



ORDNANCE SURVEY GB

# GETTING STARTED GUIDE

## – ADDRESSBASE, ADDRESSBASE PLUS AND ADDRESSBASE PLUS ISLANDS

## Version history

Version	Date	Description
2.0	04/2020	Revised document. Includes ArcGIS Pro.
2.1	03/2021	Revision to Sections 5, 6, 7 & 8; removal of OSMA.

## Purpose of this document

This is the Getting Started Guide for the AddressBase, AddressBase Plus and AddressBase Plus Islands products. This Guide provides greater insight into these products and their potential applications. For information on the contents and structure of AddressBase, AddressBase Plus and AddressBase Plus Islands, please refer to the Product Guide and Technical Specification. Unless otherwise indicated, the instructions for AddressBase Plus are also applicable to AddressBase Plus Islands.

The terms and conditions on which AddressBase, AddressBase Plus and AddressBase Plus Islands are 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 AddressBase, AddressBase Plus and/or AddressBase Plus Islands.

We may change the information in this Guide 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 Guide.

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 Guide

© Ordnance Survey Limited 2021. This Guide (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 Guide may be copied for use internally in your organisation or business so that you can use AddressBase, AddressBase Plus and AddressBase Plus Islands under the terms of your licence (but not otherwise).

No part of this Guide 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.

The data in AddressBase is provided by GeoPlace, a limited liability partnership between the Local Government Association and Ordnance Survey.

AddressBase and AddressBase Plus contain data created and maintained by English, Welsh and Scottish Local Government as well as Royal Mail and Valuation Office Agency:

- © Local Government Information House Limited copyright and database rights 2021, or © Hawlfraint a hawliau cronfa ddata cyfyngedig Tŷ Gwybodaeth ar Lywodraeth Leol 2021.
- PAF data are copyright material of Royal Mail Group Limited.

- Valuation Office Agency data is provided under the Open Government Licence and other data provided under the Open Government Licence v3.0.2.

Elements of the AddressBase Plus Island data are provided by the following authorities:

- Northern Ireland data provided by Land & Property Services © Pointer copyright and / or database right 2021.
- Channel Island Address information is the intellectual property of The States of Guernsey and / or The States of Jersey. Copyright © 2021 States of Guernsey / Jersey.
- Address information derived from Isle of Man Government Property Database. © Crown Copyright, Treasury, Isle of Man.

### **Trademarks**

Ordnance Survey, OS, the OS Logos and AddressBase are registered trademarks, and AddressBase Plus and AddressBase Plus Islands are trademarks of Ordnance Survey, Britain's mapping agency.

PAF, PO Box, Post Office and Royal Mail are registered trademarks of Royal Mail Group Ltd.

### **Contact details**

<https://www.ordnancesurvey.co.uk/contact-us>

# Contents

<b>1. Introduction.....</b>	<b>6</b>
1.1 Overview.....	6
<b>2. Prerequisites.....</b>	<b>6</b>
2.1 System requirements.....	6
2.2 Backup provision of the product.....	7
2.3 Typical data volumes .....	7
<b>3. Data supply.....</b>	<b>8</b>
3.1 Supply options .....	8
3.1.1 DVD Supply of area of interest.....	8
3.1.2 DVD Supply of Managed Great Britain Sets .....	8
3.1.3 Secure File Transfer Protocol .....	8
3.1.4 Download.....	9
3.2 Chunked files.....	9
3.2.1 Non-geographic chunks .....	9
3.2.2 Geographic chunks.....	9
3.3 Unzipping the data.....	10
3.3.1 Non-geographic chunks.....	10
3.3.2 Geographic chunks .....	10
<b>4. Preparing the CSV data .....</b>	<b>11</b>
4.1 Merging multiple CSV files .....	11
4.2 Appending a header file to the CSV .....	12
<b>5. Loading CSV into GIS software.....</b>	<b>13</b>
5.1 Loading CSV into ArcGIS Pro .....	13
5.2 Loading CSV into ArcGIS Desktop.....	19
5.3 Loading CSV into MapInfo Pro.....	21
5.4 Loading CSV into QGIS .....	22
<b>6. Loading CSV into a database .....</b>	<b>24</b>
6.1 Considerations.....	24
6.1.1 Software dependencies .....	24
6.1.2 UPRN deletions.....	24
6.2 Loading CSV into a PostgreSQL database .....	24
6.2.1 Converting coordinates to geometry.....	26
6.3 Loading CSV into an Oracle database.....	26
6.4 Loading CSV into Microsoft SQL Server .....	28
6.4.1 Setting Primary Keys.....	30
6.4.2 Creating the point geometry .....	30
<b>7. Loading GML .....</b>	<b>30</b>
<b>8. COU implementation guidance .....</b>	<b>31</b>
8.1 Introduction.....	31

8.2	Types of change .....	31
8.3	High-level COU implementation model .....	31
8.4	High-level COU implementation model – with archiving.....	32
8.5	Applying COU to tables .....	32
<b>9.</b>	<b>Creating a Single-Line or Multi-Line Address .....</b>	<b>34</b>
9.1	Introduction.....	34
9.2	Background .....	35
9.3	Delivery Point Address (PAF Address).....	35
9.3.1	Building a single-line Delivery Point Address.....	37
9.3.2	Building a multi-line Delivery Point Address .....	38
9.4	Geographic Address (Local Authority Address) .....	38
9.4.1	Constructing a single address label from the Geographic Address fields.....	39
9.4.2	Rendering SAOs and PAOs .....	40
9.4.3	Street description, town, locality and administrative area names .....	41
9.4.4	Building a single-line Geographic Address.....	42
9.4.5	Building a multi-line Geographic Address.....	43
9.5	Creating mailing lists.....	45
9.5.1	Other filters.....	45
<b>10.</b>	<b>Searching for addresses.....</b>	<b>46</b>
10.1	Free text search.....	46
10.2	Structured component search .....	47
10.3	The search operation .....	47
10.4	Generating a search query from structured user input.....	48
10.5	Structuring the query for a structured address search.....	50
10.6	Supporting case-insensitive queries and partial matches .....	51
10.7	Generating a search query from unstructured user input .....	51
10.7.1	Creating a search index for addresses .....	51
10.7.2	Structuring the query for an unstructured address search .....	52
10.8	Summary .....	54
<b>Annex I –</b>	<b>Support links.....</b>	<b>55</b>

