NAME

projinfo − Geodetic object and coordinate operation queries

SYNOPSIS

projinfo

[−o formats] [−k crs|operation|datum|ensemble|ellipsoid] [−−summary] [−q]
[[−−area name_or_code] | [−−bbox west_long,south_lat,east_long,north_lat]]
[−−spatial−test contains|intersects]
[−−crs−extent−use none|both|intersection|smallest]
[−−grid−check none|discard_missing|sort|known_available]
[−−pivot−crs always|if_no_direct_transformation|never|{auth:code[,auth:code]*}]
[−−show−superseded] [−−hide−ballpark] [−−accuracy {accuracy}]
[−−allow−ellipsoidal−height−as−vertical−crs]
[−−boundcrs−to−wgs84]
[−−authority name]
[−−main−db−path path] [−−aux−db−path path]*
[−−dump−db−structure]
[−−identify] [−−3d]
[−−output−id AUTH:CODE]
[−−c−ify] [−−single−line]
−−searchpaths | −−remote−data |
−−list−crs [list−crs−filter] |
−−dump−db−structure [{object_definition} | {object_reference}] |
{object_definition} | {object_reference} | (−s {srs_def} −t {srs_def})

where {object_definition} or {srs_def} is one of the possibilities accepted by **proj_create()**

- a proj−string,
- a WKT string,
- an object code (like "EPSG:4326", "urn:ogc:def:crs:EPSG::4326", "urn:ogc:def:coordinateOperation:EPSG::1671"),
- an Object name. e.g "WGS 84", "WGS 84 / UTM zone 31N". In that case as uniqueness is not guaranteed, heuristics are applied to determine the appropriate best match.
- a OGC URN combining references for compound coordinate reference systems (e.g "*urn:ogc:def:crs,crs:EPSG::2393,crs:EPSG::5717*" or custom abbreviated syntax "EPSG:2393+5717"),
- a OGC URN combining references for references for projected or derived CRSs e.g. for Projected 3D CRS "UTM zone 31N / WGS 84 (3D)": "*urn:ogc:def:crs,crs:EPSG::4979,cs:PROJ::ENh,coordinateOperation:EPSG::16031*" (*added in 6.2*)
- a OGC URN combining references for concatenated operations (e.g. "*urn:ogc:def:coordinateOperation,coordinateOperation:EPSG::3895,coordinateOperation:EPSG::1618*")
- a PROJJSON string. The jsonschema is at *https://proj.org/schemas/v0.4/projjson.schema.json* (*added in 6.2*)
- a compound CRS made from two object names separated with " + ". e.g. "WGS 84 + EGM96 height" (*added in 7.1*)

{object_reference} is a filename preceded by the '@' character. The file referenced by the {object_reference} must contain a valid {object_definition}.

DESCRIPTION

projinfo is a program that can query information on a geodetic object, coordinate reference system (CRS) or coordinate operation, when the **−s** and **−t** options are specified, and display it under different formats (PROJ string, WKT string or PROJJSON string).

It can also be used to query coordinate operations available between two CRS.

The program is named with some reference to the GDAL **gdalsrsinfo** that offers partly similar services.

The following control parameters can appear in any order:

−o formats

formats is a comma separated combination of: **all**, **default**, **PROJ**, **WKT_ALL**, **WKT2:2015**, **WKT2:2019**, **WKT1:GDAL**, **WKT1:ESRI**, **PROJJSON**, **SQL**.

Except **all** and **default**, other formats can be preceded by − to disable them.

NOTE:
WKT2_2019 was previously called WKT2_2018.

NOTE:
Before PROJ 6.3.0, WKT1:GDAL was implicitly calling −−boundcrs−to−wgs84. This is no longer the case.

NOTE:
When SQL is specified, −−*output−id* must be specified.

−k crs|operation|datum|ensemble|ellipsoid

When used to query a single object with a AUTHORITY:CODE, determines the (k)ind of the object in case there are CRS, coordinate operations or ellipsoids with the same CODE. The default is crs.

−−summary

When listing coordinate operations available between 2 CRS, return the result in a summary format, mentioning only the name of the coordinate operation, its accuracy and its area of use.

NOTE:
only used for coordinate operation computation

−q Turn on quiet mode. Quiet mode is only available for queries on single objects, and only one output format is selected. In that mode, only the PROJ, WKT or PROJJSON string is displayed, without other introduction output. The output is then potentially compatible of being piped in other utilities.

