NORTH ROAD
CARTOGRAPHY
DEVELOPMENT
SPATIAL ANALYSIS

north-road.com
info@north-road.com
ABN 80 769 844 078
North Road Consulting Pty Ltd

On GDA2020, PROJ 6 and QGIS: Lessons learnt and recommendations for handling GDA2020 within geospatial software development

Author: Nyall Dawson
North Road Consulting Pty Ltd

Executive Summary

This whitepaper was commissioned by the ICSM (Intergovernmental Committee on Surveying and Mapping, https://www.icsm.gov.au/) to outline recent software developments related to the adoption and use of the GDA2020 coordinate reference systems.

GDA2020 (Geocentric Datum of Australia 2020) is a modernization of Australia's spatial datum, replacing the prior GDA94 standard. GDA2020 adoption requires significant updates and changes in the way geospatial software applications transform coordinates in order to ensure accurate results.

ICSM has been able to leverage the independent work done by the PROJ library maintainers as part of the "GDAL SRS barn raising effort", and has funded investment into the QGIS Desktop GIS application to utilise this work and ensure that users of the application have an accurate and easy to use experience with GDA2020 related data.

This paper is targeted to managers and developers of spatial software projects, who have at least a background understanding of geospatial and datum concepts. It describes the motivation behind the software changes, and outlines the recommendations from QGIS software developers which other software applications may follow in adapting their software for GDA2020 compliance.

Specific recommendations are:

north-road.com
info@north-road.com
ABN 80 769 844 078
North Road Consulting Pty Ltd

1. Always use registered authority identifiers and codes when handling, storing, and retrieving Coordinate Reference System (CRS) objects, and avoid other representations of CRS parameters such as “proj” strings.
2. Application installers should include the *proj-datumgrid-oceania* grid transformation files by default, or provide an easy to use method for users to install these, without requiring administrative permissions.
3. Applications should explicitly advise users when grid transformation files would result in a more accurate coordinate transform, yet they are not available for use on their system - avoiding any prerequisite knowledge of the use or the existence of these files on the user’s behalf.
4. Wherever possible, the desired area of use for coordinate transformations should be specified a priori when constructing coordinate transformation operations
5. Applications should provide a user-friendly means for knowledgeable users to manually override the default transformation operations applied when converting to or from GDA2020 coordinates.

Introduction

In 2017, the Australian ICSM organisation (Intergovernmental Committee on Surveying and Mapping, https://www.icsm.gov.au/) released the specifications for a new, modern Australian spatial datum named “*GDA2020*” (ICSM - Geocentric Datum of Australia 2020). This new datum was introduced in direct response to demand for improved accuracy in the availability and use of spatial data coordinates within Australia.

As part of the program of works surrounding the introduction of GDA2020, the ICSM funded investment into the open source geospatial software community to improve handling of GDA2020 related data and coordinate transformation. Specifically, this investment involved funding North Road Consulting Pty Ltd (http://north-road.com) to implement a package of works aimed at upgrading the “QGIS” desktop GIS application (https://qgis.org) to correctly handle the GDA2020 reference systems, and to use the experiences and lessons encountered during this upgrade as a pathway that other software developers could follow in upgrading their applications to correctly handle GDA2020 and the needs of the Australian geospatial community.

In this paper, we describe the background behind this work, what was involved in upgrading QGIS to handle GDA2020, and the recommendations we have for other software developers looking to ensure correct GDA2020 compliance within their software.

Complications involving GDA2020

north-road.com
info@north-road.com
ABN 80 769 844 078
North Road Consulting Pty Ltd

The introduction of the GDA2020 coordinate reference systems bring new complications in handling coordinate transformations within geospatial software, and care must be taken to correctly adapt software in order to satisfy the accuracy demands of modern spatial data stakeholders.

One such complication arises from the fact that currently the GDA2020 reference systems are defined in relation to GDA94 reference systems only. The official parameters and transformation support files released by the ICSM can only be used when transforming between GDA94 geographic coordinates and GDA2020 geographic coordinates, or vice versa. Accordingly, transforming between other related coordinate systems, such as from Vicgrid GDA94 to GDA2020 requires "pivoting" the coordinates first through the GDA94 geographic to GDA2020 geographic transformation.

