

ORDNANCE SURVEY GB

CODE-POINT WITH POLYGONS GETTING STARTED GUIDE

Version history

| Version | Date | Description |
|---------|---------|-----------------------------------------------------------------------------------------------|
| 1.0 | 10/2021 | Initial release of this Getting Started Guide. Addition of vector tiles format to the product |
| 1.1 | 06/2022 | Addition of GeoPackage format to the product |

Purpose of this document

This document provides information about and insight into the Code-Point with Polygons product and its potential applications. For information on the contents and structure of Code-Point with Polygons, please refer to the Overview and Technical Specification.

The terms and conditions on which Code-Point with Polygons is made available to you and your organisation are contained in that Ordnance Survey customer contract. Please ensure your organisation has signed a valid current customer contract to be able to use Code-Point with Polygons.

We may change the information in this document at any time, giving you the notice period set out in your contract. We do not accept responsibility for the content of any third-party websites referenced or accessed in or through this document.

This document has been screened according to Ordnance Survey's Equality Scheme. If you have difficulty reading this information in its current format and would like to find out how to access it in a different format (braille, large print, computer disk or in another language), please contact us on: +44 (0)3456 05 05 05.

Copyright in this document

© Ordnance Survey Limited 2022. This document (including for the avoidance of doubt, any mapping images reproduced within it) is protected by copyright and apart from the rights expressly granted within this document to use the content, all rights are reserved. Any part of this document may be copied for use internally in your organisation or business so that you can use Code-Point with Polygons under the terms of your licence (but not otherwise).

No part of this document may be reproduced or transmitted in any form or by any means (including electronically) for commercial exploitation, onward sale or as free promotional material without getting the written consent of Ordnance Survey beforehand.

This product contains National Statistics data © Crown copyright and database right 2022.

Trade marks

Ordnance Survey, OS, the OS Logos and Code-Point are registered trade marks, and Code-Point with Polygons is a trade mark of Ordnance Survey, Britain's mapping agency.

PAF, PO Box and Royal Mail are registered trade marks of Royal Mail Group Ltd.

Contact details

[OS website 'Contact us' page \(https://www.ordnancesurvey.co.uk/contact-us\)](https://www.ordnancesurvey.co.uk/contact-us).

Contents

| | |
|----------------------------------------------------|-----------|
| 1. Introduction..... | 4 |
| 2. Using the vertical street lookup | 5 |
| 2.1 MapInfo | 5 |
| 2.2 ESRI..... | 7 |
| 2.3 PostGIS and QGIS..... | 9 |
| 2.3.1 PostGIS steps..... | 9 |
| 2.3.2 QGIS steps..... | 10 |
| 3. Creating single-space postcodes..... | 12 |
| 3.1 Centroids and polygons..... | 12 |
| 3.1.1 Outward and inward bound representation..... | 12 |
| 4. Data measures | 21 |

I. Introduction

Code-Point with Polygons is a dataset that contains the notional area of postcode units, allowing customers to display and analyse any data collected at the postcode level.

The polygons within the product are derived from georeferenced Royal Mail Postal Address File (PAF) delivery addresses. A process is undertaken to create a set of polygons around individual address records within a postcode. This is called a Thiessen process and the polygons are the result of a mathematical computation that creates polygons from point data. In this way, mathematically consistent boundaries are created between distinct postcode groups, creating this notional boundary set.



Figure 1. Polygons containing postcodes as supplied in the Code-Point with Polygons product

Postcode unit boundaries are, by definition, only the delivery point or collection of delivery points that constitutes the postcode units. The boundary is therefore a notional one, the position of which is arbitrary. What has been created, however, is a set of boundaries that follows a consistent logic and portrays the notional footprint of each postcode unit. The boundary encloses every delivery address for which positional data of sufficient quality is available, and which follow major physical features that could reasonably be regarded as part of the postcode boundary.

This getting started guide focuses on using the product in Shapefile, TAB and MID/MIF formats. For guidance on using the product in vector tiles or GeoPackage formats please see the following getting started guides, which are available on the OS website:

- [Getting Started with Vector Tiles guide \(https://www.ordnancesurvey.co.uk/documents/user-guides/getting-started-with-vector-tiles-v1.0.pdf\)](https://www.ordnancesurvey.co.uk/documents/user-guides/getting-started-with-vector-tiles-v1.0.pdf)
- [Getting Started with GeoPackage guide \(https://www.ordnancesurvey.co.uk/documents/getting-started-with-geopackage.pdf\)](https://www.ordnancesurvey.co.uk/documents/getting-started-with-geopackage.pdf)

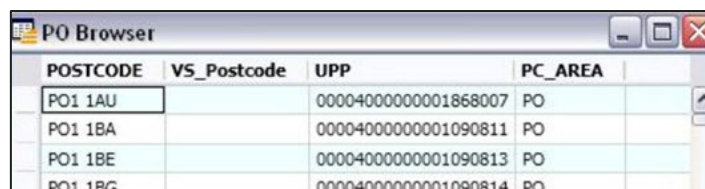
2. Using the vertical street lookup

2.1 MapInfo

MapInfo software enables a separate browser to be viewed that contains all the postcodes related to a vertical street.

To view the vertical street lookup in MapInfo, open the file in a browser.

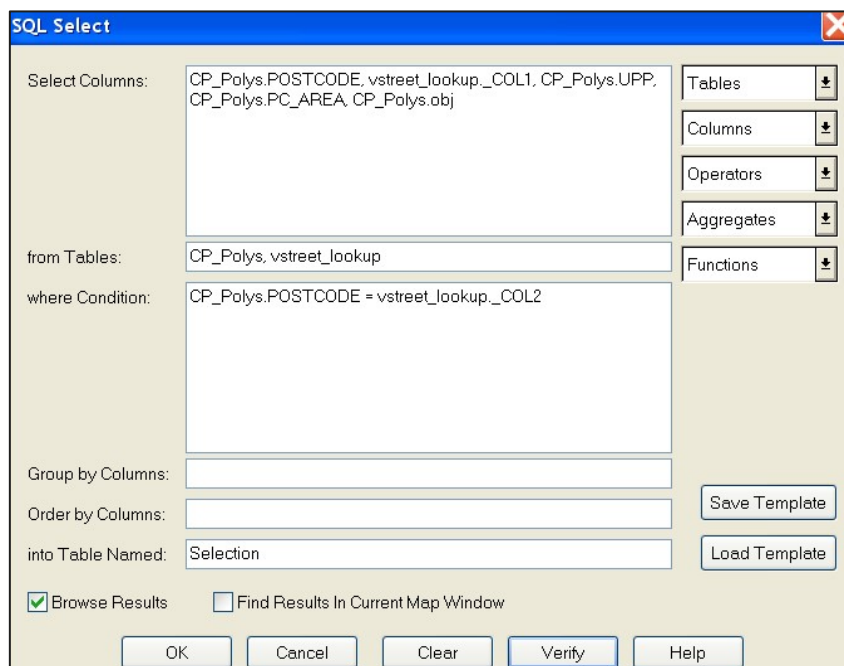
To join the Code_Point Polygons table and the vstreet_lookup table follow these instructions:



| POSTCODE | VS_Postcode | UPP | PC_AREA |
|----------|-------------|----------------------|---------|
| PO1 1AU | | 00004000000001868007 | PO |
| PO1 1BA | | 00004000000001090811 | PO |
| PO1 1BE | | 00004000000001090813 | PO |
| PO1 1BG | | 00004000000001090814 | PO |