# 1. Introduction

## 1.1 Overview

AddressBase products are created by bringing together different address sources:

- Local Authority Gazetteers across Great Britain, Northern Ireland, the Channel Islands and the Isle of Man
- Royal Mail PAF data
- References to Valuation Office Agency (VOA) data
- Additional addresses and coordinates from Ordnance Survey

The instructions contained in this document show you how to use the AddressBase, AddressBase Plus and AddressBase Plus Islands products. The data is supplied as comma-separated values (CSV) or Geography Markup Language (GML). This guide covers:

1. The pre-processing steps required to attach the header file to the data.
2. Instructions on how to load the data into various geographic information system (GIS) software applications and/or databases. It is important to note that this is not an extensive list of the applications that can be used to load the product as numerous GIS software applications and/or database can be used for this purpose.
3. How to apply a change-only update (COU).

Advice is given towards the end of this guide on how to create a single line address from the various address sources in the product and how, in turn, this can be used to create a look-up service for a customer to search for an address.

## 2. Prerequisites

### 2.1 System requirements

AddressBase data is an addressing gazetteer that can be used within GIS and database systems. For details of Ordnance Survey's licensed partners, who can incorporate the AddressBase products in their systems, please see the [systems/software page](#) on the Ordnance Survey website.

Ordnance Survey does not recommend either suppliers or software products as the most appropriate system will depend on many factors, such as the amount of data being taken, resources available within the organisation, the existing and planned information technology infrastructure and the applications that AddressBase products can be used for.

However, as a minimum, the following elements will be required in any system:

- A means of reading the data, either in its native format, or by translating it into a file format or for storage in a database.
- A means of storing and distributing the data, perhaps in a database or through a web-based service.
- A way of visualising and querying the data, typically a GIS.

The versions of software referred to in this guide are as follows:

- ArcGIS Pro version 2.3.3
- ArcGIS Desktop versions 9.3 and 10
- ArcMap version 10
- MapInfo Professional version 12
- PostgreSQL version 1.12.3
- QGIS version 2.6

## 2.2 Backup provision of the product

You are advised to copy the supplied data to a backup medium.

## 2.3 Typical data volumes

For reading purposes, it is recommended to store the data on a single hard disc. This will speed up the ability of your computer to read the data. Unzipped file sizes for the full supply of each product are as follows:

Product	Unzipped CSV file size	Unzipped GML file size
AddressBase	6Gb	32Gb
AddressBase Plus	16Gb	78Gb
AddressBase Plus Islands	450Mb	2Gb

## 3. Data supply

### 3.1 Supply options

#### 3.1.1 DVD Supply of area of interest

When you receive an order via hard media (DVD), the following files will be supplied for the contracted area of interest (AOI):

- Data
- Doc
- Order\_Details.txt

Within the Data directory, data files will be found in their compressed format.

Within the Doc folder, a text file called *Label Information.txt* will contain information that is printed on the DVD.

The Order\_Details text file will provide information about the order, including the order date, currency date and file structure.

#### 3.1.2 DVD Supply of Managed Great Britain Sets

When you receive an order of a Managed Great Britain Set (MGBS) via hard media (DVD), the following files will be supplied:

- Data
- Doc
- Resources
- readme.txt

There are several items contained within your supply:

- Data folder – This folder contains all of your data supply.
- Doc folder – This folder contains the Medialis.txt file, which outlines the contents of the data you have been supplied.
- Resources folder – This folder contains lookup tables for the local custodian code and AddressBase classification scheme as well as the Header files for the product.
- The readme text file – This document provides guidance notes on matters such as the filename referencing used and the directory structure of the DVD.

#### 3.1.3 Secure File Transfer Protocol

With a Secure File Transfer Protocol (SFTP) order, the same folder structure is supplied as in Section 3.1.1. The filenames will be slightly different, reflecting the SFTP order number, and the Docs folder will be empty.



### 3.1.4 Download

Public Sector Geospatial Agreement (PSGA) customers can download their geographic chunk data for AddressBase and AddressBase Plus as well as a full supply of AddressBase Plus Islands via our download service.

1. When you click *Download data*, you will be required to enter a password to access the PSGA Member's Area. On successful entry to the download service, you will be able to view all your orders in the Member's Area and download your data.
2. If you have ordered your data from our online portal, you will be sent an email with a link to a download page.
3. Within the PSGA Member's Area, you can order and download the data that you require by clicking on *Order Data*, which can be found under the *Map Data* heading.
4. Once you have selected *Order Data*, you will be presented with the Order page. From here, you can manage all your orders, including those for AddressBase products.
5. When you have placed an order for a product, the data will become available as a series of zipped data files. To unzip these files, please refer to [Section 3.3](#).

## 3.2 Chunked files

The data is supplied as chunked files that cover your selected area. These files are named according to the convention shown below.