−−area name_or_code

Specify an area of interest to restrict the results when researching coordinate operations between 2 CRS. The area of interest can be specified either as a name (e.g "Denmark − onshore") or a AUTHORITY:CODE (EPSG:3237) This option is exclusive of −−*bbox*.

NOTE:
only used for coordinate operation computation

−−bbox west_long,south_lat,east_long,north_lat

Specify an area of interest to restrict the results when researching coordinate operations between 2 CRS. The area of interest is specified as a bounding box with geographic coordinates, expressed in degrees in a unspecified geographic CRS. *west_long* and *east_long* should be in the [−180,180] range, and *south_lat* and *north_lat* in the [−90,90]. *west_long* is generally lower than *east_long*, except in the case where the area of interest crosses the antimeridian.

NOTE:
only used for coordinate operation computation

−−spatial−test contains|intersects
Specify how the area of use of coordinate operations found in the database are compared to the area of use specified explicitly with *−−area* or *−−bbox*, or derived implicitly from the area of use of the source and target CRS. By default, **projinfo** will only keep coordinate operations whose are of use is strictly within the area of interest (**contains** strategy). If using the**intersects** strate gy, the spatial test is relaxed, and any coordinate operation whose area of use at least partly intersects the area of interest is listed.

NOTE:
only used for coordinate operation computation

−−crs−extent−use none|both|intersection|smallest
Specify which area of interest to consider when no explicit one is specified with *−−area* or *−−bbox* options. By default (**smallest** strategy), the area of use of the source or target CRS will be looked, and the one that is the smallest one in terms of area will be used as the area of interest. If using **none**, no area of interest is used. If using **both**, only coordinate operations that relate (contain or intersect depending of the *−−spatial−test* strategy) to the area of use of both CRS are selected. If using**intersection**, the area of interest is the intersection of the bounding box of the area of use of the source and target CRS

NOTE:
only used for coordinate operation computation

−−grid−check none|discard_missing|sort|known_available
Specify how the presence or absence of a horizontal or vertical shift grid required for a coordinate operation affects the results returned when researching coordinate operations between 2 CRS. The default strategy is **sort** (if **PROJ_NETWORK** is not defined). In that case, all candidate operations are returned, but the actual availability of the grids is used to determine the sorting order. That is, if a coordinate operation involves using a grid that is not available in the PROJ resource directories (determined by the **PROJ_LIB** environment variable, it will be listed in the bottom of the results. The **none** strategy completely disables the checks of presence of grids and this returns the results as if all the grids where available. The**discard_missing** strate gy discards results that involve grids not present in the PROJ resource directories. The **known_available** strategy discards results that involve grids not present in the PROJ resource directories and that are not known of the CDN. This is the default strategy is **PROJ_NETWORK** is set to **ON**.

NOTE:
only used for coordinate operation computation

−−pivot−crs always|if_no_direct_transformation|never|{auth:code[,auth:code]*}
Determine if intermediate (pivot) CRS can be used when researching coordinate operation between 2 CRS. A typical example is the WGS84 pivot. By default, **projinfo** will consider any potential pivot if there is no direct transformation (**if_no_direct_transformation**). If using the **never** strategy, only direct transformations between the source and target CRS will be used. If using the **always** strategy, intermediate CRS will be considered even if there are direct transformations. It is also possible to restrict the pivot CRS to consider by specifying one or several CRS by their AUTHORITY:CODE.

NOTE:
only used for coordinate operation computation

−−show−superseded
When enabled, coordinate operations that are superseded by others will be listed. Note that supersession is not equivalent to deprecation: superseded operations are still considered valid although

they have a better equivalent, whereas deprecated operations have been determined to be erroneous and are not considered at all.

NOTE:
only used for coordinate operation computation

−−hide−ballpark
New in version 7.1.

Hides any coordinate operation that is, or contains, a Ballpark transformation

NOTE:
only used for coordinate operation computation

−−accuracy {accuracy}
New in version 8.0.

Sets the minimum desired accuracy for returned coordinate operations.

NOTE:
only used for coordinate operation computation

−−allow−ellipsoidal−height−as−vertical−crs
New in version 8.0.