1. Save copies of the main Code-Point Polygons and Vertical Street Lookup tables to work on; for example, CP_Polys.TAB and vstreet_lookup.TAB. Open the new copies of the datasets in MapInfo.
2. Create a new column, of Character (8) type, in the Code-Point Polygons table (CP_Polys.TAB) as column 2. Give it a meaningful name; for example.
3. VS_Postcode (go to *Table > Maintenance > Table Structure*).
4. Save the table.

Create geometry for the Vertical Street Lookup table (vstreet_lookup.TAB) by joining it to the main Code-Point Polygons table using the following SQL Select query:



SQL Select

Select Columns: CP_Polys.POSTCODE, vstreet_lookup._COL1, CP_Polys.UPP, CP_Polys.PC_AREA, CP_Polys.obj

from Tables: CP_Polys, vstreet_lookup

where Condition: CP_Polys.POSTCODE = vstreet_lookup._COL2

Group by Columns:

Order by Columns:

into Table Named: Selection

Browse Results Find Results In Current Map Window

OK Cancel Clear Verify Help

Save Template Load Template

Tables Columns Operators Aggregates Functions

1. Save a copy of the results of the selection as PC_VS_join.TAB.
2. Open the table.

| POSTCODE | _COL1 | UPP | PC_AREA | _Object |
|----------|---------|----------------------|---------|---------|
| VLS00005 | LS249RF | 00004000000001780087 | LS | Region |
| VLS00005 | LS249RQ | 00004000000001780087 | LS | Region |
| VLS00006 | LS290HX | 00004000000001780088 | LS | Region |
| VLS00006 | LS290PL | 00004000000001780088 | LS | Region |
| VLS00007 | LS249RF | 00004000000001780089 | LS | Region |
| VLS00007 | LS249RQ | 00004000000001780089 | LS | Region |
| VLS00008 | LS290BF | 00004000000001780090 | LS | Region |
| VLS00008 | LS290JE | 00004000000001780090 | LS | Region |
| VLS00009 | LS249RF | 00004000000001780091 | LS | Region |
| VLS00009 | LS249RQ | 00004000000001780091 | LS | Region |

3. Delete the vertical streets from the main Code-Point Polygons table, CP_Polys.TAB – use Query > Select... to search for postcode 'Like "v%"' and delete:

4. Save, then pack the Code-Point Polygons table.
5. Go into *Table > Maintenance > Pack Table* and select *Pack Both Types of Data*.
6. Append the Postcode – Vertical Street join table, PC_VS_join, to the main Code-Point Polygons table, CP_Polys, to create a list of all postcodes including vertical streets (*Table > Append Rows to Table*).

7. Save the CP_Polys table.

If you browse the CP_Polys table now, the rows with a 'standard' postcode will have a blank in the second field (VS_Postcode), but the rows that have a vertical street postcode in the Postcode (first) column should have a 'standard' postcode in the second field.

| POSTCODE | VS_Postcode | UPP | PC_AREA |
|----------|-------------|----------------------|---------|
| LS9 9NQ | | 00004000000000798881 | LS |
| LS9 9NR | | 00004000000000798882 | LS |
| LS9 9NS | | 00004000000000798883 | LS |
| LS9 9NT | | 00004000000000798884 | LS |
| LS9 9NU | | 00004000000000798885 | LS |
| LS9 9NW | | 00004000000000798886 | LS |
| LS9 9NX | | 00004000000000798887 | LS |
| VLS00005 | LS249RF | 00004000000001780087 | LS |
| VLS00005 | LS249RQ | 00004000000001780087 | LS |
| VLS00006 | LS290HX | 00004000000001780088 | LS |
| VLS00006 | LS290PL | 00004000000001780088 | LS |
| VLS00007 | LS249RF | 00004000000001780089 | LS |
| VLS00007 | LS249RQ | 00004000000001780089 | LS |
| VLS00008 | LS290BF | 00004000000001780090 | LS |
| VLS00008 | LS290JE | 00004000000001780090 | LS |
| VLS00009 | LS249RF | 00004000000001780091 | LS |
| VLS00009 | LS249RQ | 00004000000001780091 | LS |
| VLS00011 | LS249RF | 00004000000001780093 | LS |

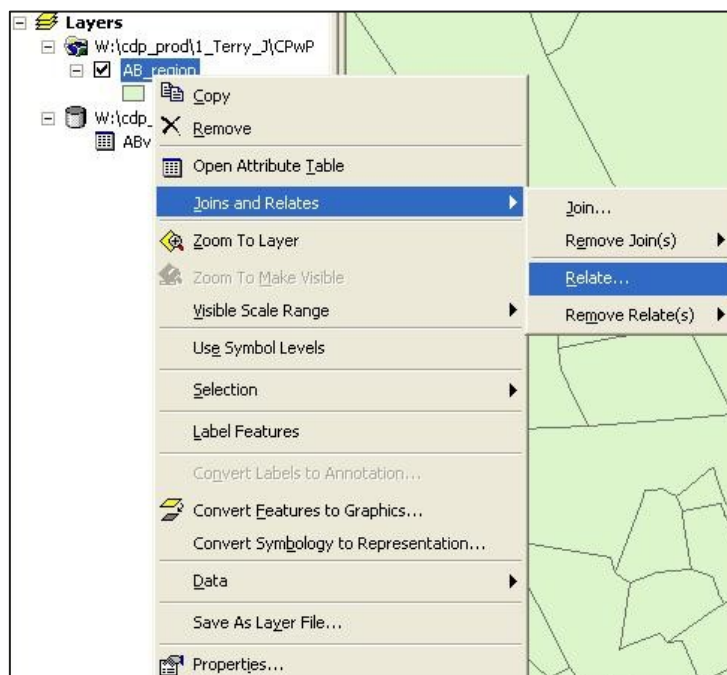
2.2 ESRI

ESRI software enables a relationship to be created between two tables. By relating the postcode area and vertical street lookup, it is possible to identify the postcodes related to a vertical street by using the *Identify* tool.