When you open your data, you will see a series of zip folders:

### 3.2.1 Non-geographic chunks

Using AddressBase Plus and Islands as an example:

*AddressBasePlus\_FULL\_2020-01-21\_001\_csv.zip* (Full supply of GB CSV)

*AddressBasePlus\_ISL\_FULL\_2020-01-21\_001\_csv.zip* (Full supply of Islands CSV)

or

*AddressBasePlus\_COU\_2020-01-21\_001\_gml.zip* (COU supply of GB GML)

*AddressBasePlus\_ISL\_COU\_2020-01-21\_001\_gml.zip* (COU supply of Islands GML)

### 3.2.2 Geographic chunks

Using AddressBase Plus as an example:

*AddressBasePlus\_FULL\_2011-07-29\_TQ2020\_csv.zip* (Full supply of CSV)

or

*AddressBasePlus\_COU\_2011-07-29\_TQ2020\_gml.zip* (COU supply of GML)

Note – The AddressBase Plus Islands product is not available in geographic chunks.

## 3.3 Unzipping the data

The GML and CSV data is supplied in a compressed form (zip). Some software can access these files directly, while other software will require the files to be unzipped.

To unzip the zipped data files (.zip extension), use an unzipping utility found on most PCs, for example, WinZip. Alternatively, open-source zipping/unzipping software can be downloaded from the Internet, for example, 7-Zip.

When you unzip the files, the data will be extracted as CSV files, which are ready to use. For example, unzipping AddressBase Plus will extract files similar to the chunks below:

### 3.3.1 Non-geographic chunks

*AddressBasePlus\_FULL\_2020-01-21\_001.csv*  
*AddressBasePlus\_ISL\_FULL\_2020-01-21\_001.csv*

(COU supply)

### 3.3.2 Geographic chunks

*AddressBasePlus\_2011-07-29\_NC4040.csv*

## 4. Preparing the CSV data

You can follow these instructions for the AddressBase, AddressBase Plus and AddressBase Plus Islands products. Where any steps differ between AddressBase and AddressBase Plus, this will be clearly indicated. Please assume all instructions for AddressBase Plus also apply to AddressBase Plus Islands, unless specifically stated.

### 4.1 Merging multiple CSV files

1. Unzip all the CSV files into a single folder as described in [Section 3.3](#). Ensure there are no spaces in your chosen folder path, for example: `C:\AddressBase_Data` or `C:\AddressBase_Plus_Data` or `C:\AddressBase_Plus_Islands_Data`.
2. We recommend merging all the CSV files together to save time importing individual files. This can be done manually using a text editor such as Notepad or TextPad, but it is much faster to use a .bat batch file or an MS-DOS command as described below.
3. To use the .bat batch function, copy the following text and paste it into a new Notepad document:

```
copy *.csv mergedABdata.csv
```

Note – In this example, *mergedABdata.csv* is the output name of the merged file which will be created, but this can be any user-defined filename with the extension .csv.

4. Save the Notepad document with the file extension .bat (for example, *mergedABdata.bat*) in the same directory as the CSV files unzipped in [Section 3.3](#) (for example, `C:\AddressBase_Data` or `C:\AddressBase_Plus_Data`).
5. Close the .bat file and navigate to the directory where you just saved it. Double-click on the .bat file (for example, *mergedABdata.bat*) and an MS-DOS window will run. Once the process is complete, the MS-DOS screen will close automatically.
6. If you look back in the directory containing the AddressBase CSV files and batch file, you'll see that there is now an additional single file called *mergedABdata.csv* (or the user-defined filename you picked when creating your batch file).

## 4.2 Appending a header file to the CSV

AddressBase and AddressBase Plus contain different attributes. This means that there is a separate header file for each of these. Header files for each product are supplied by Ordnance Survey and should be downloaded from their respective support page:

- AddressBase:  
<https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-support>
  - AddressBase Plus:  
<https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-plus-support>
  - AddressBase Plus Islands:  
<https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-islands-support>
1. Download and save the appropriate product CSV header file into the same folder as the merged AddressBase.csv file created in [Section 4.1](#).
  2. For AddressBase data, copy the relevant text below and paste it into a new Notepad document:

```
copy addressbase-header.csv+ mergedABdata.csv AB_Data.csv
```

Or, for AddressBase Plus and AddressBase Plus Islands data, copy the relevant text below and paste it into a new Notepad document:

```
copy addressbase-plus-header.csv+ mergedAB_Plusdata.csv AB_Plus_Data.csv
```

```
copy addressbase-plus-header.csv+ mergedAB_Plus_Islands_data.csv AB_Plus_Islands_Data.csv
```

Note – These examples use the name *mergedABdata.csv* or *mergedAB\_Plusdata.csv* as the file that contains the AddressBase data merged into a single CSV file created in [Section 4.1](#). If you have named this something else, amend that text above accordingly.

The order that the documents are referred to in the above text is also important as it states which file is appended to the other. In this instance, the headers CSV file comes first so that the column headers are the first line of the final AddressBase file and the merged data is appended to the column headers.

3. Save the above Notepad document with the file extension .bat (for example, *append.bat*) in the same directory as the column headers and the merged AddressBase data (for example, *C:\AddressBase\_Data* or *C:\AddressBase\_Plus\_Data* or *C:\AddressBase\_Plus\_Islands\_Data*).
4. Close the .bat file and navigate to the directory where it was saved to (for example, *C:\AddressBase\_Data* or *C:\AddressBase\_Plus\_Data*). Double-click on the new .bat file (for example, *append.bat*) and an MS-DOS window will appear. Once the process is complete, the MS-DOS screen will close automatically.
5. Navigate to the directory where the column headers and the merged AddressBase data are located. You will see that a new CSV file has been created, which is the merged column headers and AddressBase data (for example, *AddressBase.csv* or *AddressBase\_Plus.csv*).

## 5. Loading CSV into GIS software

These instructions describe how to load the CSV format of AddressBase, AddressBase Plus and AddressBase Plus Islands data. In these examples, AddressBase Plus data will be used to describe the procedures in various GI systems.

It is assumed that the preparation of the AddressBase, AddressBase Plus or AddressBase Plus Islands CSV data has been carried out as instructed in [Section 4](#) before attempting to load the data. If it has not been done, the full set of data will not load, and data loaded will not contain header information.

