, Elevation and Range given two positions' X, Y, Z (ECEF) coordinates. Three additional functions are provided to return Azimuth, Elevation and Range given two position's Latitude, Longitude and Altitude.

Calculate look angles using two positions' geodetic coordinates. Provided as two separate functions that return Azimuth and Elevation given two positions' Latitude, Longitude and Altitude and the first position's Roll, Pitch and Heading.

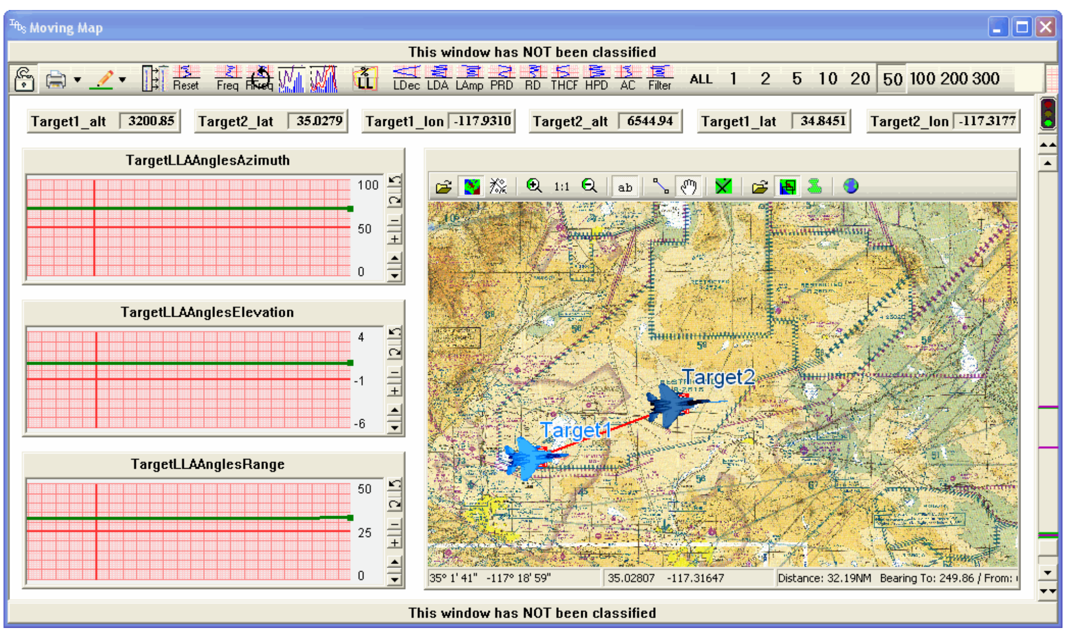


Figure 1-1 Coordinate Transformation Parameters in IADS

2. Installation

The IADS CoordinateTransformationFuncs.dll is included in IADS Version 7.0 (or greater) and registers automatically upon installation of the IADS application. The following procedures are provided for other circumstances.

2.1. To register the CoordinateTransformationFuncs.dll:

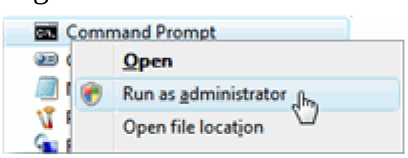
1) Copy the *CoordinateTransformationFuncs.dll* file to a known directory. If you are unsure where to place it, consider the C:\Program Files\IADS directory.
2) In Windows Explorer, navigate to the directory where you copied the file and right-click on the *CoordinateTransformationFuncs.dll* file and choose **Open With…**
3) Click the **Browse** button.
4) Browse to the **C:\Windows\System32** directory.
5) Select the **regsvr32.exe** file and then click the **Open** button.
6) Click **OK**. The function is now registered and available for use within IADS.

2.2. To turn off User Account Control (Windows Vista/7):

1) Open the *Control Panel* (Classic View).
2) Click on the **User Account** icon.
3) Click on **Turn User Account Control on or off**.
4) Uncheck **Use UAC to help protect your computer**.
5) Click **OK**.
6) Click on **Restart Now** to apply the changes.
7) Register the *CoordinateTransformationFuncs.dll*.
8) Repeat steps 1-6 in this set of procedures to turn back on *User Account Control*.

