

---

# **GB/T 2260-2007 Documentation**

***Release 0.2-dev***

**Paul Natsuo Kishimoto**

**Apr 03, 2018**



---

## Contents

---

|          |                                 |          |
|----------|---------------------------------|----------|
| <b>1</b> | <b>Documentation</b>            | <b>3</b> |
| 1.1      | Data model . . . . .            | 3        |
| 1.2      | API reference . . . . .         | 3        |
| 1.3      | Updating the database . . . . . | 5        |
| 1.4      | Copyright & license . . . . .   | 5        |
|          | <b>Python Module Index</b>      | <b>7</b> |



The GB/T 2260 standard defines six-digit numerical codes for the administrative divisions of China, at the county level and above. For instance, the Haidian district of Beijing has the code **110108**.

The most recent version of the official standard, designated “GB/T 2260-2007”, was published in 2008. However, codes are routinely revised, and the National Bureau of Statistics (NBS) [publishes an updated list online annually](#).

gb2260 exposes an up-to-date database of the GB/T 2260 codes, with extra information including English names, Pinyin transcriptions, administrative levels, etc., as well as code for updating the database with newly-released changes.



## 1.1 Data model

The database includes the following fields, and is *sparse*: not every field is populated for every entry. In particular, only `code`, `name_zh` and `level` exist for all entries.

**code** The six-digit GB/T 2260-2007 code (`int`).

**name\_zh** Name of the region in simplified Chinese (`str`).

**level** Administrative level of the region (`int`, one of 1, 2, or 3).

See [this table](#) for an explanation of the various names for these levels.

**name\_pinyin** Name of the region rendered in pinyin (`str`).

**name\_en** Name of the region in English (`str`).

**alpha** 2- or 3-digit uppercase alphabetical code for the region (`str`).

**latitude, longitude** Latitude and longitude of a point within the region (`float`).

## 1.2 API reference

The entire database is contained in the dictionary `codes`.

```
>>> from gb2260 import *
>>> codes.get(542621) == {
...     'alpha': None,
...     'latitude': 29.6365717,
...     'level': 3,
...     'longitude': 94.3610895,
...     'name_en': 'Nyingchi',
...     'name_pinyin': 'Linzhi',
...     'name_zh': '',
... }
```

(continues on next page)

(continued from previous page)

```
... }
True
>>> codes.get(632726) ['level']
3
```

**exception** `gb2260.AmbiguousRegionError`  
Exception for a lookup that returns multiple results.

**exception** `gb2260.InvalidCodeError`  
Exception for an invalid code.

**exception** `gb2260.RegionKeyError`  
Exception for a lookup that returns nothing.

`gb2260.isolike` (*code*, *prefix*='CN-')  
Return an 'ISO 3166-2-like' alpha code for *code*.

ISO 3166-2:CN defines codes like “CN-11” for Beijing, where “11” are the first two digits of the GB/T 2260 code, 110000. An 'ISO 3166-2-like' alpha code uses the official GB/T 2260 two-letter alpha codes for province-level divisions (e.g. 'BJ'), and three-letter alpha codes for lower divisions, separated by hyphens:

```
>>> alpha(130100)
'CN-HE-SJW'
```

For divisions below level 2, no official alpha codes are provided, so `alpha()` raises `ValueError`.

`gb2260.level` (*code*)  
Return the administrative level of *code*.

```
>>> level(110108)
3
```

For codes not in the database, raises `InvalidCodeError`.

`gb2260.parent` (*code*, *parent\_level*=None)  
Return a valid code that is the parent of *code*.

```
>>> parent(110108)
110100
>>> parent(110100)
110000
```

If *parent\_level* is supplied, the parent at the desired level is returned:

```
>>> parent(110108, 1)
110000
```

`gb2260.split` (*code*)  
Return a tuple containing the three parts of *code*.

```
>>> split(331024)
(33, 10, 24)
```

`gb2260.within` (*a*, *b*)  
Return True if division *a* is within (or the same as) division *b*.

```
>>> within(331024, 330000)
True
>>> within(331024, 110000)
False
>>> within(331024, 331024)
True
```

`within()` does *not* check that *a* or *b* are valid codes that exist in the database.

```
>>> within(331024, 990000)
False
>>> within(990101, 990000)
True
```

## 1.3 Updating the database

When invoked as a module:

```
$ python -m gb2260
usage: __main__.py [-h]
                  [--version {2012-10-31,2013-08-31,2014-10-31,2015-09-30}]
                  [--cached] [--verbose]
                  ACTION

positional arguments:
  ACTION                action to perform: either update or refresh-cache

optional arguments:
  -h, --help            show this help message and exit
  --version {2012-10-31,2013-08-31,2014-10-31,2015-09-30}
                        version to update the database with
  --cached              read the data from cached HTML, instead of the NBS
                        website
  --verbose             give verbose output
```

...either of `update()` or `refresh_cache()`, below, can be invoked.

## 1.4 Copyright & license

gb2260 is copyright 2014–2017 [Paul Natsuo Kishimoto](#) and distributed under the [GNU GPLv3](#).

The NBS website, which is scraped as the main and authoritative source of data, provides [this copyright statement](#).

`gbt_2260-2007.csv` and `gbt_2260-2007_sup.csv` contain transcribed information from GB/T 2260-2007, the copyright of which is unknown.

`citas.csv` is under the following “[use constraints](#)” (with emphasis added):

The University of Washington China in Time and Space (CITAS), Chinese Academy of Surveying and Mapping (CASM), and Trustees of Columbia University in the City of New York hold the copyright of this data set. **Users are prohibited from any commercial, non-free resale, or redistribution without explicit written permission** from CITAS, CASM, and CIESIN. Users should acknowledge CITAS, CASM, and CIESIN as the source used in the creation of any reports, publications, new data sets, derived

products, or services resulting from the use of this data set. CITAS, CASM, and CIESIN also request reprints of any publications and notification of any redistributing efforts

- genindex

**g**

gb2260, [3](#)



## A

[AmbiguousRegionError](#), 4

## G

[gb2260](#) (module), 3

## I

[InvalidCodeError](#), 4

[isolike\(\)](#) (in module [gb2260](#)), 4

## L

[level\(\)](#) (in module [gb2260](#)), 4

## P

[parent\(\)](#) (in module [gb2260](#)), 4

## R

[RegionKeyError](#), 4

## S

[split\(\)](#) (in module [gb2260](#)), 4

## W

[within\(\)](#) (in module [gb2260](#)), 4