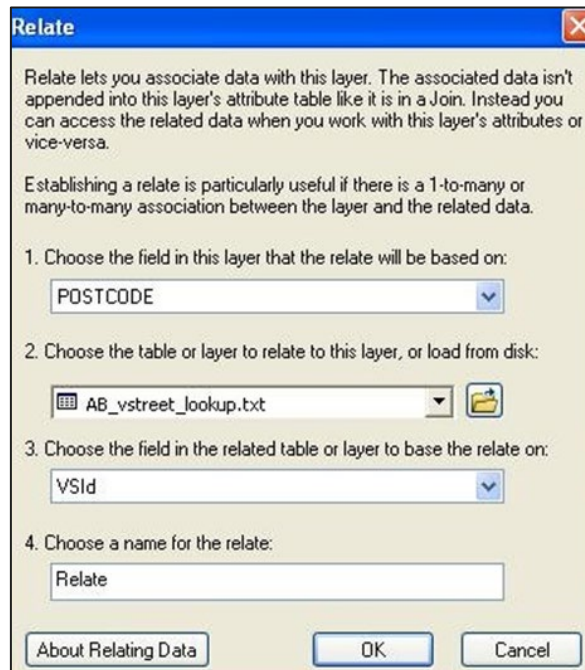
1. Load the postcode area Shapefile and relevant vertical street lookup file into ESRI ArcMap.

NOTE: it may be necessary to remove the underscores in the vertical street file name for it to load successfully into ArcMap.

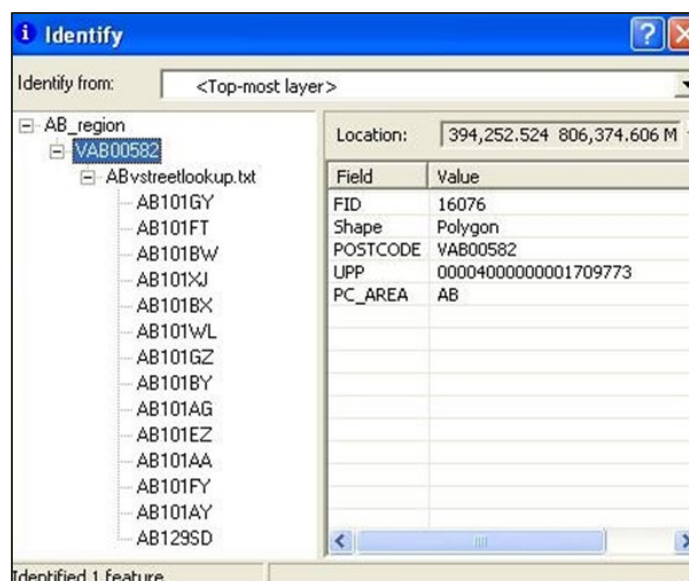
2. Right-click on the postcode area Shapefile, go to *Joins and Relates* and click on *Relate*.



The *Relate* window will open.



3. In option 1, click on the drop-down arrow and select 'POSTCODE'.
4. In option 2, select the vertical street lookup file.
5. In option 3, select Col2 or the vertical street ID column (this depends how you have named your columns).
6. In option 4, choose a suitable name for the relate.
7. Click OK.
8. When using the *Identify* tool in ESRI ArcMap, click on a vertical street polygon and expand the + signs on the left to see all the postcodes related to that vertical street polygon.



2.3 PostGIS and QGIS

PostGIS and QGIS can be used in conjunction with each other to create a vertical street lookup table, which can be visualised and interrogated within the QGIS GIS environment alongside the spatially enabled Code-Point postcode area polygons.

2.3.1 PostGIS steps

1. Load both the postcode area polygons and vertical street text file into PostGIS in their own separate tables within the same schema.
2. You will now create a new vertical street lookup table within PostGIS that contains the geometry of the vertical streets, obtained from the postcode area polygons table. Because you will be using the postcode columns from both the postcode area polygons table and the vertical street table you must ensure the postcode columns are named differently. The following SQL can be used to alter the name of the postcode column within the vertical street table:

```
alter table verticalstreets rename column postcode to vs_postcode;
```

3. Once you have ensured that all the columns in the two tables are named differently you can run the following script, which will create a new table which will contain all the vertical streets and their associated geometry:

```
create table vertical_street_lookup as select * from codepoint c, verticalstreets v where c.postcode=v.v_id;
```

4. This SQL creates a table by joining the vertical street IDs in the vertical street table with those IDs held in the postcode column of the postcode area polygons. The resulting table will effectively hold two columns of identical vertical street IDs, one from the vertical street table itself and one from the postcode column of the postcode area polygons table. The following script can be used to drop one of these columns, removing this duplicated information:

```
alter table vertical_street_lookup drop column postcode;
```

5. The postcode area polygons table can be altered to create a flag that alerts you to which postcodes form part of a vertical street. If you do not wish to create this flag, move onto the QGIS steps.
6. First you need to create a Boolean column within the postcode area polygons table to hold this information. Execute the following SQL to achieve this:

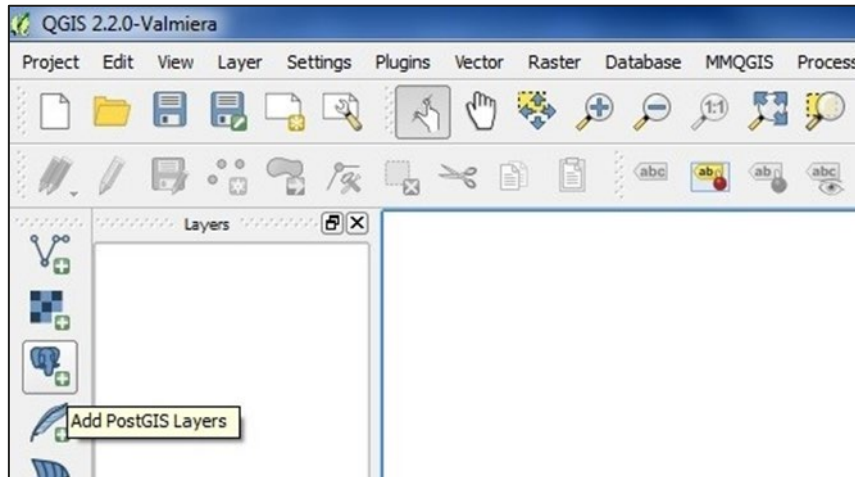
```
alter table codepoint add column vs boolean;
```

7. This column then can be updated with a value of 'true' where a postcode in the postcode area polygons table forms part of a vertical street. The following SQL can be used to do this:

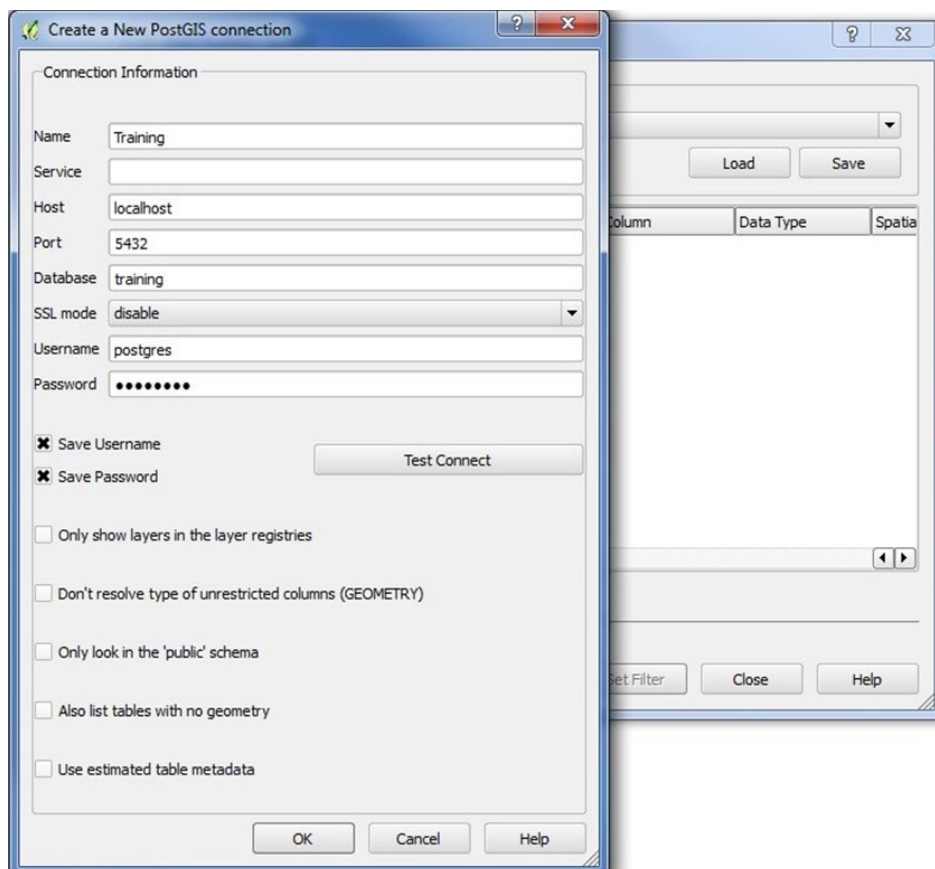
```
update codepoint  
set vs = 'True' where postcode in (select c.postcode from vertical_street_lookup v, codepoint c where  
v.vs_postcode=c.postcode or c.postcode=v.v_id);
```

2.3.2 QGIS steps

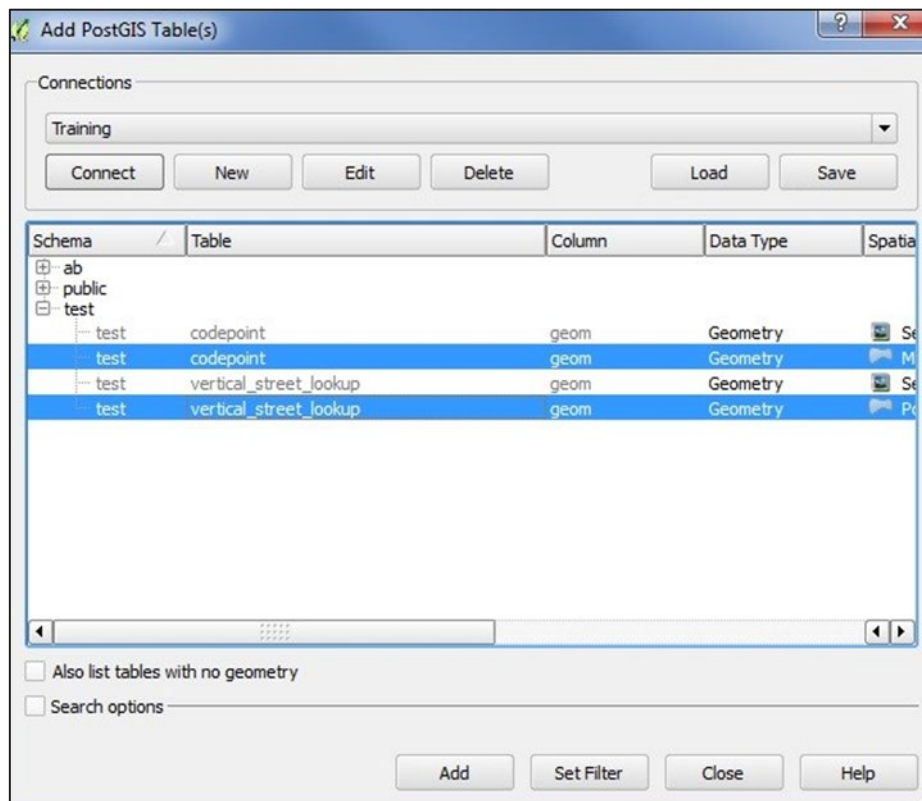
Once you have created a new vertical street lookup table with a geometry column as outlined in the steps above, this can be visualised within QGIS using the following steps:



1. Connect to PostGIS by selecting the Add PostGIS Layers button.
2. This will bring up an Add PostGIS Table(s) dialogue box. If you have not already connected to your PostGIS database select the New option and fill in the Create New PostGIS Connection dialogue box.



- Once you have added your connection details, select the *Test Connect* option to ensure the details are correct. When your connection is successful, click OK to save the connection, it should then appear in the connections drop-down. Select Connect and the schemas within your database will be populated. Expand the schema you are using and select both the postcode area polygons table and the vertical street lookup table you have created. Add them to QGIS by selecting Add at the bottom of the dialogue box. You will be prompted to select a coordinate reference system to use from a list which can be filtered by SRID.



- Once your coordinate system has been selected the two tables should be displayed spatially in QGIS. If you have chosen to add the extra Boolean column to the postcode area polygons table this flag will alert you to when a postcode is part of a vertical street. The vertical street lookup table can then be used to display all the postcodes contained within the vertical street.

3. Creating single-space postcodes

3.1 Centroids and polygons

3.1.1 Outward and inward bound representation

The current specification represents the postcodes in a set format that defines the postcodes as having an inward and outward postcode 'code'. Code-Point and Code-Point with polygon postcodes have 0, 1 or 2 spaces between the in and out code.

The table below identifies how postcodes are currently shown in the data.

| Postcode structure | Number of spaces |
|--------------------|---------------------------------------------------------|
| AANNNA | 0 spaces (represented as AANNNA) for example: PO143RW |
| ANN NAA | 1 space (represented as ANN<>NAA) for example: PO14 3RW |
| AN NAA | 2 spaces (represented as AN<><>NAA) for example: B1 5AP |

Single-space postcodes

The Code-Point and Code-Point with polygons postcodes are currently represented as above; however, there may be a user requirement to represent each postcode in a uniformed single-space format.