2.3. To register a DLL as Administrator (Windows Vista/7):

1) Right-click on the Command Prompt icon > **Run as Administrator**.



2) Enter your administrative credentials (User name, Password).
3) In the Command Prompt, navigate to the location of the dll.
4) Type: **regsvr32 CoordinateTransformationFuncs.dl**
5) Press **Enter**.

```
C:\>cd prog*

C:\Program Files>cd iads

C:\Program Files\IADS>regsvr32 CoordinateTransformationFuncs.dll

C:\Program Files\IADS>
```

ProgId	***Description***
CoordinateTransformationFuncs.ECEFtoLLA_Alt	Returns Altitude MSL given X,Y,Z coordinates
CoordinateTransformationFuncs.ECEFtoLLA_AGL_Alt	Returns Altitude AGL given X,Y,Z coordinates
CoordinateTransformationFuncs.ECEFtoLLA_Lat	Returns Latitude given X,Y,Z coordinates
CoordinateTransformationFuncs.ECEFtoLLA_Lon	Returns Longitude given X,Y,Z coordinates
CoordinateTransformationFuncs.LLAtoECEF_X	Returns X given latitude, longitude, and altitude
CoordinateTransformationFuncs.LLAtoECEF_Y	Returns Y given latitude, longitude, and altitude
CoordinateTransformationFuncs.LLAtoECEF_Z	Returns Z given latitude, longitude, and altitude
CoordinateTransformationFuncs.ECEFLookAngles_Az	Returns Azimuth given X,Y,Z coordinates
CoordinateTransformationFuncs.ECEFLookAngles_EL	Returns Elevation given X,Y,Z coordinates
CoordinateTransformationFuncs.ECEFLookAngles_Range	Returns Range given X,Y,Z coordinates
CoordinateTransformationFuncs.LLALookAngles_Az	Returns Azimuth given latitude, longitude and altitude
CoordinateTransformationFuncs.LLALookAngles_El	Returns Elevation given latitude, longitude and altitude
CoordinateTransformationFuncs.LLALookAngles_Range	Returns Range given latitude, longitude and altitude
CoordinateTransformationFuncs.LLALookAnglesWithRotations_Az	Returns Azimuth given latitude, longitude, altitude, roll, pitch and heading
CoordinateTransformationFuncs.LLALookAnglesWithRotations_El	Returns Elevation given latitude, longitude, altitude, roll, pitch and heading
CoordinateTransformationFuncs.LLtoTerrainHeight	Returns GTOPO30 Terrain Elevation given latitude and longitude

Figure 2-1 The CoordinateTransformationFuncs.dll Installed Functions

3. Instructions for Use

This section will explain how to create the derived functions in IADS. Figure 3-1 displays the coordinate transformation functions in the Parameter Defaults Table as derived parameters.

	Parameter	DataSourceType	DataSourceArguement
1	ECEFtoLLA_Alt	Derived	CoordinateTransformationFuncs.ECEFtoLLA_Alt(X,Y,Z,0)
2	ECEFtoLLA_Lat	Derived	CoordinateTransformationFuncs.ECEFtoLLA_Lat(X,Y,Z,0)
3	ECEFtoLLA_Lon	Derived	CoordinateTransformationFuncs.ECEFtoLLA_Lon(X,Y,Z,0)
4	LLAtoECEF_X	Derived	CoordinateTransformationFuncs.LLAtoECEF_X(LatitudeInDegrees,LongitudeInDegrees,AltitudeInMeters,0)
5	LLAtoECEF_Y	Derived	CoordinateTransformationFuncs.LLAtoECEF_Y(LatitudeInDegrees,LongitudeInDegrees,AltitudeInMeters,0)
6	LLAtoECEF_Z	Derived	CoordinateTransformationFuncs.LLAtoECEF_Z(LatitudeInDegrees,LongitudeInDegrees,AltitudeInMeters,0)
7	ECEFLookAngles_Az	Derived	CoordinateTransformationFuncs.ECEFLookAngles_Az(X1,Y1,Z1,X2,Y2,Z2,0)
8	ECEFLookAngles_El	Derived	CoordinateTransformationFuncs.ECEFLookAngles_El(X1,Y1,Z1,X2,Y2,Z2,0)
9	ECEFLookAngles_Range	Derived	CoordinateTransformationFuncs.ECEFLookAngles_Range(X1,Y1,Z1,X2,Y2,Z2,0)
10	LLALookAngles_Az	Derived	CoordinateTransformationFuncs.LLALookAngles_Az(LatitudeInDegrees,LongitudeInDegrees,AltitudeInMeters
11	LLALookAngles_El	Derived	CoordinateTransformationFuncs.LLALookAngles_El(LatitudeInDegrees,LongitudeInDegrees,AltitudeInMeters,
12	LLALookAngles_Range	Derived	CoordinateTransformationFuncs.LLALookAngles_Range(LatitudeInDegrees,LongitudeInDegrees,AltitudeInMet