As a specific example, a transformation from Vicgrid GDA94 (identified by EPSG code 3111) to Vicgrid GDA2020 (EPSG code 7899) requires pivoting through the GDA94 geographic (EPSG code 4283) to GDA2020 geographic (EPSG code 7844) Coordinate Reference Systems (CRS). The steps involved are:

1. Transforming the GDA94 Vicgrid coordinates to GDA94 geographic coordinates
2. Transforming the GDA94 to GDA2020 geographic coordinates (via the official ICSM transformation parameters)
3. And finally transforming from GDA2020 geographic to Vicgrid GDA2020

Performing this transformation correctly requires that software applications:

1. Are able to correctly determine the correct steps involved in the transformation, based on the source and destination CRS alone, and
2. Allow pivoting via a specific datum (in this case GDA94), instead of via WGS84 or another semi-standard pivot[1].

In a fortunate parallel development, the growing global use of complex coordinate operations (such as those involved in GDA2020) triggered large changes within the software responsible for handling coordinate transformations. One such significant change occurred within the low-level "PROJ" transformation library, which brought many benefits to stakeholders within the Australian geospatial community.

The PROJ library and the GDAL "barn raising" efforts

The PROJ library (https://proj.org/, previously known as "PROJ.4") is a "*generic coordinate transformation software that transforms geospatial coordinates from one Coordinate Reference System (CRS) to another*" (PROJ 2019). Since its initial release in the early 1980s (Evenden 1983), PROJ has become a mature, well-respected library, and is widely used as the backend for geographic transformation handling within many open-source and proprietary geospatial applications.

[1] As well as being a pre-requisite for transformation to and from GDA2020 based reference systems, allowing non-WGS84 pivot datums avoids unnecessary loss of accuracy during coordinate transforms

north-road.com
info@north-road.com
ABN 80 769 844 078
North Road Consulting Pty Ltd

Previously, the PROJ library itself did not have the required information or logic to handle complex operations such as those required for the GDA2020 coordinate reference systems. Downstream users of PROJ, which include most open source geospatial software and large commercial vendors such as Safe Software, SAP SE, and Oracle (amongst many others) would have to implement their own individual versions of this logic and manually handle complex situations like non-WGS84 pivot datums themselves. Aside from the duplication of effort required, coordinate transformation is an incredibly (and increasingly) complicated field, requiring in-depth knowledge of geodesy and international standards. Getting this logic correct was extremely difficult, leading to inaccuracies and discrepancies between the transformation results from software to software.

The maintainers of the PROJ library were long aware of these issues, and had worked to implement the low-level building bricks necessary for handling complex geodetic transformations (see "Transformation pipelines" below). With the release of PROJ version 5 in February 2018 (PROJ 5.0.0 Release Notes 2018), these building bricks were made available for general use, although at this stage they were mostly useful for geodetic expert users only. To improve the accessibility for the casual user, and to increase the interoperability with international standards and conventions for the representation of geodetic systems and transformations, a number of key PROJ stakeholders undertook a large fund-raising drive in order to cover the extensive low-level work required to update PROJ to handle these use cases within the PROJ library itself. The fund raising drive was entitled the "*GDAL SRS barn raising effort*" (https://gdalbarn.com/), reflecting the PROJ library's close links to the GDAL Geospatial Data Abstraction Library (https://gdal.org/). The drive successfully hit its target in May 2018 (Butler, Ramsey, Evers and Rouault 2018), in part due to the contributions of some of the large commercial vendors who rely on the library. The improvements were made publicly available in the PROJ version 6.0.0 release in March 2019 (PROJ 6.0.0 Release Notes 2019).