The aim of this section is to offer some guidance on how to process the Code-Point and the related Code-Point with polygons data to generate postcodes with a single space.

The single-space instructions are applicable to both the postcode point and unit polygon products. Microsoft Excel, Microsoft Access®, MapInfo, ESRI and QGIS GIS formats have been included to provide guidance when using comma-separated values (CSV) and other formats.

The underlying theory for all of the methods is principally the same, in that all current spaces are removed and then a single space added before the third character from the right.

NOTE: the NTF format is not included in this chapter as it is not compatible to a single-space format.

These instructions apply to postcode units only, and not to vertical streets (which are only found in the Code-Point with polygons dataset). To ensure that the vertical street references are not corrupted, remove them from your data before applying these instructions.

CSV single-space postcodes using Microsoft Excel and Access

1. Open the CSV file with Excel so that the data is displayed in columns.
2. In a new blank column, click in the first cell of the new column (excluding the row column names).
3. In the function line, enter in the following function command where A1 is the column containing the postcode:

```
=TRIM(LEFT(A1,LEN(A1)-3))&" "&RIGHT(A1,3)
```

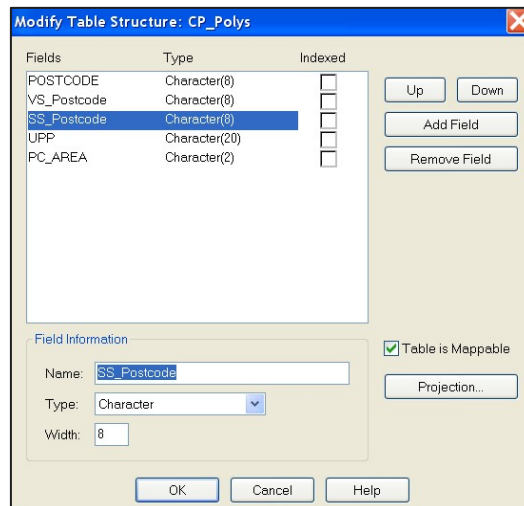
This should now produce a column containing postcodes with a single space. For polygon data, extend this formula downwards, stopping when you reach any vertical street references; for example, those beginning with 'V'.

The same method can be employed in Microsoft Access, using an update query rather than the function line. The functions listed above are the same for Access and Excel.

MID/MIF and TAB single-space postcodes using MapInfo

The process within MapInfo is the same regardless of whether the original supply is in TAB or MID/MIF, as both get imported to MapInfo and opened as a .TAB file.

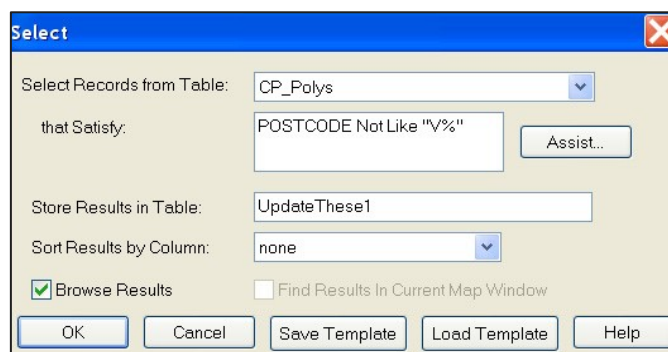
1. Create a new column, of Character (8) type, in the Code-Point Polygons table (CP_Polys.TAB) as column 3. Give it a meaningful name like 'SS_Postcode' (go to Table > Maintenance > Table Structure). Save the table.



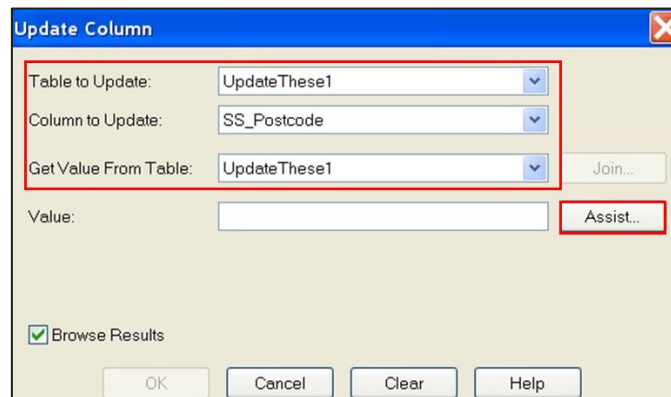
2. Click on Query, Select – this brings up the Select Query dialogue box. Ensure that the Code-Point Polygons table, CP_Polys, is being selected from and enter the following expression in the that Satisfy field:

POSTCODE not like "V%"

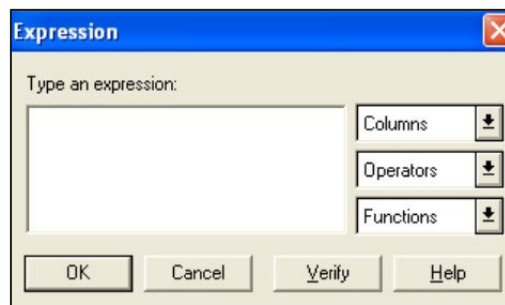
3. Change the Store Results in Table field to read 'UpdateThese1'. Click the OK button to apply the query.



- The next step is to update the SS_Postcode attribute. Click Table and then the Update column tab. This opens the following dialogue box:



- Ensure that the *Table to Update* field has 'UpdateThese1' as its value. From the *Column to Update* drop-down menu, select the previously-added column; for example, 'SS_Postcode', making sure the *Get Value from Table* is the same table as in the *Table to Update* field.
- Click the Assist button. This opens the following *Expression* dialogue box:



- In the Type an expression box, key in the following function command:

RTrim\$(Left\$(<insert column name of original postcodes>*, (Len*<insert column name of original postcodes>* (*<insert column name of original postcodes>*e)-3)))+ " "+Right\$(*<insert column name of original postcodes>*,3)*

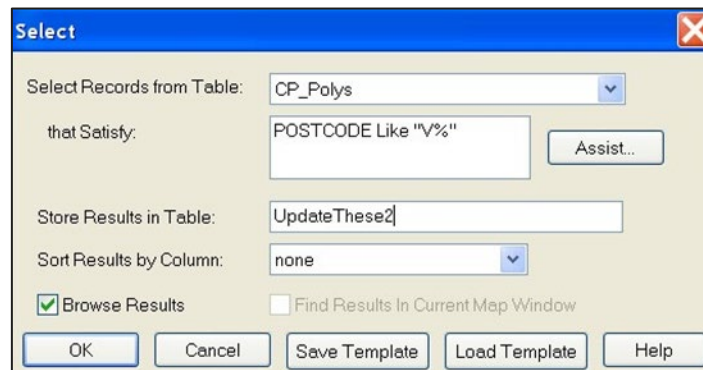
- Click the OK button to apply the update.