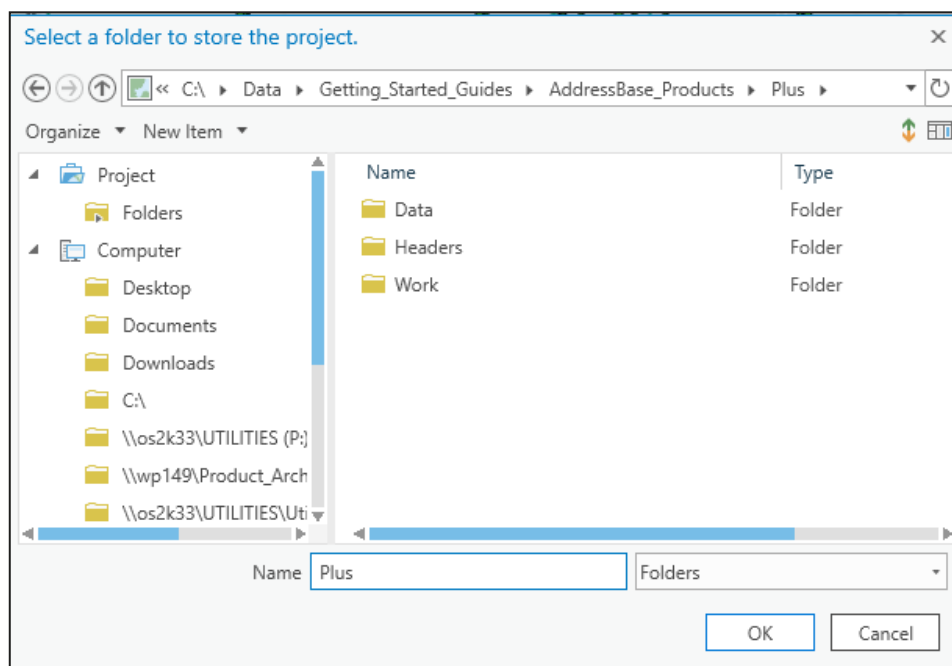
Note – AddressBase, AddressBase Plus and AddressBase Plus Islands are also available from Ordnance Survey as a supply in GML format. Loading GML into most GI applications often requires the use of third-party translation software, which is not covered in this guide. If more information is required in the loading of GML format, please [contact Ordnance Survey](#).

### 5.1 Loading CSV into ArcGIS Pro

These instructions are based on ArcGIS Pro version 2.3.3.

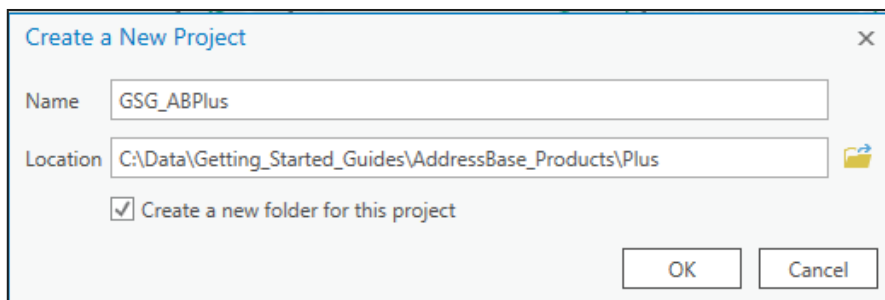
When using CSV data in ArcGIS Pro, it is necessary to have column headings. Please ensure that headings have already been prepared as instructed in [Section 4](#).