Transformation pipelines

Of particular note to the Australian geospatial community, PROJ version 5.0 introduced the concept of "*transformation pipelines*" (Knudsen & Evers 2017, Evers & Knudsen 2017). Pipelines allow complex geodetic transformations by expressing operations via a series of underlying component steps, such as unit conversions, transformations, axis swaps and projections. Accordingly, complex transformations such as that between Vicgrid GDA94 (EPSG:3111) and Vicgrid GDA2020 (EPSG:7899) can be completely represented via the transformation pipeline:

```
+proj=pipeline
  +step +inv +proj=lcc +lat_0=-37 +lon_0=145 +lat_1=-36 +lat_2=-38
  +x_0=2500000 +y_0=2500000 +ellps=GRS80
  +step +proj=push +v_3
  +step +proj=cart +ellps=GRS80
  +step +proj=helmert +x=0.06155 +y=-0.01087 +z=-0.04019
  +rx=-0.0394924 +ry=-0.0327221 +rz=-0.0328979 +s=-0.009994
  +convention=coordinate_frame
  +step +inv +proj=cart +ellps=GRS80
```

north-road.com
info@north-road.com
ABN 80 769 844 078
North Road Consulting Pty Ltd

```
+step +proj=pop +v_3
+step +proj=lcc +lat_0=-37 +lon_0=145 +lat_1=-36 +lat_2=-38
+x_0=2500000 +y_0=2500000 +ellps=GRS80
```

Further benefiting downstream users of the PROJ library, version 6.0 also introduced a comprehensive database containing the information stored in several CRS and transformation registries, including the authoritative EPSG registry (EPSG 2019, https://www.epsg-registry.org/). This database contains all the metadata and information required in order to allow PROJ to automatically calculate the potential candidate transformation pipelines (or "*coordinate operations*") between a source and destination CRS, and includes API calls allowing applications to query and expose the metadata regarding these coordinate operations.

In short, by utilising the new functionality introduced in PROJ version 6.0, developers of geospatial software can easily, confidently, and consistently retrieve the optimal coordinate operations and transforms between coordinate reference systems without in-depth knowledge of the low-level geodesic or standards requirements previously required, and accordingly ensure that their applications correctly handle GDA2020 coordinate systems.

ICSM Sponsorship of GDA2020 adoption in open source geospatial applications

Following the release of PROJ version 6, ICSM funded work in upgrading the open-source desktop GIS application "QGIS" to utilise the new functionality available in PROJ 6, and to ensure that Australian users of the application would have an optimal experience when working with data in the GDA2020 reference system. Specifically, the goals of this work were to ensure that QGIS users:

1. Have access to all required tools and support files desirable for GDA2020 related transforms in the default QGIS install packages
2. Would obtain the most accurate results possible when transforming coordinates to or from GDA2020 references systems, without any prerequisite geodetic or GDA2020 knowledge
3. Would always be warned (in a user-friendly way) when the preferred transforms to or from GDA2020 references systems could not be used, and given user-friendly steps to remedy the issue

QGIS is a free and open source Geographic Information System, available for use with desktop, server and mobile clients and on a wide range of operating systems (QGIS 2019, https://qgis.org). It is arguably the most widely used open-source desktop GIS client available today. QGIS was selected for investment by ICSM due to the open-source nature of the application, and for the wider-reaching benefits to other open-source geospatial software projects (including PROJ, GDAL, and other open-source client software). QGIS was seen as an ideal "test case" for determining what was required in making sure other geospatial software is "GDA2020 ready".

The ICSM funded work was completed in June 2019 (Dawson 2019) by North Road Consulting Pty Ltd (https://north-road.com), and was made publicly available with the QGIS 3.8.0 release

north-road.com
info@north-road.com
ABN 80 769 844 078
North Road Consulting Pty Ltd

(Changelog for QGIS 3.8 2019). The work was completed within the public QGIS code repository (https://github.com/qgis/QGIS, available under a GPL2 or later license), and was accompanied by required changes in the PROJ library (https://github.com/OSGeo/PROJ - partly included in PROJ 6.1.1, partly scheduled for future inclusion in PROJ 6.2.0 in September 2019).

The remainder of this paper outlines the steps taken to upgrade QGIS to utilise PROJ version 6, the main differences between the library versions, and how PROJ 6's metadata capabilities can be used to expose important transformation information to users. Technical challenges encountered during the QGIS upgrade are also detailed, along with the steps taken to overcome or avoid these challenges. It is the author's belief that by following the experiences of the QGIS upgrade, other software developers can implement a "best practice" approach in making software GDA2020 compliant.

