

# An Open-Source, Object-Oriented General Cartographic Transformation Program (GCTP)

Michael S. Williams<sup>a,\*</sup>, Michael P. Finn<sup>a</sup>, and Robert A. Buehler<sup>a,1</sup>

<sup>a,\*</sup>U.S. Geological Survey (USGS)  
National Geospatial Technical Operations Center  
1400 Independence Road  
Rolla, Missouri 65401

[mwilliams@usgs.gov](mailto:mwilliams@usgs.gov)

## Commission IV, WG IV/3

**KEY WORDS:** Cartographic Transformation, Object-Oriented Program, Map Projection, Open-Source Code, Geospatial Data

### EXTENDED ABSTRACT:

A version of the USGS' General Cartographic Transformation Program (GCTP) (USGS, 1994) that adheres to object oriented software development concepts (Booch, 1994) and is compliant with the definition of open source (Open Source Initiative (OSI), 2005) has been developed. The basis of this effort is the GCTP created by the USGS' National Mapping Division in the 1960's and maintained through the mid-1990s. The program is designed to perform coordinate pair transformation between any two map projections. Originally written in the FORTRAN programming language, GCTP was later converted to C using an automated translator. It currently (2006) has parametric code for 30 map projections. Although still technically useful, the program required several major feature modifications to bring it up to current practices for modern programming languages and applications programs. For example, GCTP previously required an output file for writing status and error reports, and the most common method of performing the coordinate transformations (calling the **gctp** function) resulted in redundant code execution.

In our new object-oriented version of GCTP, written using C++, every projection is contained in its own class, with each class inheriting from a base class called **Projection**. Each projection uses the **forward** and **inverse** functions to perform the actual coordinate transformations. A new way to set the parameters for each projection through special set functions is included (yet allowing the old C-based manner to set parameters); each projection has set functions to set its parameters, as well as functions to retrieve particular parameter values. If an error occurs in any part of a coordinate transformation, a flag is set and the **error** function can be called to retrieve the code.

The original GCTP supported a projection type called USDEF (user defined) that allowed the user to write his or her own custom projections to use within the GCTP framework. Users can easily define their own projection in the new GCTP, or GCTP\_CPP, simply by implementing a new projection object that inherits from the base **Projection** class and implements three pure virtual functions. This can be thought of as extensibility by function overriding (see, for example, Stroustrup, 1997).

The GCTP\_CPP has been tested and continues to undergo testing to ensure that it is comparably as robust as the previous GCTP code. This testing is meant to provide the user community with as much confidence in the GCTP\_CPP application

---

\* Corresponding author: phone: 1-573-308-3931, fax: 1-573-308-3652

<sup>1</sup> Now with University of Missouri – Rolla, Department of Computer Science

Any use of trade names in this publication is for descriptive purposes only and does not constitute endorsement by the U.S. Government.

programming interface (API) as possible. This was a complete object-oriented redesign, and not merely just another C++ wrapper around the old code.

All the underlying mathematical code used to perform the map projections is carried forward from the original GCTP. The testing to date includes, in addition to visual testing of map projections using new and old code, a testing framework that feeds the information contained in extensible-markup-language-based configuration files to both APIs to perform the projection transformations. The testing framework builds a base configuration file describing an image in geographic space to reproject to each of the supported projections, reprojects every point of the geographic image into each supported projection using GCTP\_CPP and the original GCTP, and checks that the forward and inverse transformations performed by the new and the old APIs produce the same output coordinates.

## REFERENCES

Booch, Grady (1994). *Object-Oriented Analysis and Design with Applications*, Second Edition. Reading, Massachusetts: Addison-Wesley.

OSI (2005). *OSI*. Internet at: <http://www.opensource.org/>. Last Accessed 14 September 2005.

Stoustrup, Bjarne (1997). *The C++ Programming Language*, Third Edition. Reading, Massachusetts: Addison-Wesley.

USGS (1994). *GCTP Software Documentation*. Internet at: [http://geography.usgs.gov/ftp/software/current\\_software/gctp/gctp-sd.txt](http://geography.usgs.gov/ftp/software/current_software/gctp/gctp-sd.txt). Last Accessed 14 September 2005.