**Address Validation Rules**

**Introduction**

Only “valid” addresses will be geocoded. This document outlines the business rules that define a “valid” address.

**Rule 1** – the address must have a valid **postcode** OR **state**. [[1]](#footnote-1) [[2]](#footnote-2)

**Rule 2** – the address should have a valid suburb[[3]](#footnote-3). The address must have an acceptable combination of **suburb**, **postcode** and **state** (minimum two out of three). Table 1 defines the business rules for acceptable combinations.

**Assumptions:** Suburbs may exist in multiple states. Suburb may cross state boundaries. Postcodes may cross state boundaries. Where a suburb crosses a state boundary it retains the same postcode in both states. Suburbs may exist in multiple postcodes, but only once within a given postcode (the combination of postcode and suburb is unique[[4]](#footnote-4))

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Rule** | **Have suburb** | **Have postcode** | **Have state** | **suburb/ postcode comb’n defined[[5]](#footnote-5)** | **suburb/ state comb’n defined[[6]](#footnote-6)** | **postcode/ state comb’n defined[[7]](#footnote-7)** | **Only one suburb for this postcode/state combination** | **Minimum geocoding accuracy** | **Comment** |
| V1 | True | True | True | True | True | True | - | suburb | all 3 O.K. |
| V2 | True | True | - | True | - | - | - | suburb[[8]](#footnote-8) | bad state |
| V3 | True | - | True | - | True | - | - | suburb[[9]](#footnote-9) | bad postcode |
| V4 | - | True | True | - | - | True | True | suburb[[10]](#footnote-10) | bad suburb |
| V5 | - | True | True | - | - | True | False | postcode[[11]](#footnote-11) | bad suburb |
| V6 |  |  |  |  |  |  |  | **invalid address** |  |

Table 1 – acceptable combinations of suburb, postcode and state

**Rule 3** – P.O. Box addresses will be geocoded as per **Rule 2**.

**Rule -4** – Compute “validated” suburb, postcode and state. “validated” suburb, postcode and state are the original value, unless they are identified in Rule 2 as being incorrect (See Footnotes 7, 8, 9 and 10 for replacing at most one of suburb, postcode and state with a corrected value, where necessary)

**Rule 5** – if the address does not have a valid street name/street type combination[[12]](#footnote-12) then address will be geocoded as per **Rule 2**.

**Rule 6** – if the address has a valid street name/street type combination[[13]](#footnote-13) then Table 2 rules will apply.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rule** | **Have “validated” suburb** | **street name/ street type unique within suburb** | **street name/ street type unique within state** | **Minimum geocoding accuracy** |
| S1 | True | True | - | street |
| S2 | True | False | True | street[[14]](#footnote-14) |
| S3 |  |  |  | **Apply Rule 2.**  **No further business rules are applicable.** |

Table 2 – street level geocoding accuracy

**Rule 7** – if the house number exist within the valid street name/street type combination, then geocode to the accuracy of the property associated with that house number.[[15]](#footnote-15)

**Fuzz Levels and Accuracy**

The solution will attempt, with ever increasing levels of generosity, to geocode each address. Those increasing levels of generosity are called “Fuzz Level”, because with each increase in Fuzz Level, we get further and further away from the original raw data.

As Fuzz Levels go up, accuracy seems to improve, but this is a two-edged sword. It may be, that at Fuzz Level 7, the address gets geocoded to the accuracy of a specific house, in a specific street. But that street may look very different to the original street in the raw data; it may just “sound like” the original street. Or it may just “look like” the original street if you screw up your face and squint in just the right way. Similarly, the suburb could be quite different to the original suburb in the raw data; it could be a “sounds like” suburb, or a “looks like” suburb, or even a neighboring suburb that “sounds like” or “looks like” the original suburb because the street or a street that “sounds like” or “looks like” the original street just happens to exist a few suburbs across. And, to get some sort of a “match” with all this squinting and squirming, it may have been necessary to pick a street/suburb combination from a different postcode, or a different state, or both. So, a street accuracy match at Fuzz Level 2 may be a better outcome than a specific house number match at Fuzz Level 7. However, in order to make that determination, it is necessary to understand the processing that happens at each Fuzz Level, which varies depending upon whether or not there was a street name in the raw data.