Recommendations for porting applications to PROJ version 6

While existing guides such as the PROJ team's "Version 4 to 6 API Migration" (PROJ 2019) and PROJ.6 (Spatialite 2019) cover the general considerations in porting applications to PROJ version 6 well (and are recommended initial reading), this guide will focus on considerations specific to the GDA2020 transformations and how best to utilise PROJ 6 to cover the use cases for Australian users.

Consideration 1: identifying coordinate reference systems

Applications should prioritize using the authority and code identifiers for particular CRSes, falling back to WKT definitions where an authority is not available.

Previous PROJ versions heavily utilised "*proj*" definitions of CRSes, which are custom strings used only by the PROJ library.
For instance, the proj string definition of Vicgrid GDA94 is:

```
+proj=lcc +lat_1=-36 +lat_2=-38 +lat_0=-37 +lon_0=145 +x_0=2500000
+y_0=2500000 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs
```

However, a proj string representation of a CRS is a lossy format, and their use is discouraged for version 6. In order to fully utilise PROJ's new CRS registry database and metadata, developers should utilise known authority identifiers of CRSes wherever possible (e.g. "EPSG:3111" for Vicgrid GDA 94, where EPSG is the associated registry authority and 3111 is the unique identifier of the CRS within that registry).

Internally, PROJ uses these identifiers to lookup the associated metadata from the registry databases, e.g. to determine the recorded accuracy of transformations associated with the CRS, and accordingly create the optimal coordinate transformation for this CRS.

north-road.com
info@north-road.com
ABN 80 769 844 078
North Road Consulting Pty Ltd

In PROJ 6, creating a CRS from an authority and identifier is done using the proj_create_from_database API, for instance:

```
PJ* crs = proj_create_from_database( 0, "EPSG", "3111",
PJ_CATEGORY_CRS, false, 0 );
```

Wherever an authority is not associated with a particular CRS, the WKT2 definition of the CRS is preferred over a proj string definition. Unlike proj strings, WKT2 definitions are not lossy, and still provide sufficient information to the PROJ library for it to make an informed choice of coordinate operations using the CRS.

```
PJ* crs = proj_create_from_wkt( 0, "PROJCRS[\"GDA94 / Vicgrid\",
BASEGEOGCRS[\"GDA94\"...", 0, 0 );
```

This consideration extends to how applications themselves store and restore CRS definitions. For example, the QGIS desktop application allows users to set a particular CRS for each mapping project they create. When a mapping project is stored to a disk-based format, QGIS stores the current project CRS by recording the authority and identifier, rather than storing a proj string representation of this CRS. Accordingly, when an existing project is restored, the PROJ CRS object can be re-created using the recorded authority and identifier, ensuring no loss of information.

By correctly prioritizing the non-lossy representations of CRS objects, applications will ensure that PROJ will always make informed choices relating to the correct pivot datums to utilise in coordinate transformations, and as a result ensure that Australian users will obtain correct transforms when GDA2020 based CRSes are involved.

Consideration 2: availability of transformation grids

Where possible, application installations should include the proj-datumgrid-oceania package, or allow users to manually install grid transform files to a user-writable location

Depending on the origin of coordinates being transformed, accurate transformation of pre-GDA2020 coordinates to GDA2020 CRSes (or vice versa) may require use of a "*transformation grid*" file. These transformation grids have been made available by the ICSM, and are used to model local distortion which must be applied when transforming to or from GDA2020 reference systems. While the PROJ library is fully capable of utilising these transformation grid files, they unfortunately are **NOT** provided with the typical PROJ library install. Rather, the large size of these files has resulted in them being provided as a separate package for download and install from the PROJ maintainers.

Specifically, the grid transformation files required for use with GDA2020 coordinates are provided in the "*proj-datumgrid-oceania"* package (see https://github.com/OSGeo/proj-datumgrid/tree/master/oceania or available for direct download at

north-road.com
info@north-road.com
ABN 80 769 844 078
North Road Consulting Pty Ltd

https://download.osgeo.org/proj/proj-datumgrid-oceania-latest.zip). In order for these grid transformation files to be utilised by PROJ, they must be placed in the PROJ_LIB folder, alongside the proj.db file.

The QGIS application installers for Microsoft Windows and MacOS now always include the *proj-datumgrid-oceania* packages **by default.** This ensures that they are available for use by any QGIS user (worldwide) working with GDA2020 reference systems, and removes any pre-requisite knowledge on the user's behalf of the existence or importance of these transformation support files.

