GeoPostcodes

The most comprehensive
postal database in the world

Additional Datasets
Address Formats

Version 2022.03.25

Table of contents

Introduction

This document focuses on the Address Formats complementing GeoPostcodes's postal and street databases. For more information about the core products, please refer to their product sheets.

Address formats

Each postal operator in the world produces its own requirements for writing addresses in its country. In this additional dataset, GeoPostcodes indicates which field, from its Postal and Street databases, should appear on addresses, and where, when willing to meet the requirements of the local post.

The address format is delivered using a python f-string syntax, where fields are enclosed in curly brackets.

The address formats provided in this dataset cover the fields in GeoPostcodes database. Details about the recipient(s) (typically their name, sometimes the name of their building) must be added above the information presented in the Address Formats dataset.

Data Design

The data is delivered as a csv file (semicolon separator), with 2 fields:

Field name	Field type	Description	Comments
ISO	Char(2)	ISO 3166-1 Country code	The ISO 3166-1 standard is published by the International Organization for Standardization (ISO) and defines a unique code for the name of each country. The country codes are represented as a two-letter code (alpha-2).
gpc_format	Char(200)	Format of the address	See explanations below

The gpc_format field is written following the python f-string syntax, where variables are enclosed within curly brackets.

We assume each record is grouping the following information from the Postal/Street databases, some of which may be empty depending on the countries:

- number: house number
- street: name of the street
- locality: name of the locality
- postcode: the postal code
- suburb: name of the suburb (optional)
- post_town: postal town (if applicable)

- po_box: P.O. Box (if applicable)
- regions 1 to 4, each with their respective name, iso2 code and stat code

Example: {street} {number}\n{postcode} {locality.upper()} {region1.iso2[-2:]}\nSWITZERLAND

- {street} must be replaced by the street name
- {number} must be replaced by the house number
- {postcode} must be replaced by the postal code
- {locality.upper()} must be replaced by the name of the locality, in capital letters
- {region1.iso2[-2:]} must be replaced by the last 2 characters of the iso2 code of the region 1

  Which yields a formatted address similar to:

  MyStreet 32A
  1234 MYTOWN XX
  SWITZERLAND

Sample code

The following python code can be used to print/try the address formats:

```
import csv

class reg:

    def __init__(self,name=None,iso2=None,stat=None):
        self.name=name
        self.iso2=iso2
        self.stat=stat

class record:

    def __init__(self,locality,postcode,street=None,
number=None,region1=None,region2=None,region3=None,region4=None,suburb=None,p
ost_town=None,pobox=''):
        self.locality=locality
        self.postcode=postcode
        self.street=street
        self.number=number
        self.suburb=suburb
        self.post_town=post_town
        self.region1=region1
        self.region2=region2
        self.region3=region3
        self.region4=region4
        self.pobox=pobox

def fstr(template):
    return eval(f"f'{template}'")

with open('GPC-address_formats.csv','r') as csvfile:
    address_formats=csv.reader(csvfile,delimiter=';')
    next(address_formats) # skipping header
    for row in address_formats:
        print('--------- ' + str(row[0]) + ' : ' + row[1])
        print(fstr(str(row[1]).replace('{','{r.')))
```

Sample data

iso	gpc_format
AD	{number} {street.upper()}\n{postcode} {locality.upper()}\n{pobox}\nANDORRA
AE	{pobox}\n{region1.name.upper()}\nUNITED ARAB EMIRATES
AF	{region2.name}, {street}, {locality}\n{number}\n{region1.name.upper()}\n{postcode}\nAFGHANISTAN
AG	{locality}\nANTIGUA AND BARBUDA
AI	{pobox}\n{locality}\n{postcode}\nANGUILLA, BWI
AL	{street}\n{number}\n{postcode}\n{locality.upper()}\nALBANIA
AM	{street} {number}\n{postcode} {locality.upper()}\nARMENIA
AO	{street} {number}\n{pobox}\n{locality.upper()}\nANGOLA
AR	{street.upper()} No {number}\n{postcode} {locality.upper()}\nARGENTINA
AS	{number} {street.upper()}\n{locality.upper()} AS {postcode}\nUNITED STATES OF AMERICA