1. Launch ArcGIS Pro and start a new blank project.
2. Select a folder to save the project to.

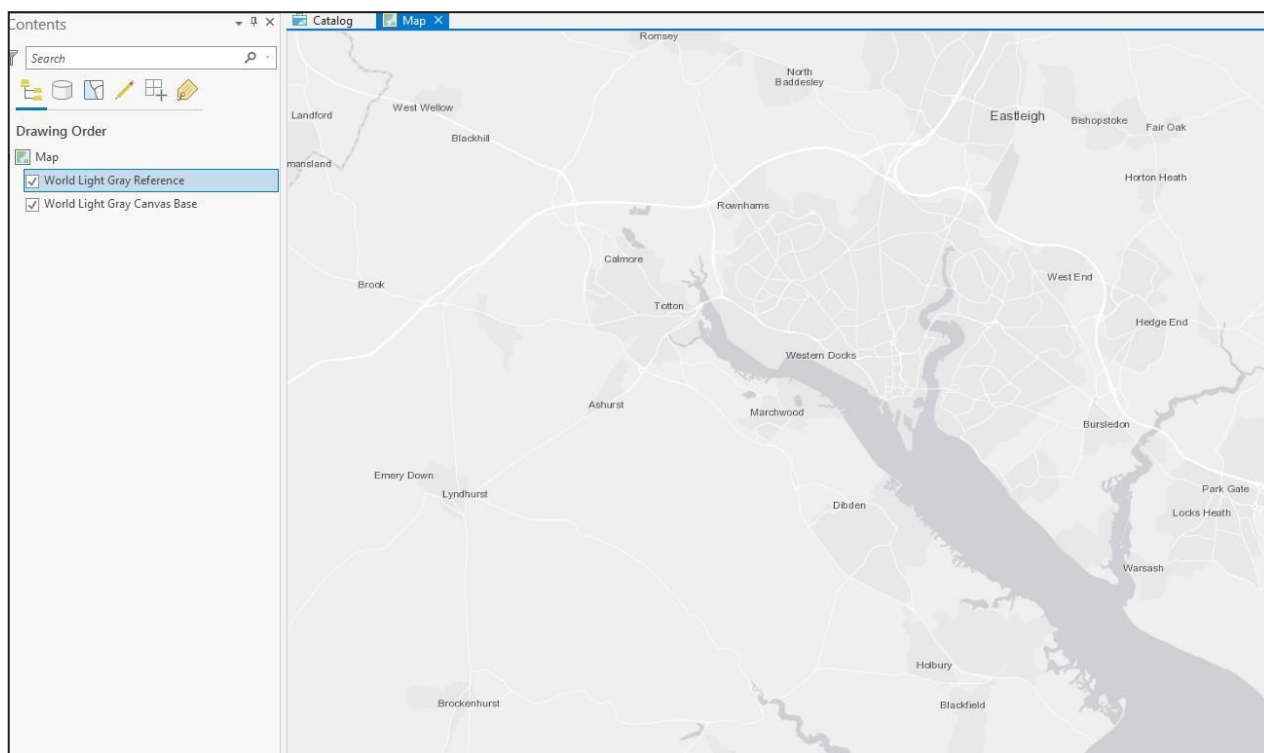


3. Name your project and click OK. The project will then be created.

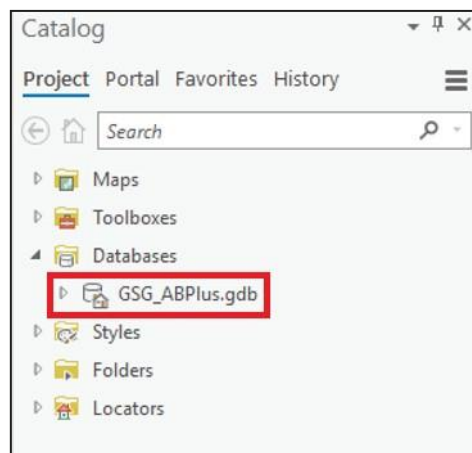
Note – ArcGIS Pro automatically creates a new File Geodatabase (.gdb) within the project folder created. This is different to the creation process in the older ESRI application ArcMap.



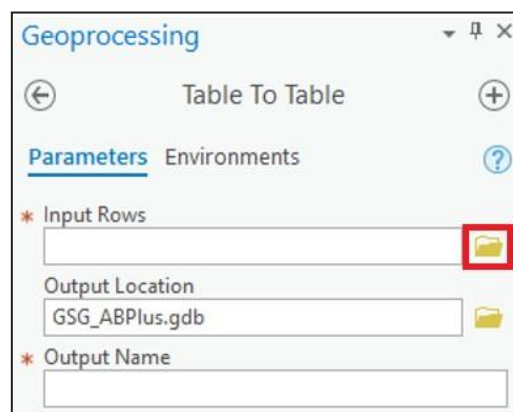
4. You can add a backdrop map for contextual purposes from the available backdrop maps supplied by ESRI or add one of your own from a different File Geodatabase. In this example, we have added a light grey backdrop map canvas supplied by ESRI.



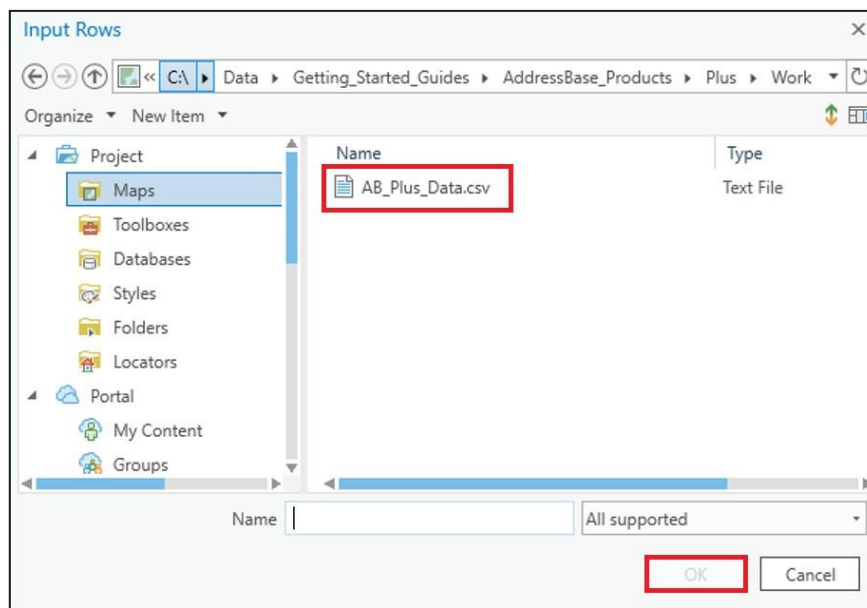
5. Open the *Catalog* pane on the right-hand side of the window and expand the listing to see the File Geodatabase created with the project.



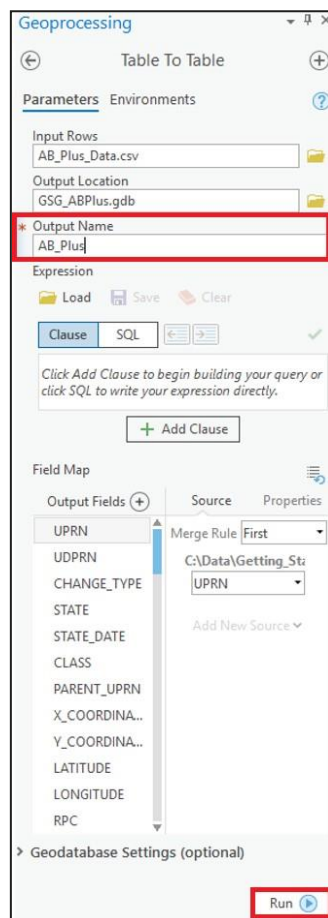
6. To import the AddressBase or AddressBase Plus data, right-click the File Geodatabase, then select *Import* and from that sub-menu, select *Table*. A new geoprocessing window will appear in the right-hand pane.
7. Click the folder icon on the right-hand side of the *Input Rows* field. A new window will appear.