Suburb Fuzz Levels – applies to P.O. Box addresses, but also any address that only contains any two of state, postcode and suburb.

|  |  |
| --- | --- |
| **Suburb Fuzz Level** | **Data for a matching (Rule 2)** |
| 1 | Suburb from raw data |
| 2 | Plus, soundex matching suburbs |
| 3 | Plus, Levenshtein matching suburbs  (where Levenshtein distance is less than |

Fuzz Levels – when looking for street and house number matches

|  |  |
| --- | --- |
| Fuzz Level | Data for matching (Rule 4 and 5) |
| 1 | Suburb from raw data. Street name/street type [street suffix] from raw data. |
| 2 | Plus, soundex matching street names within the postcode and state  (where Levenshtein distance is less than (len(suburb) + 6)/4 |
| 3 | Plus, Levenshtein matching street names within the postcode and state  (where Levenshtein distance is less than (len(street name) + 2)/4 |
| 4 | Plus, streets from soundex matching suburbs within the postcode and state  (where Levenshtein distance is less than (len(suburb + 6)/4 |
| 5 | Plus, streets from Levenshtein matching suburbs within the postcode and state  (where Levenshtein distance is less than (len(suburb) + 2)/4 |
| 6 | Plus, streets from all “adjacent” suburbs [G-NAF does not have a precise network data model] within the state |
| 7 | Plus, soundex matching street names within the state  (where Levenshtein distance is less than (len(suburb + 6)/4,  Plus, Levenshtein matching street names within the state  (where Levenshtein distance is less than (len(street name) + 2)/4 |
| 8 | Plus, streets from soundex matching suburbs within the state  (where Levenshtein distance is less than (len(suburb) + 6)/4,  Plus, streets from Levenshtein matching suburbs within the postcode and state  (where Levenshtein distance is less than (len(suburb) + 2)/4 |
| 9 | Plus, street name from raw data with every other possible street type (no street suffix) within the state |
| 10 | Plus, matching streets from suburbs matching the raw data suburb, but from other states/postcodes |

Accuracy

|  |  |
| --- | --- |
| 1 | Accurate to a single postcode |
| 2 | Accurate to a single suburb |
| 3 | Accurate to a single street |
| 4 | Accurate to a single property |

Confidence – ought to be a simple combination of Fuzz Level and Accuracy. Unfortunately, it’s not that simple. Each address element has its own concept of reliability. “Accurate to a single property” doesn’t mean that the house number was found, in this street, in this suburb. If the house number isn’t found, but a neighboring house number is found, then the solution will geocode to the accuracy of that neighboring property. And Fuzz Level doesn’t help with this analysis. At Fuzz Level 5 the geocoding solution may have found “Lakeview Ave” as a match for “Lakeview Rd”, but did the geocoding solution find this street in the raw data suburb, or a soundex matched suburb, or a Levenshtein matched suburb? That information can be returned by the solution, but will the client be able to interpret it?

Soundex – is a well know, if not well understood, algorithm for converting long names into short codes; a large number of names get grouped into a small number of sets of names, where all of the names in each set share the same Soundex code. The algorithm is relatively simple; keep the first letter, discards vowel, reduces doubled up letters to a single letter, them place the remaining consonants with a number between 1 and 6 where each number represent a group of consonant that sort of sound similar (i.e. m and n, d and t). Soundex is always a fixed length code; if you have more than three consonants, then discard the extras, if you have less than three then add the number ‘0’ until you have a four character code.

So, “Russell” becomes “Rsl” which becomes “R240”. And “Russel” also become “R240”. As does “Rouselle”.