Unfortunately, external constraints (such as the large file size of the grid files), may prevent them from being included in the default install of a software application on some platforms. In this case, it is recommended that application developers provide an option for users to place these grid transformation files in a user-writable location, allowing these users to manually install the files and have them available for use in the application, without requiring administrator privileges to do so.

In QGIS, this is exposed via a user-writable "proj" folder which sits within the QGIS application settings directory under the current user's profile. Since this path differs from the default PROJ_LIB folder, the proj_context_set_search_paths API is used to append the path to PROJ's internal list of known folders in which to search for grid transformation files. To do so, first the existing list of search paths is obtained via a call to proj_info().searchpath. The new, user-writable, path is appended to the end of the existing path list, and then proj_context_set_search_paths called during the application initialization to ensure that any manually installed grid files will be available for use during transformations.

Consideration 3: Advise users when grid transformation files are desirable

Applications should notify users when more accurate transformations would be possible, but are not available due to missing grid transformation files

By default, the PROJ 6 API automatically selects the best possible **available** coordinate operation to use when transforming between a source and destination CRS. In some cases, the default operation created by PROJ may not represent the preferred operation, but rather a secondary choice only. The most common cause of this is due to missing (or incorrectly installed) grid transformation files.

Instead of silently proceeding to utilise a known inferior coordinate operation, it is our recommendation that instead applications should check whether the preferred operation is unavailable, and if so, present a warning to users and advise them of how they can remedy the situation by installing the missing grid transformation file. The QGIS desktop application does this by showing a warning bar in the main application window, alerting users that the preferred transformation was unavailable. Clicking the warning displays a dialog explaining the situation in greater detail, along with links to download the corresponding *proj-datumgrid* package (where available).

north-road.com
info@north-road.com
ABN 80 769 844 078
North Road Consulting Pty Ltd

In order to detect this situation, it is necessary for applications to:

1. Create a new PROJ PJ_OPERATION_FACTORY_CONTEXT via a call to `proj_create_operation_factory_context`. E.g.

   ```
   PJ_OPERATION_FACTORY_CONTEXT *operation_factory_context =
   proj_create_operation_factory_context( 0, 0 );
   ```

2. Configure the factory to ignore grid file availability when calculating available coordinate operations between a pair of CRSes:

   ```
   proj_operation_factory_context_set_grid_availability_use( 0,
   operation_factory_context, PROJ_GRID_AVAILABILITY_IGNORED );
   ```

3. Create a list of the available operations between the source and destination CRS objects, using the configured PJ_OPERATION_FACTORY_CONTEXT:

   ```
   PJ_OBJ_LIST *ops = proj_create_operations( 0, source_crs,
   dest_crs, operation_factory_context )
   ```

4. Iterate through the returned list of available operations using `proj_list_get_count` and `proj_list_get`. This list will be ordered by preference - so the most preferred coordinate operation will be the first in the list.
5. Test whether the operation is available for use, via a call to `proj_coordoperation_is_instantiable`. If the first returned operation is available, it indicates that the preferred coordinate operation to use between the source and destination CRS is available for use on the system.
6. If, however, the first returned operation is not instantiable, it is likely due to the use of a missing grid transformation file. In this case, the application should iterate through the grids used by the operation via calls to `proj_coordoperation_get_grid_used_count` and `proj_coordoperation_get_grid_used`. The `proj_coordoperation_get_grid_used` API helpfully provides a means for retrieving the name, download URL, software license, and other metadata for missing grid files. These details can be used to provide helpful warning messages to users, which also give the user the steps required to download and install the missing grid files.

By warning users when more accurate transformations between a source and destination CRS are available, we ensure that **all** users (including those relying on GDA2020 based coordinate systems) are informed of the cause of possible inaccuracies in their data handling, not just those who are previously informed and hold knowledge of the available operations for transforming between systems.

north-road.com
info@north-road.com
ABN 80 769 844 078
North Road Consulting Pty Ltd

Consideration 4: Specifying area of use when creating coordinate operations

In order to correctly determine the best operation to use when transforming to or from GDA2020 coordinate systems, the correct area of use of a transform must be specified in advance.