8. Navigate to the location with the merged AddressBase or AddressBase Plus CSV file with the appended headers that you prepared in [Section 4](#). Select the file and click OK.

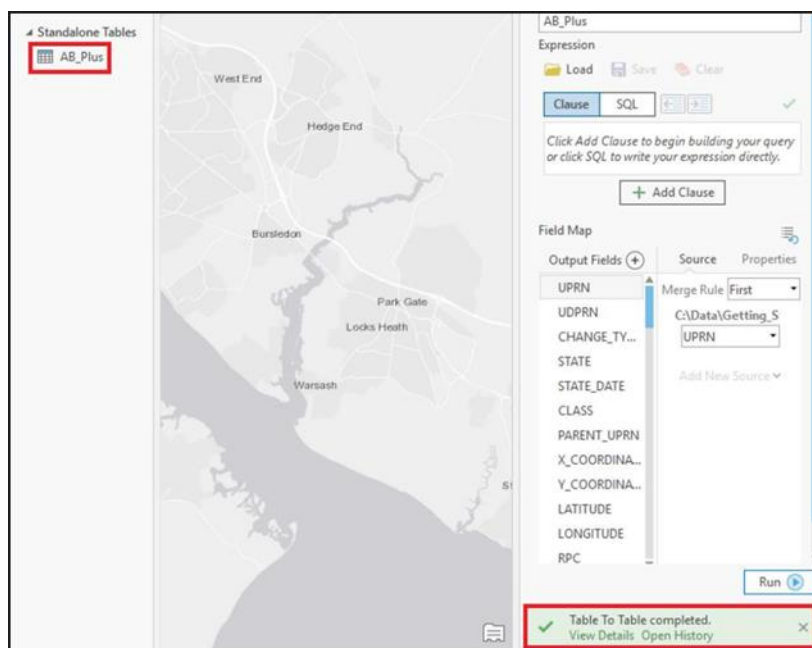


9. Back in the Geoprocessing window, type a name in the *Output Name* field, then click *Run* at the bottom of the window.



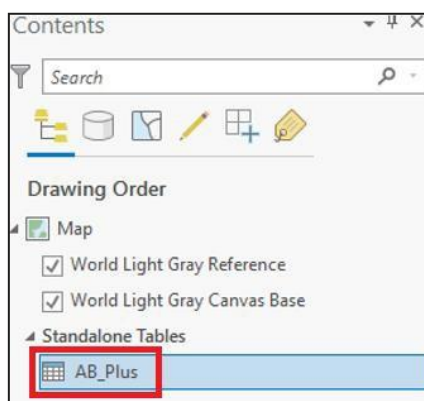


10. Once the process has run, a green box will appear at the bottom of the Geoprocessing window and the new AddressBase table will be listed in the left-hand panel.

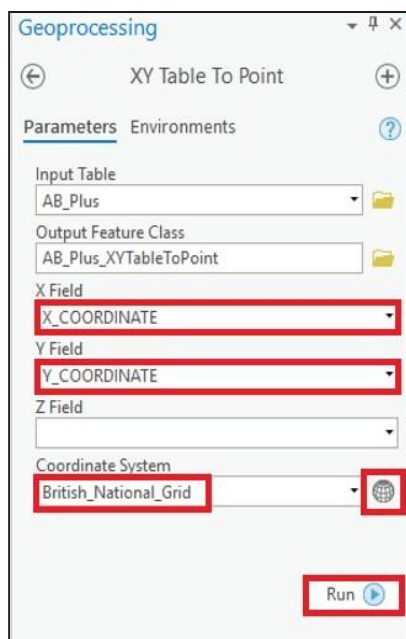



The data has loaded as a non-geometry table.

11. To make the data visible against the mapping backdrop, the XY Coordinate fields need to be specified.
- a. In the Contents pane, right-click the *AB\_Plus* table (or the output name you chose) and in the dropdown, click *Display XY data*.