This process must then be re-run for the postcodes that were originally vertical streets also:

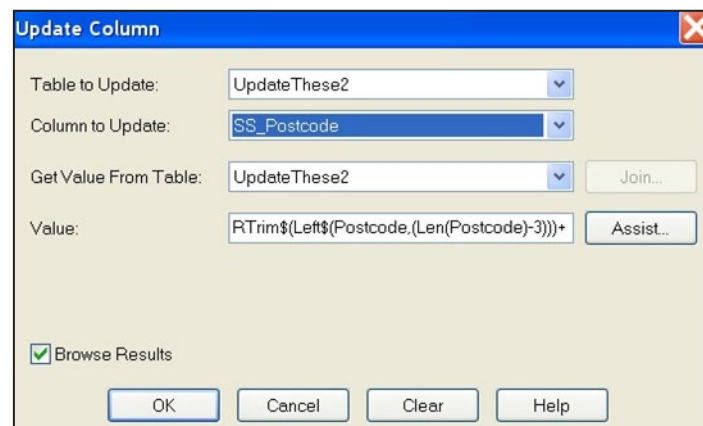
- Click on Query > Select – this brings up the Select Query dialogue box. Ensure that the Code-Point Polygons table, CP_Polys, is being selected from and enter the following expression in the that Satisfy field:

POSTCODE like "V%"

2. Change the Store Results in Table field to read 'UpdateThese2'. Click the OK button to apply the query.



3. The next step is to update the SS_Postcode attribute. Click Table and then the Update column tab. This opens the following dialogue box:



4. Ensure that the Table to Update field has 'UpdateThese2' as its value. From the Column to Update drop-down menu, select the previously-added column; for example, 'SS_Postcode', making sure the Get Value from Table is the same table as in the Table to Update field.
5. Click the Assist button and in the Type an expression box, key in the following function command:

```
RTrim$(Left$(<insert column name of original postcodes>,(Len(<insert column name of original postcodes>)-3)))+ " "+Right$(<insert column name of original postcodes>,3)
```
6. Click the OK button to apply the update.

If you now browse the Code-Point Polygons table you should see that the SS_Postcode column is fully populated with single-space postcodes for both 'standard' postcodes and vertical street postcodes. You can choose to delete the VS_Postcode column at this point, as the new SS_Postcode contains all the single-spaced postcodes for all features.

Shapefile single-space postcodes using ESRI ArcGIS

These steps assume that the data has already been imported into ArcGIS and that the user has the correct permissions to edit the Shapefile. It is advised to try this method on a copy of the original data, and not the only copy of the data, in case of error.

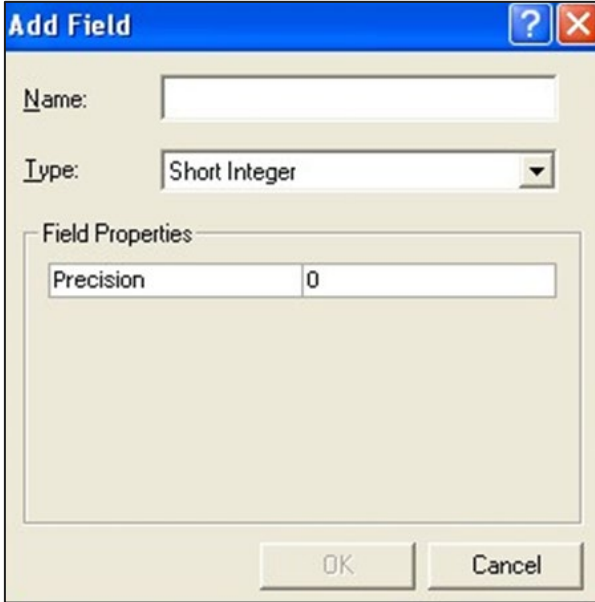
Once the Shapefile is open, the required new fields can be added.

1. Right-click on the layer in the Table of Contents (down the left-hand side of the map window) and open the Open Attribute Table.

The first step is to add a new column to hold the newly formatted postcode.

2. Click on the Options button then click on Add Field.

This opens the following dialogue box:



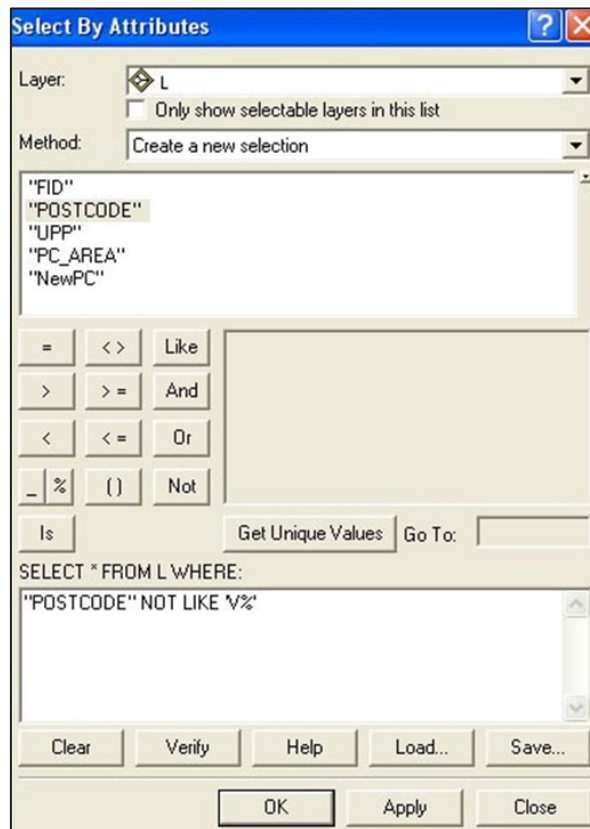
The screenshot shows the 'Add Field' dialog box. The 'Name' field is empty. The 'Type' dropdown is set to 'Short Integer'. The 'Field Properties' section has a 'Precision' field set to '0'. The 'OK' and 'Cancel' buttons are at the bottom.

3. Key in an appropriate name, for example, 'NewPC', in the Name box and change the Type box on the drop-down menu to 'text'; also change the Field Properties Precision/Length to '8'.
4. Click OK and the field is added.

The next stage is to ensure that only postcode units are updated and not any vertical street references that might also be in the table.