While the PROJ library does its best to always select the optimal coordinate operation to use when transforming between reference systems, its ability to do so is improved by providing additional context to the library about how the operation will be used in advance. Specifically, when calculating the preferred coordinate operation, PROJ considers the available area of use for candidate operations. By specifying in advance a bounding box indicating the area where coordinate transforms will be performed, PROJ is able to best use this area of use information to rank possible candidate operations.

In particular, this is known to affect the ordering of GDA2020 based coordinate operations (Rouault 2019), so care must be taken to ensure correct results for Australian GDA2020 based operations.

Within PROJ API, this is achieved by calling `proj_operation_factory_context_set_area_of_interest` on a PJ_OPERATION_FACTORY_CONTEXT prior to using that factory context to create coordinate operations. `proj_operation_factory_context_set_area_of_interest` allows specification of a bounding box (via latitude and longitude) indicating the desired area of interest for resulting coordinate transformations.

In the case where this information is **not** available in advance (e.g. the QGIS desktop application must create coordinate transforms before the area of use information is available), the `proj_operation_factory_context_set_spatial_criterion` API should be called to allow partial intersections to be considered when calculating operation candidates. E.g.:

```
proj_operation_factory_context_set_spatial_criterion( 0,
operation_factory_context,
PROJ_SPATIAL_CRITERION_PARTIAL_INTERSECTION );
```

Failure to do so may result in inferior coordinate operations being selected for use when transforming to or from GDA2020 based reference systems.

Consideration 5: Allow knowledgeable users to manually select from available coordinate operations

Whilst the PROJ library does a great job at automatically determining preferred candidates for coordinate operations, there are situations where user-held knowledge may alter the desired operation used for a particular transformation.

north-road.com
info@north-road.com
ABN 80 769 844 078
North Road Consulting Pty Ltd

Again, this is of particular importance to users of GDA2020 based reference systems. Specifically, depending on the origin of a particular dataset, either a conformal grid transformation should be used (coordinates observed from Global Navigation Satellite System (GNSS) technology) or a conformal + distortion grid should be used (coordinates recorded using survey control marks for referencing) (ICSM 2017).

Since the PROJ library is not aware of the original source of datasets being transformed, this choice can **only** be made by an informed user. Accordingly, it is our recommendation that application developers consider exposing a method for asking users to select from available coordinate operations whenever multiple possible operations exist for transforming between a source and destination CRS, and allow users to override the default selected operation when required.

In QGIS, this is exposed via a user-defined setting controlling whether to show a list of available operations whenever multiple operations are available for a particular transformation. When this situation arises, the user is shown a dialog listing all available operations, along with their determined accuracy, area of use, and other available metadata allowing them to make an informed choice.

In particular, the PROJ API for retrieving operation scope and remarks is invaluable to Australian users, as these scope and remark metadata fields contain the descriptions of when users would prefer one operation over the other.

This can be done by the PROJ API via:

1. Creating a factory context and iterating over available operations, as described in Consideration 3.
2. Retrieving the operation accuracy via `proj_coordoperation_get_accuracy`
3. Retrieving the available area of use for the operation via `proj_get_area_of_use`
4. Retrieving the scope and remarks for the operation via `proj_get_remarks` and `proj_get_scope` (requires PROJ 6.2 or later)
5. Retrieve an internal definition of the coordinate operation for later use via a call to `proj_as_proj_string`. Note that unlike the limitations of a proj string definition of a CRS described in Consideration 1, the proj string definition of a coordinate operation is non-lossy and the recommended way to store and retrieve these objects.
6. Collating this information into a user-friendly list, sorted in the order returned from the PROJ API (i.e. sorted with PROJ's preferred operation listed first)
7. Allow users to select their desired operation from the list, and re-creating the desired PROJ coordinate operation from its proj string representation via `proj_create`.

Difficulties encountered while adapting QGIS to PROJ version 6

While the previous section describes recommendations specific to best-practice use of PROJ version 6 within a GDA2020 framework, we encountered a number of difficulties whilst porting QGIS to PROJ 6 which are worthwhile discussing here. The solutions to these issues were

north-road.com
info@north-road.com
ABN 80 769 844 078
North Road Consulting Pty Ltd

resolved whilst in close contact with the developers and team behind the PROJ library, ensuring that the correct solutions to the issues were identified.

