



# (U) GEOTRANS 3.0 Release Notes

December 2009



# (U) GEOTRANS Overview

- (U) Coordinate conversions between the following coordinate systems:
  - (U) Geodetic coordinates
  - (U) Geocentric coordinates
  - (U) Local Cartesian coordinates
  - (U) Twenty-eight different types of map projection-based coordinates
  - (U) Military Grid Reference System (MGRS)
  - (U) United States National Grid (USNG)
  - (U) Global Area Reference System (GARS)
  - (U) World Geographic Reference System (GEOREF) grid coordinates
- (U) Datum Transformations between global or local horizontal datums
  - (U) WGS 84, WGS 72, and 228 local datums
- (U) Conversions between ellipsoid heights and geoid (MSL) heights
  - (U) EGM96 & EGM84 using various grids and interpolation methods



# (U) Migration from GEOTRANS 2.4.x to 3.0

- (U) GEOTRANS 2.4.x (legacy) & 3.0 (MSP) code baselines have been kept in synch
  - (U) Identical functionality
  - (U) As much common source code as possible
- (U) User Interfaces
  - (U) Application GUI – Interactive and Batch Coordinate Conversion
    - (U) No significant differences (Windows/C++ vs. Java implementations)
  - (U) Coordinate File Format – Batch Coordinate Conversion
    - (U) No differences
- (U) Application Programmer Interface (API)
  - (U) Migration from C to C++ allowed significant improvements to API
  - (U) Object-oriented API via Coordinate Conversion Service class
  - (U) Far fewer functions in API (~165 In v2.4.x vs. ~22 In v3.0)
  - (U) Fewer steps required in coordinate conversion operations
  - (U) Errors reported via exception handling
  - (U) Thread safe implementation



# (U) GEOTRANS 3.0 Fixes & Enhancements

- (U) An error was corrected in the MGRS module that had allowed polar format MGRS coordinate strings beginning with the letters C or D to be accepted.
- (U) The MGRS, UTM, and UPS modules were updated to correct several reported problems; these involved the following three issues:
  - (U) MGRS coordinates are now truncated rather than rounded; this has eliminated problems resulting from rounding up to various boundaries,
  - (U) Conversions along the boundaries of the polar regions (84°N and 80°S),
  - (U) Conversions in and on the boundaries of the irregular MGRS zones 31V and 31-37X in the north Atlantic.
- (U) The precision of the supported ellipsoid parameters has been improved based on input from NGA.
- (U) A second variant of the Polar Stereographic projection is now supported which specifies the scale factor at the pole as a parameter. The UPS module has been updated to use this variant of the Polar Stereographic projection.
- (U) A second variant of the Mercator projection is now supported which specifies the scale factor at the equator as a parameter.
- (U) It is now possible specify the order of geodetic coordinates (latitude-longitude or longitude-latitude) in coordinate files, using a new header keyword (COORDINATE ORDER) and new radio buttons in the file processing GUI.
- (U) Heights can be included in input coordinate files containing map projection coordinates; these are passed through without change to the output coordinate file.



# (U) GEOTRANS 3.0 Programming Environment

- (U) The GEOTRANS software was developed and tested using the Microsoft Windows XP operating system, Sun Solaris 8 UNIX, and Red Hat Enterprise 4 LINUX.
  - (U) It should also work on all later versions of these operating systems.
- (U) The GEOTRANS Coordinate Conversion Service was developed in C++.
  - (U) The Windows version was built using Microsoft Visual C++ .Net 2003.
  - (U) The UNIX versions was built using the Sun Forte Workshop 6 Update 2 compiler.
  - (U) The LINUX versions was built using the GNU C++ compiler (gcc, version 3.4.6).
- (U) The GEOTRANS application GUI was developed in Java and requires the Java Runtime Environment (JRE) 1.5 or later to execute
  - (U) MSP recommends using JRE 1.5 Update 18 or later that addresses all of the vulnerability issues in earlier versions of JRE.



# (U) Application GUIs

## (U) GEOTRANS 2.4.x

The figures below are UNCLASSIFIED in their entirety

## (U) GEOTRANS 3.0

**GEOTRANS V2.4.3 - Geographic Translator V2.4.3**

File Options Datum Ellipsoid Convert Help

Datum: WGE: World Geodetic System 1984 Ellipsoid: WE: WGS 84

Geodetic

No Height

Longitude: 0 0 0.0E Latitude: 0 0 0.0N Height (m): 0

90% CE (m): 90% LE (m): 90% SE (m): Sources: Unk N/A N/A Unknown

Convert: Upper -> Lower Convert: Lower -> Upper

Datum: WGE: World Geodetic System 1984 Ellipsoid: WE: WGS 84

Universal Transverse Mercator (UTM)

