

GEOTRANS ENGINE

10xxxxxx.1

LEGAL RESTRICTIONS:

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

DESCRIPTION:

GEOTRANS ENGINE is a C language code component that performs general-purpose datum transformation and coordinate conversion operations. Input coordinates that are defined with respect to a specified input datum and coordinate system, projection, or grid system are converted to a specified output datum and coordinate system, projection, or grid system. Conversions between ellipsoid heights and geoid (or MSL) heights are also supported.

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 GEOTRANS ENGINE is to provide a reusable component that supports the following coordinate conversions:

- Conversions between geodetic, geocentric, and local cartesian coordinate systems,
- Conversions between geodetic coordinates (latitude and longitude in radians) and various types of map projection coordinates (easting and northing in linear units),

- Conversions between geodetic coordinates and Military Grid Reference System (MGRS) or World Geographic Reference System (GEOREF) grid coordinates,
- Transformations of geodetic or geocentric coordinates between different global or local horizontal datums,
- Transformations between ellipsoid heights and geoid (or local MSL) heights, and
- Conversions that combine two or more of the above operations.

HARDWARE/ENVIRONMENT CONSTRAINTS:

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

FUNCTIONS:

`Initialize_Engine` – This function sets the initial state of the engine in preparation for coordinate conversion and/or datum transformation operations.

`Get_Coordinate_System_Count` – This function returns the number of different coordinate systems (including projections and grid systems) supported by the engine.

`Get_Coordinate_System_Index` – This function returns the index of the coordinate system with the specified code.

`Get_Coordinate_System_Type` – This function returns the type of the coordinate system with the specified index.

`Get_Coordinate_System_Name` – This function returns the name of the coordinate system with the specified index.

`Get_Coordinate_System_Code` – This function returns the code of the coordinate system with the specified index.

`Set_Coordinate_System` – This function sets the current input or output coordinate system to the specified type.

`Get_Coordinate_System` – This function returns the type of the current input or output coordinate system.

`Get_Datum_Count` – This function returns the number of different datums supported by the engine.

Get_Datum_Index – This function returns the index of the datum with the specified code.

Get_Datum_Name – This function returns the name of the datum with the specified index.

Get_Datum_Code – This function returns the code of the datum with the specified index.

Get_Datum_Ellipsoid_Code – This function returns the code for the ellipsoid associated with the datum with the specified index.

Set_Datum – This function sets the current input or output datum to the datum with the specified index.

Get_Datum – This function returns the index of the current input or output datum.

Define_Datum – This function creates a new local 3-parameter datum with the specified code, name, shift values (relative to WGS84), standard errors values, and domain of validity.

Get_Ellipsoid_Count – This function returns the number of different ellipsoids supported by the engine.

Get_Ellipsoid_Index – This function returns the index of the ellipsoid with the specified code.

Get_Ellipsoid_Name – This function returns the name of the ellipsoid with the specified index.

Get_Ellipsoid_Code – This function returns the code of the ellipsoid with the specified index.

Define_Ellipsoid – This function creates a new ellipsoid with the specified code, name, and semi-major and semi-minor axis values.

Set_Precision – This function sets the current output precision to the specified level.

Get_Precision – This function returns the current output precision level.

Set_Parameters – This function sets the parameters of the current input or output coordinate system to their default values.

Set_Geocentric_Coordinates – This function sets the current input or output Geocentric coordinates to the specified values.

Get_Geocentric_Coordinates – This function returns the current input or output Geocentric coordinates.

Set_Geodetic_Params – This function sets the current input or output Geodetic coordinate system parameters to the specified values.

Get_Geodetic_Params – This function returns the current input or output Geodetic coordinate system parameters.

Set_Geodetic_Coordinates – This function sets the current input or output Geodetic coordinates to the specified values.

Get_Geodetic_Coordinates – This function returns the current input or output Geodetic coordinates.

Set_GEOREF_Coordinates – This function sets the current input or output GEOREF coordinate string to the specified value.

Get_GEOREF_Coordinates – This function returns the current input or output GEOREF coordinate string.

Set_Albers_Equal_Area_Conic_Params – This function sets the current input or output Albers Equal Area Conic projection parameters to the specified values.

Get_Albers_Equal_Area_Conic_Params – This function returns the current input or output Albers Equal Area Conic projection parameters.

Set_Albers_Equal_Area_Conic_Coordinates – This function sets the current input or output Albers Equal Area Conic projection coordinates to the specified values.

Get_Albers_Equal_Area_Conic_Coordinates – This function returns the current input or output Albers Equal Area Conic projection coordinates.

Set_Bonne_Params – This function sets the current input or output Bonne projection parameters to the specified values.

Get_Bonne_Params – This function returns the current input or output Bonne projection parameters.

Set_Bonne_Coordinates – This function sets the current input or output Bonne projection coordinates to the specified values.

Get_Bonne_Coordinates – This function returns the current input or output Bonne projection coordinates.

Set_Cassini_Params – This function sets the current input or output Cassini projection parameters to the specified values.

Get_Cassini_Params – This function returns the current input or output Cassini projection parameters.