Figure 3-1 Derived Parameter Setup in the Parameter Defaults Table

To create a coordinate transformation derived parameter:

1) In IADS, on the Dashboard click the **Configuration** button.
2) Open the **Data** folder, then click **Parameter Defaults**.
3) Copy and paste an existing row of data for a parameter that is similar to the one you are creating.
4) In the *Parameter* column enter a unique parameter name.
5) In the *Data Source Type* column, select **Derived**.
6) Enter the desired *Coordinate Transformation* function in the *Data Source Argument* column, for example:

   CoordinateTransformationFuncs.LLAtoECEF_X(LatitudeInDegrees,LongitudeInDegrees,AltitudeInMeters,0)

7) Click the *Save* button.

4. Coordinate Transformation Function Reference

The tables below detail the input arguments required for each of the internal coordinate transformation functions. If an equation is entered incorrectly in the *Data Source Argument* cell of the Parameter Defaults Table the system will flag a syntax error.

The following functions use Earth-Centered, Earth-Fixed, (ECEF) X,Y,Z coordinates and geodetic latitude, longitude and altitude coordinates as function inputs.

ECEF is a Cartesian coordinate system; sometimes known as a conventional terrestrial system. It represents positions as an X, Y, and Z coordinate. The point (0,0,0) denotes the mass center of the earth. The X axis intersects the sphere of the earth at the 0° latitude, 0° longitude (passes through the equator at the prime meridian). The Y axis can be determined by the right-hand rule to be passing through the equator at 90° longitude. The Z axis is defined as being parallel to the earth rotational axes, pointing towards north. This means the ECEF rotates with the earth around its Z axis. Therefore, coordinates of a point fixed on the surface of the earth do not change.

4.1. ECEF to Geodetic Coordinate Functions

These functions currently exist in IADS as XYZtoLat, XYZtoLon, and XYZtoAlt, and also as the following *Coordinate Transformation* internal functions.

Syntax:

CoordinateTransformationFuncs.ECEFtoLLA_Lat(X, Y, Z, 0)

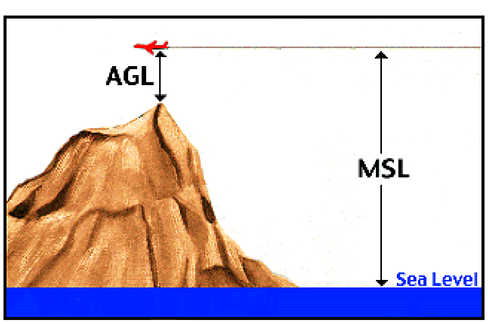
CoordinateTransformationFuncs.ECEFtoLLA_Lon(X, Y, Z, 0)

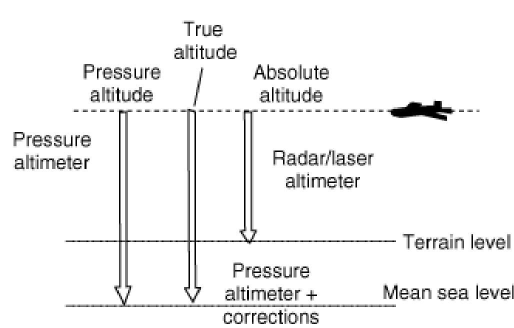
CoordinateTransformationFuncs.ECEFtoLLA_Alt(X, Y, Z, 0) *Altitude above Mean Sea Level*

CoordinateTransformationFuncs.ECEFtoLLA_AGL_Alt(X, Y, Z, 0) *Altitude Above Ground Level*

Input	*Description*
Arg1	X Coordinate
Arg2	Y Coordinate
Arg3	Z Coordinate
Arg4	0 = Units in meters; 1 = Feet

Figure 4-1 ECEF to Geodetic Function Inputs





4.2. Geodetic to ECEF Coordinate Functions

These functions currently exist in IADS as LatLonAlttoX, LatLonAlttoY, and LatLonAlttoZ, and also as the following *Coordinate Transformation* internal functions.