5. Click on *Selection, Selection By Attributes*.

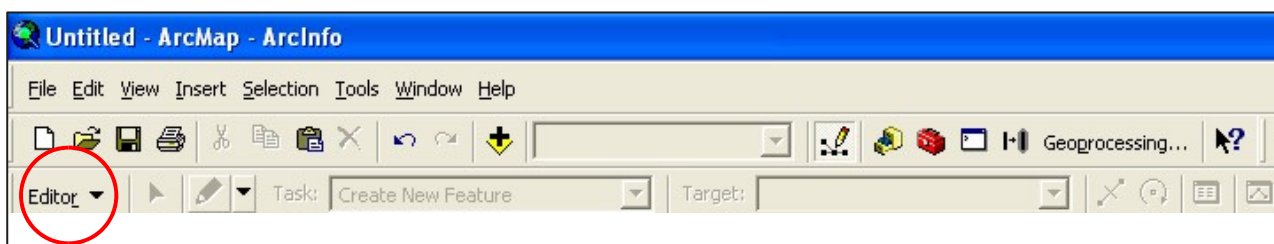
This brings up the *Select By Attributes* dialogue box:



6. Ensure that the correct Layer is being selected from, in this example, 'L'.
7. Ensure that the Method field you have chosen is 'Create a new selection'.
8. Then, enter the following expression in the query field at the bottom of the dialogue box:
`"POSTCODE" NOT LIKE 'V%'`
9. Click the OK button to apply the query. This selects all the postcodes that are not vertical streets from the attribute table.

The final stage is to update the new column:

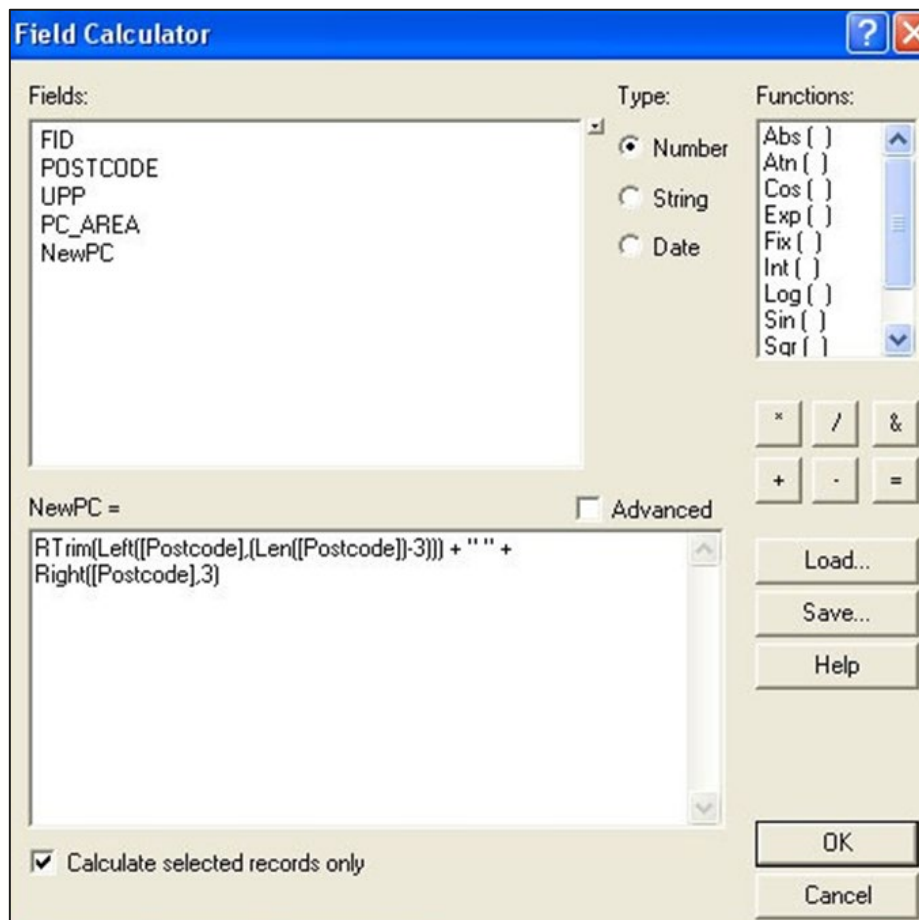
1. To populate the fields, the table must be made editable.
2. Return to the map window. Do not close the attribute table as it will be required later.
3. Click on the *Editor* drop-down selection = ▼.



If this toolbar is not already loaded, then right-click on an empty part of the grey area on the map window and all the available toolbars will be listed. Simply click on the Editor toolbar and it will be loaded to the toolbar.

4. Click the Editor drop-down and select the first option, Start Editing.
5. Once Start Editing has been selected, return to the attribute table.
6. Right-click on the column name of the column added previously; for example, 'NewPC', and select Calculate Values.

This opens the following Field Calculator dialogue box:



7. In the bottom dialog box, enter the following function command:
$$RTrim(Left([<insert column name of original postcodes>],(Len([<insert column name of original postcodes>])-3)))+" "+Right([<insert column name of original postcodes>],3)$$
8. Ensure that the lower-left tick box Calculate selected records only is checked.
9. Click OK to update the column. This will update your new column with a single-space postcode.
10. Finally, go back to the map window, click on the Editor toolbar and select Stop editing. It will prompt to save the edits. Click Yes.

Shapefile single-space postcodes using QGIS

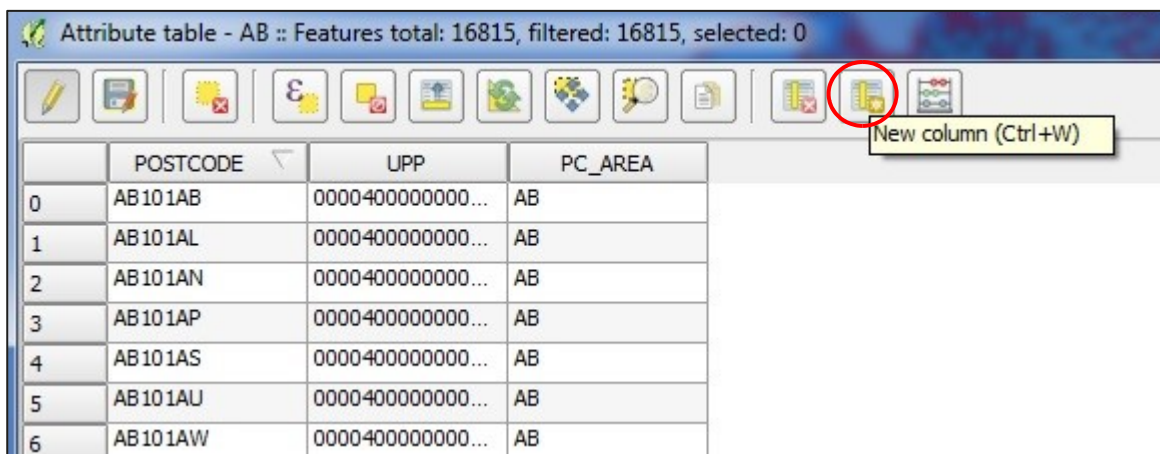
These steps assume that the data has already been imported into QGIS and that the user has the correct permissions to edit the Shapefile. It is advised to try this method on a copy of the original data, and not the only copy of the data, in case of error.