Set_Cassini_Coordinates – This function sets the current input or output Cassini projection coordinates to the specified values.

Get_Cassini_Coordinates – This function returns the current input or output Cassini projection coordinates.

Set_Cylindrical_Equal_Area_Params – This function sets the current input or output Cylindrical Equal Area projection parameters to the specified values.

Get_Cylindrical_Equal_Area_Params – This function returns the current input or output Cylindrical Equal Area projection parameters.

Set_Cylindrical_Equal_Area_Coordinates – This function sets the current input or output Cylindrical Equal Area projection coordinates to the specified values.

Get_Cylindrical_Equal_Area_Coordinates – This function returns the current input or output Cylindrical Equal Area projection coordinates.

Set_Eckert4_Params – This function sets the current input or output Eckert IV projection parameters to the specified values.

Get_Eckert4_Params – This function returns the current input or output Eckert IV projection parameters.

Set_Eckert4_Coordinates – This function sets the current input or output Eckert IV projection coordinates to the specified values.

Get_Eckert4_Coordinates – This function returns the current input or output Eckert IV projection coordinates.

Set_Eckert6_Params – This function sets the current input or output Eckert VI projection parameters to the specified values.

Get_Eckert6_Params – This function returns the current input or output Eckert VI projection parameters.

Set_Eckert6_Coordinates – This function sets the current input or output Eckert VI projection coordinates to the specified values.

Get_Eckert6_Coordinates – This function returns the current input or output Eckert VI projection coordinates.

Set_Equidistant_Cylindrical_Params – This function sets the current input or output Equidistant Cylindrical projection parameters to the specified values.

Get_Equidistant_Cylindrical_Params – This function returns the current input or output Equidistant Cylindrical projection parameters.

Set_Equidistant_Cylindrical_Coordinates – This function sets the current input or output Equidistant Cylindrical projection coordinates to the specified values.

Get_Equidistant_Cylindrical_Coordinates – This function returns the current input or output Equidistant Cylindrical projection coordinates.

Set_Lambert_Conformal_Conic_Params – This function sets the current input or output Lambert Conformal Conic projection parameters to the specified values.

Get_Lambert_Conformal_Conic_Params – This function returns the current input or output Lambert Conformal Conic projection parameters.

Set_Lambert_Conformal_Conic_Coordinates – This function sets the current input or output Lambert Conformal Conic projection coordinates to the specified values.

Get_Lambert_Conformal_Conic_Coordinates – This function returns the current input or output Lambert Conformal Conic projection coordinates.

Set_Local_Cartesian_Params – This function sets the current input or output Local Cartesian coordinate system parameters to the specified values.

Get_Local_Cartesian_Params – This function returns the current input or output Local Cartesian coordinate system parameters.

Set_Local_Cartesian_Conic_Coordinates – This function sets the current input or output Local Cartesian coordinates to the specified values.

Get_Local_Cartesian_Conic_Coordinates – This function returns the current input or output Local Cartesian coordinates.

Set_Mercator_Params – This function sets the current input or output Mercator projection parameters to the specified values.

Get_Mercator_Params – This function returns the current input or output Mercator projection parameters.

Set_Mercator_Coordinates – This function sets the current input or output Mercator projection coordinates to the specified values.

Get_Mercator_Coordinates – This function returns the current input or output Mercator projection coordinates.

Set_MGRS_Coordinates – This function sets the current input or output MGRS coordinate string.

Get_MGRS_Coordinates – This function returns the current input or output MGRS coordinate string.

Set_Miller_Cylindrical_Params – This function sets the current input or output Miller Cylindrical projection parameters to the specified values.

Get_Miller_Cylindrical_Params – This function returns the current input or output Miller Cylindrical projection parameters.

Set_Miller_Cylindrical_Coordinates – This function sets the current input or output Miller Cylindrical projection coordinates to the specified values.

Get_Miller_Cylindrical_Coordinates – This function returns the current input or output Miller Cylindrical projection coordinates.

Set_Mollweide_Params – This function sets the current input or output Mollweide projection parameters to the specified values.

Get_Mollweide_Params – This function returns the current input or output Mollweide projection parameters.

Set_Mollweide_Coordinates – This function sets the current input or output Mollweide projection coordinates to the specified values.

Get_Mollweide_Coordinates – This function returns the current input or output Mollweide projection coordinates.

Set_Orthographic_Params – This function sets the current input or output Orthographic projection parameters to the specified values.

Get_Orthographic_Params – This function returns the current input or output Orthographic projection parameters.

Set_Orthographic_Coordinates – This function sets the current input or output Orthographic projection coordinates to the specified values.

Get_Orthographic_Coordinates – This function returns the current input or output Orthographic projection coordinates.

Set_Polar_Stereo_Params – This function sets the current input or output Polar Stereographic projection parameters to the specified values.

Get_Polar_Stereo_Params – This function returns the current input or output Polar Stereographic projection parameters.

Set_Polar_Stereo_Coordinates – This function sets the current input or output Polar Stereographic projection coordinates to the specified values.

Get_Polar_Stereo_Coordinates – This function returns the current input or output Polar Stereographic projection coordinates.