Allows exporting a geographic or projected 3D CRS as a compound CRS whose vertical CRS represents the ellipsoidal height.

NOTE:
only used for CRS, and with WKT1:GDAL output format

−−boundcrs−to−wgs84
When specified, this option researches a coordinate operation from the base geographic CRS of the single CRS, source or target CRS to the WGS84 geographic CRS, and if found, wraps those CRS into a BoundCRS object. This is mostly to be used for early−binding approaches.

−−authority name
Specify the name of the authority into which to restrict looks up for objects, when specifying an object by name or when coordinate operations are computed. The default is to allow all authorities.

When used with SQL output, this restricts the authorities to which intermediate objects can belong to (the default is EPSG and PROJ). Note that the authority of the *−−output−id* option will also be implicitly added.

−−main−db−path path
Specify the name and path of the database to be used by **projinfo**. The default is **proj.db** in the PROJ resource directories.

−−aux−db−path path
Specify the name and path of auxiliary databases, that are to be combined with the main database. Those auxiliary databases must have a table structure that is identical to the main database, but can be partly filled and their entries can refer to entries of the main database. The option may be repeated to specify several auxiliary databases.

−−identify
When used with an object definition, this queries the PROJ database to find known objects, typically CRS, that are close or identical to the object. Each candidate object is associated with an approximate likelihood percentage. This is useful when used with a WKT string that lacks a EPSG identifier, such as ESRI WKT1. This might also be used with PROJ strings. For example, *+proj=utm +zone=31 +datum=WGS84 +type=crs* will be identified with a likelihood of 70% to EPSG:32631

−−dump−db−structure
New in version 8.1.

Outputs the sequence of SQL statements to create a new empty valid auxiliary database. This option can be specified as the only switch of the utility. If also specifying a CRS object and the *−−output−id* option, the definition of the object as SQL statements will be appended.

−−list−crs [list−crs−filter]
New in version 8.1.

Outputs a list (authority name:code and CRS name) of the filtered CRSs from the database. If no filter is provided all authority names and types of non deprecated CRSs are dumped. list−crs−filter is a comma separated combination of: allow_deprecated,geodetic,geocentric, geographic,geographic_2d,geographic_3d,vertical,projected,compound. Affected by options *−−authority*, *−−area*, *−−bbox* and *−−spatial−test*

−−3d New in version 6.3.

"Promote" the CRS(s) to their 3D version. In the context of researching available coordinate transformations, explicitly specifying this option is not necessary, because when one of the source or target CRS has a vertical component but not the other one, the one that has no vertical component is automatically promoted to a 3D version, where its vertical axis is the ellipsoidal height in metres, using the ellipsoid of the base geodetic CRS.

−−output−id=AUTH:NAME
New in version 8.1.

Identifier to assign to the object (for SQL output).

It is strongly recommended that new objects should not be added in common registries, such as **EPSG**, **ESRI**, **IAU**, etc. Users should use a custom authority name instead. If a new object should be added to the official EPSG registry, users are invited to follow the procedure explained at *https://epsg.org/dataset−change−requests.html*.

Combined with *−−dump−db−structure*, users can create auxiliary databases, instead of directly modifying the main **proj.db** database. See the *example how to export to an auxiliary database*.

Those auxiliary databases can be specified through **proj_context_set_database_path()** or the **PROJ_AUX_DB** environment variable.

−−c−ify
For developers only. Modify the string output of the utility so that it is easy to put those strings in C/C++ code

−−single−line
Output PROJ, WKT or PROJJSON strings on a single line, instead of multiple indented lines by default.

−−searchpaths
New in version 7.0.

Output the directories into which PROJ resources will be looked for (if not using C API such as **proj_context_set_search_paths()** that will override them.

−−remote−data
New in version 7.0.

Display information regarding if network is enabled, and the related URL.