Difficulty 1: Handling axis order within PROJ 6

Unlike previous versions of the PROJ API, version 6 no longer forces coordinate systems into an X/Y ordering of coordinate pairs. Instead, the ordering of coordinates is dependant on the definition of the CRS in which these coordinates are represented. In particular, this means that instead of representing geographic coordinates as (longitude, latitude) values, these may be represented in PROJ 6 as (latitude, longitude).

Depending on the end-user application, this may have large flow on effects, especially given that many geospatial applications strictly adhere to an X/Y coordinate order. To adapt applications, developers will either need to:

1. Query the PROJ CRS object to determine the actual ordering of coordinates expected for that CRS, and ensure that coordinates follow this order. This is achieved by calling `proj_cs_get_axis_count` and `proj_cs_get_axis_info` on the CRS object, and checking the result of the out_direction argument (e.g. comparing it to the string “north”, which indicates that latitude coordinates would come before longitude for the CRS).
2. Alternative, `proj_normalize_for_visualization` API can be used to attempt to convert a coordinate operation calculated by PROJ into a typical X/Y ordering operation. The operation returned by `proj_normalize_for_visualization` will take and return coordinates in an X/Y order, avoiding the need to manually handle other ordering. (Note that there are exceptional cases where the output from `proj_normalize_for_visualization` does not follow X/Y ordering, such as tilted CRSes or CRSes with coordinates expressed in geocentric Cartesian coordinates. Special care must be taken when handling these edge cases.).

Depending on the wider context of an application, either approach may be preferable. When porting the QGIS desktop application to PROJ 6, we made use of the `proj_normalize_for_visualization` approach, as QGIS rigorously enforces X/Y coordinate ordering throughout the rest of its internal API.

Difficulty 2: Handling PROJ CRS objects created from a range of inputs

The second considerable difficulty we encountered while porting QGIS resulted from the range of possible object types which may be returned when creating a PROJ CRS from a particular input. In particular, alongside the choice of standard CRSes from a registered authority (such as EPSG), QGIS allows users to manually create their own CRS definitions using a range of formats, including proj strings or WKT text strings.

Over time, QGIS has built up an extensive unit test suite covering a whole range of custom user-crafted CRS definitions. After porting QGIS’ internal CRS handling to PROJ 6, many of these unit tests were failing for non-obvious reasons.

north-road.com
info@north-road.com
ABN 80 769 844 078
North Road Consulting Pty Ltd

Consultation with the PROJ team eventually determined that the root cause of these issues was that PROJ has numerous types of CRS objects which can be crafted from proj strings or WKT text input, some of which have particular considerations which affect how other parts of the PROJ API handle these objects.

In the end, the solution employed inside QGIS involved wrapping all PROJ CRS objects created from free-form user input in a custom "crsToSingleCrs" function, which:

1. Calls `proj_get_type` in order to determine the type of CRS object was created by PROJ.
2. If the type is PJ_TYPE_BOUND_CRS (a BoundCrs object), take only the source CRS using `proj_get_source_crs`
3. If the type is PJ_TYPE_COMPOUND_CRS (a CompoundCrs object), iterate through the sub CRS components using `proj_crs_get_sub_crs`, and testing the type of each sub CRS (using `proj_get_type`) until a type which is **not** at PJ_TYPE_VERTICAL_CRS or PJ_TYPE_TEMPORAL_CRS is found, and taking this sub CRS.
4. For all other types, use the original CRS object created by PROJ from the user definition.

While complex, this approach resolved all existing test failures, and ensured correct handling of custom, user-created CRS definitions worked correctly.

Future considerations

While the process described in this paper specifically covers the static GDA2020 reference systems, Australia has further complications looming on the horizon with the upcoming introduction of a temporal, dynamic datum. This datum will take advantage of the relatively stable motion of Australia's tectonic plate in order to model the plate movement based on a temporal dimension (ICSM 2019 - Upgrades to the Australian Geospatial Reference System).

The introduction of a temporal, dynamic datum such as this is a pioneering effort from the ICSM, and introduces many additional challenges and considerations within the development and use of geospatial software.