Levenshtein Distance is a measure of the similarity of two words. You can think of it as “How many typing errors did I make”; I wanted to type “Russell”, but I typed “Rustle”. Soundex says that they are not similar, because they have different codes; R240 and R234. Levenshtein distance also suggests that these are not similar, because there are four typing errors; the ‘t’ should be and ‘s’, then we need to add an ‘e’, and then add an ‘l’, then delete the ‘e’ at the end. However, “Russell” and “Russel” are similar because there is only one typing mistake. One refinement allows more typing mistakes in longer names. This variant is a little more tolerant when “Corhanwarrabul Close” is misspelt, but less tolerant when “First Street” is mistyped as “Fist Street”.

1. This is a conservative approach. Some street/suburb combinations are unique within Australia, but without a postcode or state for reference it is impossible to guarantee that the street/suburb combination is not a reference to an address in another country. [↑](#footnote-ref-1)
2. A valid state is one defined in the G-NAF reference data. A valid postcode is one defined in the reference file of Australian Postcodes and Suburbs. [↑](#footnote-ref-2)
3. A valid suburb is a location that is defined in the G-NAF reference data or a suburb that is defined in the reference file of Australian Postcodes and Suburbs. [↑](#footnote-ref-3)
4. There are duplicate definitions for some postcode/suburb combinations in the Australian Postcodes and Suburbs reference file, but the latitude and longitudes are either identical, or geocode to the same SA1/LGA. Only one of these duplicates will be used. [↑](#footnote-ref-4)
5. This suburb/postcode combination must be defined in the reference file of Australian Postcodes and Suburbs. [↑](#footnote-ref-5)
6. Suburb must match a G-NAF Location name and this combination of G-NAF Location name and state must be defined in the G-NAF reference data. [↑](#footnote-ref-6)
7. This postcode/state combination must be defined in the reference file of Australian Postcodes and Suburbs. [↑](#footnote-ref-7)
8. Incorrect or missing state – suburb/postcode combinations are unique (see Assumptions). If state is being returned as part of the meta data and the postcode does not cross a state boarder, then return the state matching this postcode. [↑](#footnote-ref-8)
9. Incorrect or missing postcode – suburb/state combinations are unique (see Assumptions), but the suburb may cross a postcode boundary. If postcode is being returned as part of the meta data and

   The reference file of Australian Postcodes and Suburbs does not contain a postcode for this suburb/state combination, then return a ‘blank’ postcode.

   The reference file of Australian Postcodes and Suburbs contains one or more postcodes for this suburb/state combination, then return a postcode from the reference file of Australian Postcodes and Suburbs, that matches this suburb/state combination. **Note:** this is an approximate postcode; a “best guess” postcode. [↑](#footnote-ref-9)
10. Incorrect or missing suburb – if suburb is being returned as part of the meta data then return the suburb that matches this unique postcode/state combination. This replacement suburb becomes the “validated” suburb. [↑](#footnote-ref-10)
11. Incorrect or missing suburb – if suburb is being returned as part of the meta data then return an empty string as the suburb; ‘accuracy’ will identify the suburb in the returned meta data as invalid. In this situation we do not have a “validated” suburb. [↑](#footnote-ref-11)
12. A valid street name/street type combination is one defined in the G-NAF reference data. [↑](#footnote-ref-12)
13. ibid. [↑](#footnote-ref-13)
14. The suburb, and possibly postcode, are incorrect. The G-NAF reference data has a suburb for this street. The reference file of Australian Postcodes and Suburbs should have a postcode for this suburb/”validate” state combination. If suburb is being returned as part of the meta data, then return the matching suburb from the G-NAF reference data. If postcode is being returned as part of the meta data and

    The reference file of Australian Postcodes and Suburbs does not contain a postcode for this suburb/”validated” state combination, then return a ‘blank’ postcode.

    The reference file of Australian Postcodes and Suburbs contains one or more postcodes for this suburb/”validated” state combination, then return a postcode from the reference file of Australian Postcodes and Suburbs, that matches this suburb/”validated” state combination. **Note:** this is an approximate postcode; a “best guess” postcode. [↑](#footnote-ref-14)
15. Geocoding will be accurate to the property (land area) not the individual building/structure. Flats/Units/Apartments with the same street no. will all return the same geocoding meta data. [↑](#footnote-ref-15)