EXAMPLES

1. Query the CRS object corresponding to EPSG:4326

```
projinfo EPSG:4326
```

Output:

```
PROJ.4 string:
+proj=longlat +datum=WGS84 +no_defs +type=crs

WKT2:2019 string:
GEOGCRS["WGS 84",
    DATUM["World Geodetic System 1984",
        ELLIPSOID["WGS 84",6378137,298.257223563,
            LENGTHUNIT["metre",1]]],
    PRIMEM["Greenwich",0,
        ANGLEUNIT["degree",0.0174532925199433]],
    CS[ellipsoidal,2],
        AXIS["geodetic latitude (Lat)",north,
            ORDER[1],
            ANGLEUNIT["degree",0.0174532925199433]],
        AXIS["geodetic longitude (Lon)",east,
            ORDER[2],
            ANGLEUNIT["degree",0.0174532925199433]],
    USAGE[
        SCOPE["unknown"],
        AREA["World"],
        BBOX[−90,−180,90,180]],
    ID["EPSG",4326]]
```

2. List the coordinate operations between NAD27 (designed with its CRS name) and NAD83 (designed with its EPSG code 4269) within an area of interest

```
projinfo −s NAD27 −t EPSG:4269 −−area "USA − Missouri"
```

Output:

```
DERIVED_FROM(EPSG):1241, NAD27 to NAD83 (1), 0.15 m, USA − CONUS including EEZ

PROJ string:
```

```
+proj=pipeline +step +proj=axisswap +order=2,1 +step +proj=unitconvert \
+xy_in=deg +xy_out=rad +step +proj=hgridshift +grids=conus \
+step +proj=unitconvert +xy_in=rad +xy_out=deg +step +proj=axisswap +order=2,1

WKT2:2019 string:
COORDINATEOPERATION["NAD27 to NAD83 (1)",
    SOURCECRS[
        GEOGCRS["NAD27",
            DATUM["North American Datum 1927",
                ELLIPSOID["Clarke 1866",6378206.4,294.978698213898,
                    LENGTHUNIT["metre",1]]],
            PRIMEM["Greenwich",0,
                ANGLEUNIT["degree",0.0174532925199433]],
            CS[ellipsoidal,2],
                AXIS["geodetic latitude (Lat)",north,
                    ORDER[1],
                    ANGLEUNIT["degree",0.0174532925199433]],
                AXIS["geodetic longitude (Lon)",east,
                    ORDER[2],
                    ANGLEUNIT["degree",0.0174532925199433]]]],
    TARGETCRS[
        GEOGCRS["NAD83",
            DATUM["North American Datum 1983",
                ELLIPSOID["GRS 1980",6378137,298.257222101,
                    LENGTHUNIT["metre",1]]],
            PRIMEM["Greenwich",0,
                ANGLEUNIT["degree",0.0174532925199433]],
            CS[ellipsoidal,2],
                AXIS["geodetic latitude (Lat)",north,
                    ORDER[1],
                    ANGLEUNIT["degree",0.0174532925199433]],
                AXIS["geodetic longitude (Lon)",east,
                    ORDER[2],
                    ANGLEUNIT["degree",0.0174532925199433]]]],
    METHOD["CTABLE2"],
    PARAMETERFILE["Latitude and longitude difference file","conus"],
    OPERATIONACCURACY[0.15],
    USAGE[
        SCOPE["unknown"],
        AREA["USA - CONUS including EEZ"],
        BBOX[23.81,-129.17,49.38,-65.69]],
    ID["DERIVED_FROM(EPSG)",1241]]
```

3. Export an object as a PROJJSON string

```
projinfo GDA94 -o PROJJSON -q
```

Output:

```
{
    "type": "GeographicCRS",
    "name": "GDA94",
    "datum": {
        "type": "GeodeticReferenceFrame",
        "name": "Geocentric Datum of Australia 1994",
```

```
            "ellipsoid": {
                "name": "GRS 1980",
                "semi_major_axis": 6378137,
                "inverse_flattening": 298.257222101
            }
        },
        "coordinate_system": {
            "subtype": "ellipsoidal",
            "axis": [
            {
                "name": "Geodetic latitude",
                "abbreviation": "Lat",
                "direction": "north",
                "unit": "degree"
            },
            {
                "name": "Geodetic longitude",
                "abbreviation": "Lon",
                "direction": "east",
                "unit": "degree"
            }
            ]
        },
        "area": "Australia − GDA",
        "bbox": {
            "south_latitude": −60.56,
            "west_longitude": 93.41,
            "north_latitude": −8.47,
            "east_longitude": 173.35
        },
        "id": {
            "authority": "EPSG",
            "code": 4283
        }
    }
```