Encouragingly, PROJ version 6 has introduced support for temporal dimensions and dynamic datums, so applications which port to PROJ version 6 will be well-placed for the future changes required to correctly handle time-dependent coordinate transformations.

It is our recommendation that application developers consider this future work while porting applications to PROJ version 6. In the case of QGIS, a new class for storing the wider context of coordinate transforms was created and is utilised whenever a coordinate transform is created within the software. In future, the context will be populated with additional information regarding the source and destination epoch, reflecting the reference dates and times at which source coordinates were recorded, and the desired reference date and time at which output transformed coordinates should be modelled. This information will be passed to the PROJ library to utilise while

north-road.com
info@north-road.com
ABN 80 769 844 078
North Road Consulting Pty Ltd

performing time-dependent coordinate operations, and will be populated in future based on pending standards regarding storage and handling of epoch related information in geospatial datasets. Creating the basis for the transform coordinate class in advance allows QGIS to prepare for time-dependent transforms, ensuring that no break in their stable API will be required to handle this situation.

Conclusion

Thanks to the support of ICSM, the QGIS desktop application has been upgraded to fully utilise the new capabilities made available in PROJ version 6, and is now fully ready to accurately and easily handle transformations involving the GDA2020 reference systems.

It is the belief of the ICSM, and the authors of this paper, that the steps outlined above will give other application developers a clear pathway to follow while upgrading their applications to ensure that their particular applications correctly handle GDA2020 and the requirements of Australian users.

References

Butler, Ramsey, Evers and Rouault (2018). GDAL SRS Barn Raising. Retrieved from https://gdalbarn.com/

Dawson (2019) QGIS and Proj 6. Retrieved from https://lists.osgeo.org/pipermail/qgis-developer/2019-June/057525.html

EPSG 2019, EPSG Geodetic Parameter Registry. Retrieved from https://www.epsg-registry.org/

Evenden, Gerald I. (1983). Forward and inverse cartographic projection procedures. Open-File Report 83-625. U.S. Geological Survey. p. 76. Retrieved 2015-08-14.

Evers, Kristian and Thomas Knudsen, 2017. Transformation pipelines for PROJ.4. In FIG Working Week 2017 Proceedings. Retrieved from https://www.fig.net/resources/proceedings/fig_proceedings/fig2017/papers/iss6b/ISS6B_evers_knudsen_9156.pdf

ICSM 2017 - GDA94 to GDA2020 transformation grids, ICSM, December 2017. Retreived from https://www.icsm.gov.au/sites/default/files/DatumMattersT1FactSheet.pdf

ICSM 2019 - Geocentric Datum of Australia 2020. Retrieved from https://www.icsm.gov.au/gda2020

north-road.com
info@north-road.com
ABN 80 769 844 078
North Road Consulting Pty Ltd

ICSM 2019 - Upgrades to the Australian Geospatial Reference System. Retrieved from https://www.icsm.gov.au/upgrades-australian-geospatial-reference-system

Knudsen, Thomas and Kristian Evers, 2017: Transformation pipelines for PROJ.4. Geophysical Research Abstracts, Vol. 19, EGU2017-8050. Retrieved from https://meetingorganizer.copernicus.org/EGU2017/EGU2017-8050.pdf

PROJ (2018). 5.0.0 Release Notes. Retrieved from https://proj.org/news.html#proj-5-0-0

PROJ (2019). 6.0.0 Release Notes. Retrieved from https://proj.org/news.html#id41

PROJ (2019). PROJ Homepage. Retrieved from https://proj.org/

PROJ (2019). Version 4 to 6 API Migration. Retrieved from https://proj.org/development/migration.html

QGIS (2019). QGIS Homepage. Retrieved from https://qgis.org/en/site/about/index.html

QGIS (2019). Changelog for QGIS 3.8. Retrieved from https://qgis.org/en/site/forusers/visualchangelog38/#feature-much-improved-coordinate-transform-handling

Rouault 2019 - [PROJ] Obtaining possible transformation pipelines. Retrieved from https://lists.osgeo.org/pipermail/proj/2019-May/008604.html

Spatialite (2019) PROJ.6. Retrieved from https://www.gaia-gis.it/fossil/libspatialite/wiki?name=PROJ.6