Syntax:

CoordinateTransformationFuncs.LLAtoECEF_X(Lat, Lon, Alt, 0)

CoordinateTransformationFuncs.LLAtoECEF_Y(Lat, Lon, Alt, 0)

CoordinateTransformationFuncs.LLAtoECEF_Z(Lat, Lon, Alt, 0)

Input	***Description***
Arg1	Lat - Latitude in degrees
Arg2	Lon - Longitude in degrees
Arg3	Alt - Altitude in meters
Arg4	0 = Units in meters; 1 = Feet

Figure 4-2 Geodetic to ECEF Function Inputs

4.3. ECEF to Azimuth, Elevation, and Range Look Angle Functions

Syntax:

CoordinateTransformationFuncs.ECEFLookAngles_Az(X1, Y1, Z1, X2, Y2, Z2, 0)

CoordinateTransformationFuncs.ECEFLookAngles_EL(X1, Y1, Z1, X2, Y2, Z2, 0)

CoordinateTransformationFuncs.ECEFLookAngles_Range(X1, Y1, Z1, X2, Y2, Z2, 0)

Input	***Description***
Arg1	X1 - Observer (origin) point X coordinate
Arg2	Y1 - Observer (origin) point Y coordinate
Arg3	Z1 - Observer (origin) point Z coordinate
Arg4	X2 - Interest point X coordinate
Arg5	Y2 - Interest point Y coordinate
Arg6	Z2 - Interest point Z coordinate
Arg7	0 = Units in meters; 1 = Feet

Figure 4-3 ECEF to Az, El and Range Look Angle Function Inputs

4.4. LLA to Azimuth, Elevation, and Range Look Angle Functions

Syntax:

CoordinateTransformationFuncs.LLALookAngles_Az(Lat1, Lon1, Alt1, Lat2, Lon2, Alt2, 0)

CoordinateTransformationFuncs.LLALookAngles_EL(Lat1, Lon1, Alt1, Lat2, Lon2, Alt2, 0)

CoordinateTransformationFuncs.LLALookAngles_Range(Lat1, Lon1, Alt1, Lat2, Lon2, Alt2, 0)

Input	***Description***
Arg1	Lat1 - Observer (origin) point latitude
Arg2	Lon1 - Observer (origin) point longitude
Arg3	Alt1 - Observer (origin) point altitude
Arg4	Lat2 - Interest point latitude
Arg5	Lon2 - Interest point longitude
Arg6	Alt2 - Interest point altitude
Arg7	0 = Units in meters; 1 = Feet

Figure 4-4 LLA to Az, El and Range Look Angle Function Inputs

4.5. LLA to Azimuth and Elevation with Rotations Look Angle Functions

Syntax:

CoordinateTransformationFuncs.LLALookAnglesWithRotations_Az(Lat1, Lon1, Alt1, Lat2, Lon2, Alt2, Roll, Pitch, Heading, 0, RotationOrder)

CoordinateTransformationFuncs.LLALookAnglesWithRotations_El(Lat1, Lon1, Alt1, Lat2, Lon2, Alt2, Roll, Pitch, Heading, 0, RotationOrder)

Input	*Description*
Arg1	Lat1 - Observer (origin) point latitude
Arg2	Lon1 - Observer (origin) point longitude
Arg3	Alt1 - Observer (origin) point altitude
Arg4	Lat2 - Interest point latitude
Arg5	Lon2 - Interest point longitude
Arg6	Alt2 - Interest point altitude
Arg7	Roll in Degrees
Arg8	Pitch in Degrees
Arg9	Heading in Degrees
Arg10	0 = Units in meters; 1 = Feet
Arg11	Rotation Order – 0=HPR, 1=HRP, 2=PHR, 3=PRH, 4=RHP, 5=RPH