Set_Polyconic_Params – This function sets the current input or output Polyconic projection parameters to the specified values.

Get_Polyconic_Params – This function returns the current input or output Polyconic projection parameters.

Set_Polyconic_Coordinates – This function sets the current input or output Polyconic projection coordinates to the specified values.

Get_Polyconic_Coordinates – This function returns the current input or output Polyconic projection coordinates.

Set_Sinusoidal_Params – This function sets the current input or output Sinusoidal projection parameters to the specified values.

Get_Sinusoidal_Params – This function returns the current input or output Sinusoidal projection parameters.

Set_Sinusoidal_Coordinates – This function sets the current input or output Sinusoidal projection coordinates to the specified values.

Get_Sinusoidal_Coordinates – This function returns the current input or output Sinusoidal projection coordinates.

Set_Transverse_Cylindrical_Equal_Area_Params – This function sets the current input or output Transverse Cylindrical Equal Area projection parameters to the specified values.

Get_Transverse_Cylindrical_Equal_Area_Params – This function returns the current input or output Transverse Cylindrical Equal Area projection parameters.

Set_Transverse_Cylindrical_Equal_Area_Coordinates – This function sets the current input or output Transverse Cylindrical Equal Area projection coordinates to the specified values.

Get_Transverse_Cylindrical_Equal_Area_Coordinates – This function returns the current input or output Transverse Cylindrical Equal Area projection coordinates.

Set_Transverse_Mercator_Params – This function sets the current input or output Transverse Mercator projection parameters to the specified values.

Get_Transverse_Mercator_Params – This function returns the current input or output Transverse Mercator projection parameters.

Set_Transverse_Mercator_Coordinates – This function sets the current input or output Transverse Mercator projection coordinates to the specified values.

Get_Transverse_Mercator_Coordinates – This function returns the current input or output Transverse Mercator projection coordinates.

Set_UPS_Coordinates – This function sets the current input or output Universal Polar Stereographic (UPS) projection coordinates to the specified values.

Get_UPS_Coordinates – This function returns the current input or output Universal Polar Stereographic (UPS) projection coordinates.

Set_UTM_Params – This function sets the current input or output Universal Transverse Mercator (UTM) projection parameters to the specified values.

Get_UTM_Params – This function returns the current input or output Universal Transverse Mercator (UTM) projection parameters.

Set_UTM_Coordinates – This function sets the current input or output Universal Transverse Mercator (UTM) projection coordinates to the specified values.

Get_UTM_Coordinates – This function returns the current input or output Universal Transverse Mercator (UTM) projection coordinates.

Set_Van_der_Grinten_Params – This function sets the current input or output Van der Grinten projection parameters to the specified values.

Get_Van_der_Grinten_Params – This function returns the current input or output Van der Grinten projection parameters.

Set_Van_der_Grinten_Coordinates – This function sets the current input or output Van der Grinten projection coordinates to the specified values.

Get_Van_der_Grinten_Coordinates – This function returns the current input or output Van der Grinten projection coordinates.

Convert – This function converts the current input coordinates, according to the current input datum, coordinate system, and parameters, and the current output datum, coordinate system, and parameters.

Get_Conversion_Errors – This function returns the 90% horizontal (circular), vertical (linear), and spherical error values for the most recent conversion.

Get_Conversion_Status – This function returns the current input or output conversion status.

Get_Conversion_Status_String – This function returns a character string that corresponds to the current input or output conversion status.

Get_Return_Code_String – This function returns a character string that corresponds to the current input or output conversion status.

EXAMPLE APPLICATIONS:

The following example illustrates how the Engine can be used to convert Geodetic coordinates to Mercator projection coordinates, both relative to the WGS 84 datum:

Function Call:

```
long status;  
  
long datum_index;  
  
double ce90, le90, se90;  
  
Geodetic_parameters input_params;  
  
Geodetic_coordinates input_coords;  
  
Mercator_parameters output_params;
```

```

Mercator_coords output_coords;

status = Initialize_Engine();

status = Get_Datum_Index ("WGE", datum_index);

status = Set_Datum (Interactive, Input, datum_index);

status = Set_Coordinate_System (Interactive, Input, Geodetic);

status = Set_Geodetic_Params (Interactive, Input, input_params);

status = Set_Geodetic_Coordinates (Interactive, Input, input_coords);

status = Set_Datum (Interactive, Output, datum_index);

status = Set_Coordinate_System (Interactive, Output, Mercator);

status = Set_Mercator_Params (Interactive, Output, output_params);

status = Convert();

status = Get_Mercator_Coordinates (Interactive, Output, output_coords);

status = Get_Conversion_Errors (Interactive, ce90, le90, se90);

```

Inputs:

input_params.height_type:	Ellipsoid_Height
input_coords.latitude: -	35.0
input_coords.longitude:	75.0
input_coords.height:	0.0.
output_params.central_meridian	0.0
output_params.origin_latitude	0.0
output_params.false_easting	0.0
output_params.false_northing	0.0

Outputs:

output_coords.easting:	-1910804.77
output_coords.northing:	-1463929.01
output_coords.height:	0.0
ce90:	10.0
le90	5.0
se90	12.0