Zone (1-60): 0 Hemisphere: N S

Override: ☐ Easting / X (m): 500000 Northing / Y (m): 0

90% CE (m): 90% LE (m): 90% SE (m): Sources: Unk N/A N/A Unknown

Ready NUM

**MSP GEOTRANS 3.0**

File Edit Options Datum Ellipsoid Convert Help

Datum: WGE: World Geodetic System 1984 Ellipsoid: WE: WGS 84

Geodetic

No Height

Longitude: 0 0 0.0E Latitude: 0 0 0.0N Height (m): 0

90% CE: 90% LE: 90% SE: Sources: Unk N/A N/A Unknown

Convert Upper -> Lower Convert Lower -> Upper

Datum: WGE: World Geodetic System 1984 Ellipsoid: WE: WGS 84

Universal Transverse Mercator (UTM)

Zone (1-60): 0 Hemisphere: N S

Override: ☐ Easting / X (m): 500000 Northing / Y (m): 0

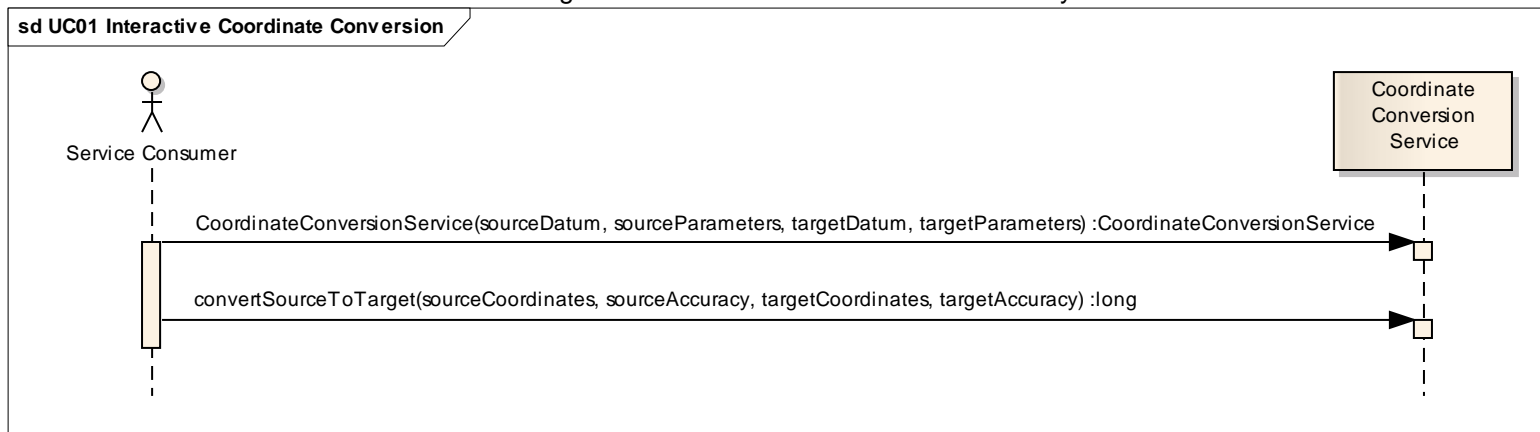
90% CE: 90% LE: 90% SE: Sources: Unk N/A N/A Unknown



# (U) Coordinate Conversion Sequence

1. (U) Construct a Coordinate Conversion object, specifying:
  - a. (U) source datum,
  - b. (U) source coordinate system, including parameters, if any,
  - c. (U) target datum,
  - d. (U) target coordinate system, including parameters, if any.
2. (U) Convert Source to Target, specifying:
  - a. (U) source coordinates,
  - b. (U) source coordinate accuracy,
  - c. (U) target coordinates to be returned,
  - d. (U) target coordinate accuracy to be returned,
  - e. (U) coordinate conversion status to be returned.

The figure below is UNCLASSIFIED in its entirety







# (U) GEOTRANS 3.0 API

The figure below is UNCLASSIFIED in its entirety

## class Coordinate Conversion Service

### CoordinateConversionService

- + CoordinateConversionService(sourceDatum, sourceParameters, targetDatum, targetParameters) : CoordinateConversionService
- + convertSourceToTarget(sourceCoordinates, sourceAccuracy, targetCoordinates, targetAccuracy) : long
- + convertTargetToSource(targetCoordinates, targetAccuracy, sourceCoordinates, sourceAccuracy) : long
- + convertSourceToTargetCollection(sourceCoordinates, sourceAccuracy, targetCoordinates, targetAccuracy, conversionStatus) : long
- + convertTargetToSourceCollection(targetCoordinates, targetAccuracy, sourceCoordinates, sourceAccuracy, conversionStatus) : long
- + getEllipsoidLibrary() : EllipsoidLibrary
- + getDatumLibrary() : DatumLibrary
- + getServiceVersion() : int