Figure 4-5 LLA to Az and El with Rotations Look Angle Function Inputs

4.6. GTOPO30 Terrain Elevation Function

Syntax:

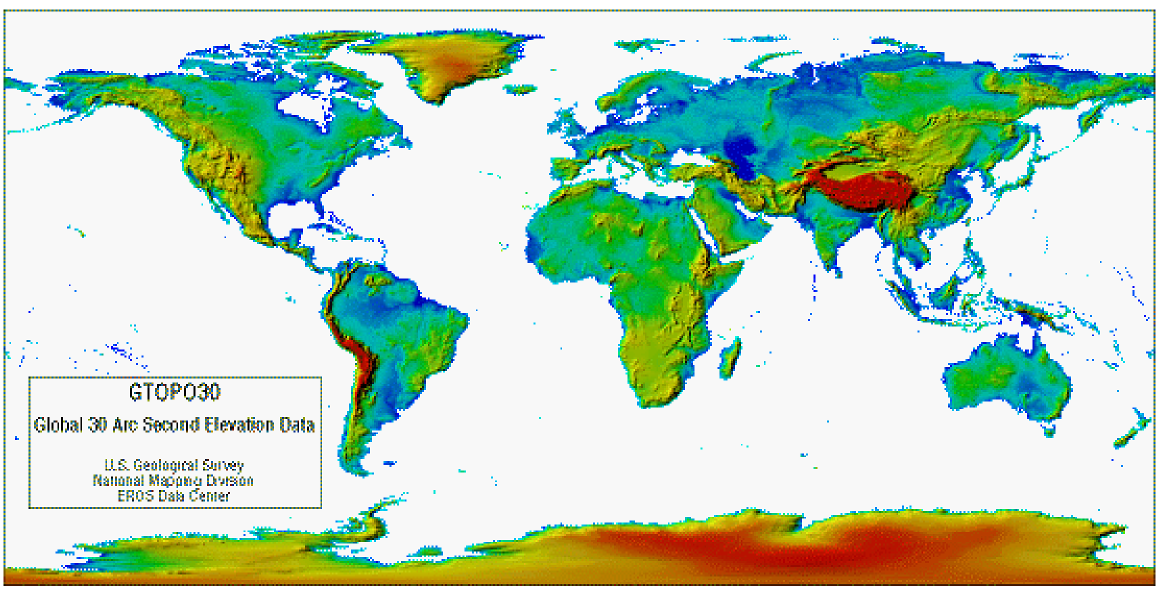
CoordinateTransformationFuncs.LLtoTerrainHeight(Lat, Lon, Units, “Terrain Data Path”, InterpId)

For example:

CoordinateTransformationFuncs.LLtoTerrainHeight(34.6002, -117.8335, 0, “D:\Test\TerrainData”, 0)

Input	*Description*
Arg1	Lat1 - Observer (origin) point latitude
Arg2	Lon1 - Observer (origin) point longitude
Arg3	0 = Units in meters; 1 = Feet
Arg4	Terrain Data File Path
Arg5	Interp Method (0 = bilinear, any other number = nearest)

Figure 4-6 GTOPO30 Terrain Elevation Function



To download the GTOPO30 Terrain Data:

GTOPO30 is available electronically through an Internet anonymous File Transfer Protocol (FTP) account at the EROS Data Center (at no cost).

1. Navigate to ftp://edcftp.cr.usgs.gov/pub/data/gtopo30/global
2. Download the following files: **w100n40.tar.gz**, **w140n40.tar.gz**, **w100n90.tar.gz** and **w140n90.tar.gz**. Double click on the file name to select a directory on your machine to save the file. If you do not have access to Gzip, you can leave off the .gz extension

and the FTP server will decompress the tar file as it is downloaded. You will still have to run the tar command to extract separate files. Please view the README.TXT file for information on extracting the tar files.

3. Extract the .DEM file from each tar file into the same folder.
4. Use the path to this folder as ***Arg4*** in the derived equation.

GTOPO30 tiles