4. Exporting the SQL statements to insert a new CRS in an auxiliary database.

```
# Get the SQL statements for a custom CRS
projinfo "+proj=merc +lat_ts=5 +datum=WGS84 +type=crs +title=my_crs" −−output−
cat my_crs.sql

# Initialize an auxiliary database with the schema of the reference database
echo ".schema" | sqlite3 /path/to/proj.db | sqlite3 aux.db

# Append the content of the definition of HOBU:MY_CRS
sqlite3 aux.db < my_crs.db

# Check that everything works OK
projinfo −−aux−db−path aux.db HOBU:MY_CRS
```

or more simply:

```
# Create an auxiliary database with the definition of a custom CRS.
projinfo "+proj=merc +lat_ts=5 +datum=WGS84 +type=crs +title=my_crs" −−output−
```

```
# Check that everything works OK
projinfo --aux-db-path aux.db HOBU:MY_CRS
```

Output:

```
INSERT INTO geodetic_crs VALUES('HOBU','GEODETIC_CRS_MY_CRS','unknown','','geog
INSERT INTO usage VALUES('HOBU','USAGE_GEODETIC_CRS_MY_CRS','geodetic_crs','HOB
INSERT INTO conversion VALUES('HOBU','CONVERSION_MY_CRS','unknown','','EPSG','9
INSERT INTO usage VALUES('HOBU','USAGE_CONVERSION_MY_CRS','conversion','HOBU',
INSERT INTO projected_crs VALUES('HOBU','MY_CRS','my_crs','','EPSG','4400','HOB
INSERT INTO usage VALUES('HOBU','USAGE_PROJECTED_CRS_MY_CRS','projected_crs','M

PROJ.4 string:
+proj=merc +lat_ts=5 +lon_0=0 +x_0=0 +y_0=0 +datum=WGS84 +units=m +no_defs +type

WKT2:2019 string:
PROJCRS["my_crs",
    BASEGEOGCRS["unknown",
        ENSEMBLE["World Geodetic System 1984 ensemble",
            MEMBER["World Geodetic System 1984 (Transit)"],
            MEMBER["World Geodetic System 1984 (G730)"],
            MEMBER["World Geodetic System 1984 (G873)"],
            MEMBER["World Geodetic System 1984 (G1150)"],
            MEMBER["World Geodetic System 1984 (G1674)"],
            MEMBER["World Geodetic System 1984 (G1762)"],
            ELLIPSOID["WGS 84",6378137,298.257223563,
                LENGTHUNIT["metre",1]],
            ENSEMBLEACCURACY[2.0]],
        PRIMEM["Greenwich",0,
            ANGLEUNIT["degree",0.0174532925199433]],
        ID["HOBU","GEODETIC_CRS_MY_CRS"]],
    CONVERSION["unknown",
        METHOD["Mercator (variant B)",
            ID["EPSG",9805]],
        PARAMETER["Latitude of 1st standard parallel",5,
            ANGLEUNIT["degree",0.0174532925199433],
            ID["EPSG",8823]],
        PARAMETER["Longitude of natural origin",0,
            ANGLEUNIT["degree",0.0174532925199433],
            ID["EPSG",8802]],
        PARAMETER["False easting",0,
            LENGTHUNIT["metre",1],
            ID["EPSG",8806]],
        PARAMETER["False northing",0,
            LENGTHUNIT["metre",1],
            ID["EPSG",8807]]],
    CS[Cartesian,2],
        AXIS["(E)",east,
            ORDER[1],
            LENGTHUNIT["metre",1]],
        AXIS["(N)",north,
            ORDER[2],
            LENGTHUNIT["metre",1]],
    ID["HOBU","MY_CRS"]]
```

5. Get the WKT representation of EPSG:25832 in the WKT1:GDAL output format and on a single line

```
projinfo -o WKT1:GDAL --single-line EPSG:25832
```

Output:

```
WKT1:GDAL string:
PROJCS["ETRS89 / UTM zone 32N",GEOGCS["ETRS89",DATUM["European_Terrestrial_Ref
```

SEE ALSO

cs2cs(1), **cct(1)**, **geod(1)**, **gie(1)**, **proj(1)**, **projsync(1)**

BUGS

A list of known bugs can be found at *https://github.com/OSGeo/PROJ/issues* where new bug reports can be submitted to.

HOME PAGE

https://proj.org/

AUTHOR

Even Rouault

COPYRIGHT