### DatumLibrary

- + DatumLibrary() : DatumLibrary
- + defineDatum(datumType, code, name, ellipsoidCode, deltaX, deltaY, deltaZ, sigmaX, sigmaY, sigmaZ, westLongitude, eastLongitude, southLatitude, northLatitude, rotationX, rotationY, rotationZ, scaleFactor) : long
- + removeDatum(code) : long
- + getDatumCount(count) : long
- + getDatumIndex(code, index) : long
- + getDatumInfo(index, code, name, ellipsoidCode) : long
- + getDatumParameters(index, datumType, deltaX, deltaY, deltaZ, sigmaX, sigmaY, sigmaZ, westLongitude, eastLongitude, southLatitude, northLatitude, rotationX, rotationY, rotationZ, scaleFactor) : long

### EllipsoidLibrary

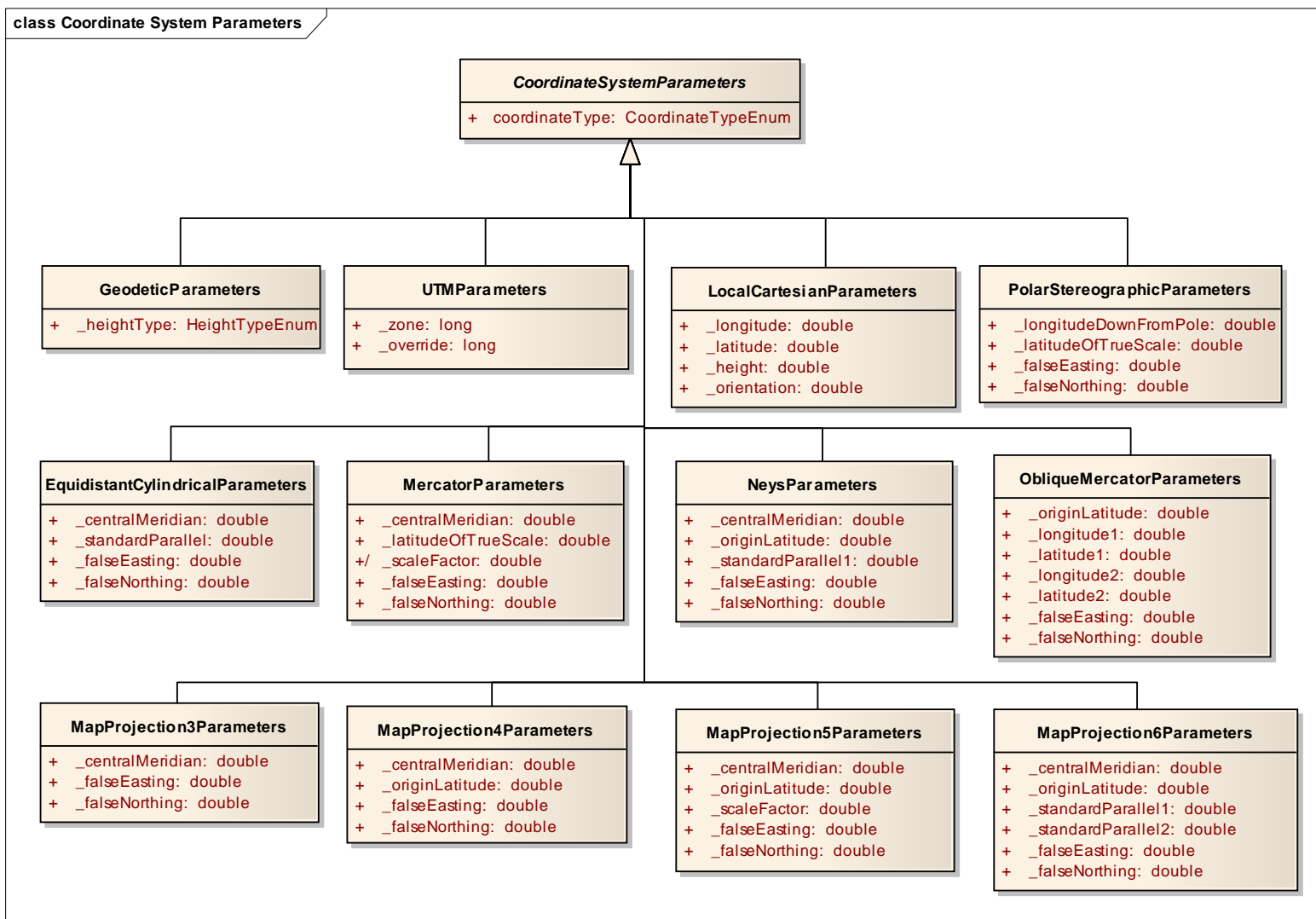
- + EllipsoidLibrary() : EllipsoidLibrary
- + defineEllipsoid(code, name, semiMajorAxis, flattening) : long
- + removeEllipsoid(code) : long
- + getEllipsoidCount(count) : long
- + getEllipsoidIndex(code, index) : long
- + getEllipsoidInfo(index, code, name) : long
- + getEllipsoidParameters(index, semiMajorAxis, flattening) : long





# (U) GEOTRANS 3.0 API (cont'd)

The figure below is UNCLASSIFIED in its entirety





# (U) GEOTRANS 3.0 API (cont'd)

The figure below is UNCLASSIFIED in its entirety