12. In the Geoprocessing window, the *XY Table To Point* parameters will be displayed.

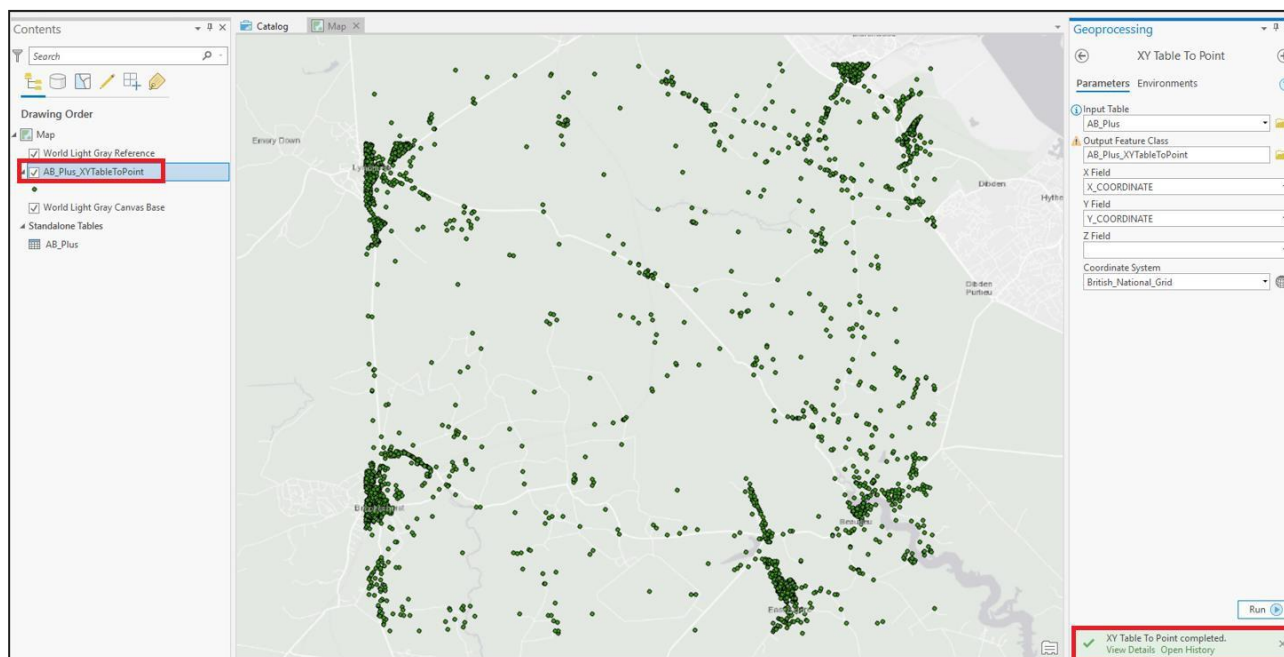


- a. Using the dropdown options, change the X Field to *X\_COORDINATE* or *Longitude* and the Y Field to *Y\_COORDINATE* or *Latitude*.
- b. Change the Coordinate System to *British\_National\_Grid* by clicking the globe icon .
- c. Then select *Projected Co-ordinate Systems > National Grids > Europe > British National Grid*.

Note – If you selected X and Y as *Longitude* and *Latitude* in the step above, then you need to select *ETRS89 [EPSG: 4258]* instead.

13. Click *Run*.

14. Once the process has run, a green box will appear at the bottom of the Geoprocessing window and the output *XYTableToPoint* map layer should appear ticked on the left-hand Contents pane. In the Map window, the addresses will now be displayed as point features.



You have now successfully loaded the data in ArcGIS Pro.

## 5.2 Loading CSV into ArcGIS Desktop

These instructions are based on ArcGIS Desktop versions 9.3 and 10.

When using CSV data in ArcGIS, it is necessary to have column headings. Please ensure that headings have already been prepared as described in [Section 4](#).

1. Launch ArcCatalog as a separate program, or within ArcMap if you are using version 10.
2. Connect to a folder where the AddressBase data you wish to use can be accessed, for example: `C:\AddressBase_Data` or `C:\AddressBase_Plus_Data`. To do this:
  - a. Click *File*, or select *Folder Connections* if you are using version 10.
  - b. Click *Connect Folder*, or in version 10, right-click on *Folder connections* > *Connect Folder* and navigate to the relevant folder.
  - c. From the main window, select the folder to connect to and click *OK*.
3. The folder should now appear in the navigation window to the left of the screen, or within your Catalog window if you have opened it within ArcGIS Map.
4. Create a File Geodatabase to store the address data. Using the file tree, go to folder connections and navigate to the directory where you wish to create the File Geodatabase, for example: `C:\AddressBase_Geodatabase\AddressBase_Plus`. This may need to be set up as a new connection as per the above.

5. Right-click on the folder where you the File Geodatabase to be contained, then select *New* and *File Geodatabase*.
6. A File Geodatabase will be created and named by default as *New File Geodatabase*. Rename the File Geodatabase to a name of your choice.
7. Right-click on your new File Geodatabase, and select *Import > Table (single)...*
  - a. For *Input Rows*, navigate to the location of the CSV data file that contains the merged header and AddressBase or AddressBase Plus data file.
  - b. The *Output Location* should automatically populate with the location of the File Geodatabase that is to be updated; this should be the File Geodatabase you created above.
  - c. Insert a relevant name for the *Output Table*, for example: *AddressBase\_data*. Ensure that there are no spaces in the table name. This name will appear under your geodatabase.
8. Click *OK*.
9. To create a map of the locations of the AddressBase records, they need to be geocoded.
  - a. Right-click on the AddressBase table in the geodatabase that you have just created and select *Create Feature Class*.
  - b. In the *XY Table...* window, you can use the dropdowns to change the X Field to either *X\_COORDINATE* or *Longitude*, and the Y Field to *Y\_COORDINATE* or *Latitude*.
  - c. Click on the *Input Coordinates* icon and navigate to *Projected Co-ordinate Systems > National Grids > Europe > British National Grid*.

Note – If you selected X and Y as *Longitude* and *Latitude* in the step above, then you need to select *ETRS89 [EPSG: 4258]* instead.

- d. Double-click on the chosen *Coordinate System*, then click *Apply* and *OK*.
- e. Click on the folder icon alongside the *Output* field and navigate to the File Geodatabase you just created above. If you cannot see the File Geodatabase, ensure that the *Save as type* box at the bottom of the dialog box is set to *File and Personal Geodatabase feature classes*.
- f. Type in a name for it and click *Save*.
- g. Leave the *Configuration* keyword dropdown menu as *DEFAULTS*. Click *OK*.

Note – You may need to right-click on the Personal Geodatabase where it was saved and select *Refresh* in order to see your points.

At this stage, if you have completed the steps above in ArcCatalog and not within ArcMap, please continue to follow the steps below. Otherwise, if you have been using version 10 with the catalog inside ArcMap, the data can now be loaded into ArcMap.

10. In ArcMap, select *File > Add Data* and navigate to the folder where the File Geodatabase was created above.
11. Double-click on the File Geodatabase to open it, then select all the files inside.

12. Click *Add*.
13. Once the data has been loaded into ArcMap, you may wish to display more than the ESRI-defined Object ID in the Info tool. To change this:
  - a. Double-click on the spatial dataset.
  - b. Select the *Fields* tab.
  - c. Change the *Primary Display Field* to your desired field, for example, *UPRN*.

## 5.3 Loading CSV into MapInfo Pro

These instructions are based on MapInfo Pro version 12.

Note – MapInfo has a size limit of 2Gb on each table. This equates to a maximum number of approximately 4 million AddressBase records.