Once the Shapefile has been opened in QGIS it can be edited and a new field with the reformatted postcodes added.

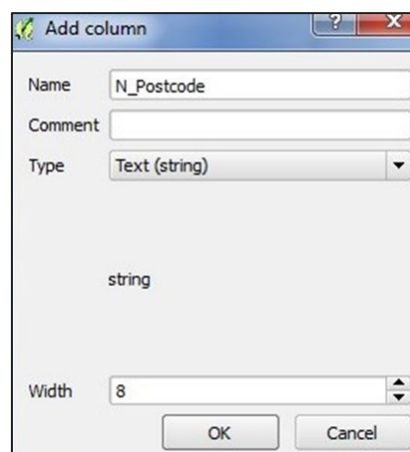
1. Right-click on the layer you wish to edit in the *Layers* window and select *Open Attribute Table*.

A new column must be created to hold the newly formatted postcodes.

2. Select the Toggle Editing Mode button in the top left-hand corner of the attribute table. This will activate a range of other buttons in the top bar. Select the New Column button in the top right-hand corner of the attribute table.



3. The following dialogue box will open, prompting you to set the name and parameters of the field. Ensure you set the field type as Text (String) and set the width to 8 characters long as shown below.

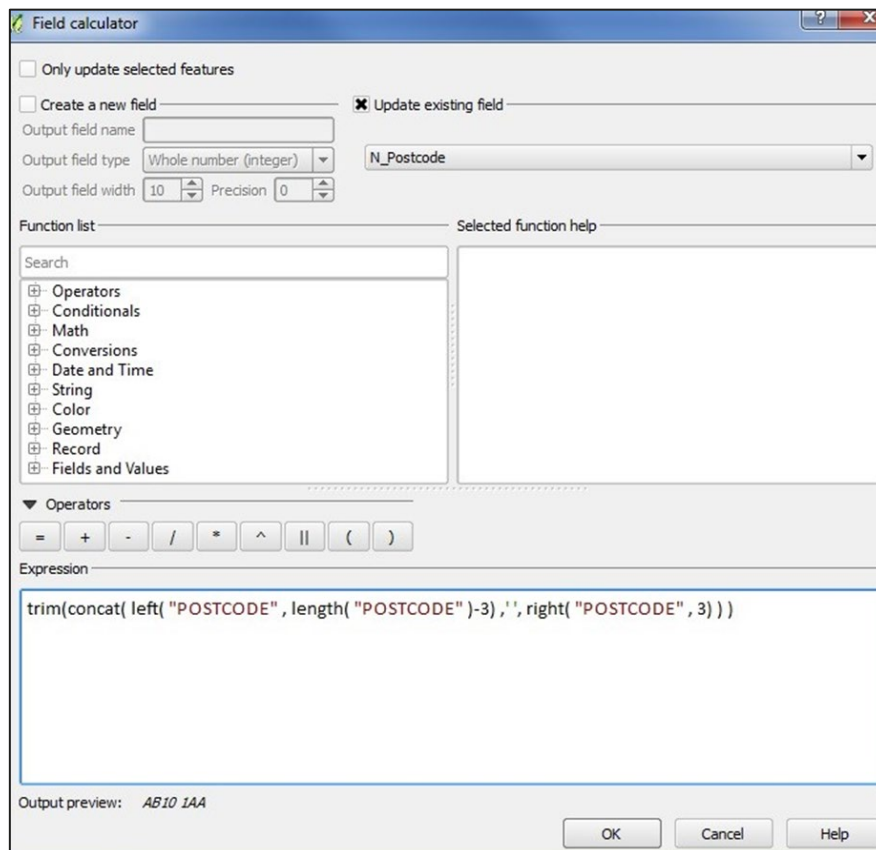


4. When the parameters have been set click OK. The extra field will be created and will contain null values. To populate this field, select the Open Field Calculator button in the top right-hand corner of the attribute table. The following dialogue box will be opened.

5. Click the checkbox Update existing field and then choose the field you have created from the drop-down underneath. In the Expression box copy and paste the following code:

```
trim(concat( left( "<insert column name of original postcodes>" , length( "<insert column name of original postcodes>" )-3) , ',' , right( "<insert column name of original postcodes>" , 3) ) )
```

See image below for example.



6. After selecting OK, the dialogue box will close, and the new column will be updated with single-spaced postcodes.

4. Data measures

Ordnance Survey measures the data in its products in one or more of the ways set out in table 1 below.

Table 1: Definitions of data measures.

| Data measure | Definition | Sub-measure | Definition |
|-----------------------------------------------|---------------------------------------------------------------------------------------|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Completeness | Presence and absence of features against the specified data content* | Omission | Features representing objects that conform to the specified data content but are not present in the data. |
| | | Commission | Features representing objects that do not conform to the specified data content but are present in the data. |
| Logical consistency | Degree of adherence to logical rules of data structure, attribution and relationships | Conceptual consistency | How closely the data follows the conceptual rules (or model). |
| | | Domain consistency | How closely the data values in the dataset match the range of values in the dataset specification. |
| | | Format consistency | The physical structure (syntax): how closely the data stored and delivered fits the database schema and agreed supply formats. |
| | | Topological consistency | The explicit topological references between features (connectivity) – according to specification. |
| Positional accuracy | Accuracy of the position of features | Absolute accuracy | How closely the coordinates of a point in the dataset agree with the coordinates of the same point on the ground (in British National Grid EPSG: 27700 for shapefile, TAB, MID/MIF and GeoPackage formats, and Web Mercator projection EPSG: 3857 for vector tile format). |
| | | Relative accuracy | Positional consistency of a data point or feature in relation to other local data points or features within the same or another reference dataset. |
| | | Geometric fidelity | The ‘trueness’ of features to the shapes and alignments of the objects they represent ¹ . |
| Temporal accuracy | Accuracy of temporal attributes and temporal relationships of features | Temporal consistency | How well-ordered events are recorded in the dataset (life cycles). |
| | | Temporal validity (currency) | Validity of data with respect to time: the amount of real-world change that has been incorporated in the dataset that is scheduled for capture under current specifications. |
| Thematic accuracy (attribute accuracy) | Classification of features and their attributes | Classification correctness | How accurately the attributes within the dataset record the information about objects ¹ . |

¹ When testing the data according to the dataset specification against the ‘real world’ or reference dataset.