When using CSV data in MapInfo, it is not a critical requirement to have column headings. However, for ease of use we recommend using the headings supplied by Ordnance Survey. Instructions on how to merge the data and append the header files can be found in [Section 4](#) of this guide.

1. Launch *MapInfo*.
2. Cancel the Quick Start prompt.
3. Click *File > Open* and navigate to the folder that contains the AddressBase data.
4. In the *Files of Type* dropdown menu, select *Comma delimited CSV (\*.csv)*, then click on the AddressBase data to be loaded. Click *Open*.
5. In the next window, tick the box *Use First Line for Column Titles* and select the character set of *INSERT CHARACTER SET*. Click *OK*.

Note – When adding data this way, the field type classifications and field sizes of each column automatically try to fit the type of data that MapInfo believes is contained within the column and the largest value of that classification found within that column. This means that the classifications and field sizes of some attributes may not match the field types and sizes stated in the Technical Specification. The following instructions outline how to change these columns to match those values:

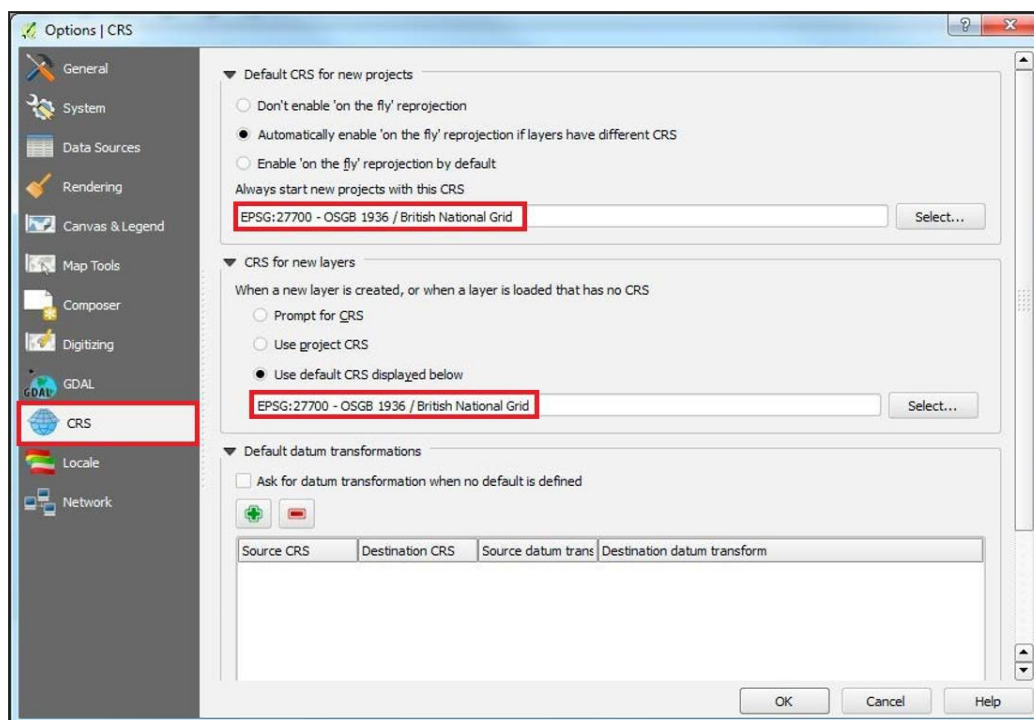
6. Select *File > Save Copy As...* and select the AddressBase table that was loaded. Select *Save As...* and name the table to be created, then click *Save*.
7. Open the table that was just created via *File > Open*. Navigate to and select the copy of the table you just named. Click *Open*.
8. Navigate to *Table > Maintenance > Table Structure* and select the table to be edited. Click *OK*.

9. Here you can change the *Type* and *Width* of each attribute to match the ones stated in the Technical Specification. The Technical Specification can be found on your respective Product Support page:
  - AddressBase:  
<https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-support>
  - AddressBase Plus:  
<https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-plus-support>
  - AddressBase Plus Islands:  
<https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-islands-support>
10. *Type* and *Width* should be changed for all attributes, apart from the following, due to software-specific dependencies:
  - *UPRN*, which should be classified as *Float*.
  - All attributes that have a Field Type of *Date* in the Technical Specification, which should be classified as *Character* with a length of 10.
11. After all changes have been made, click *OK*.
12. To create a map of the location of the AddressBase records, they need to be geocoded:
  - a. Ensure the table of AddressBase records that you wish to geocode is open, then navigate to *Table > Create Points*.
  - b. Select the table you wish to geocode from the *Create Points for Table* dropdown menu.
  - c. Expand the *Get X Coordinates from Column* dropdown menu and select either *X\_Coordinate* or *Longitude*.
  - d. Expand the *Get Y Coordinates from Column* dropdown menu and select either *Y\_Coordinate* or *Latitude*.
  - e. Click on the *Projection* icon, then select the *British Coordinate Systems* option from the *Category* dropdown menu. Select the *British National Grid [EPSG: 27700]*, or if you selected *Longitude* and *Latitude* in the steps above, select *ETRS89 [EPSG: 4258]*.
  - f. Click *OK* to close that window and *OK* again to close the next window.
  - g. Finally, click *Window > New Map Window* to view the loaded geocoded points.

## 5.4 Loading CSV into QGIS

These instructions are based on QGIS version 2.6.

1. Launch QGIS and click *Settings > Options*.
2. Select *CRS* from the left-hand menu and check that the Coordinate Reference System is set to *British National Grid*.



Check this is set for both *Default CRS for new projects* and the *CRS for new layers* sections. If these are not already set, click *Select* at the end of each section and type *27700* into the Filter Box to find and select *British National Grid*. Alternatively, if you intend to use Latitude and Longitude columns, select *ETRS89 [EPSG: 4258]*.

3. Click *OK*.
4. Back in the QGIS window, go to *Layer* and select *Add Delimited Text Layer*.
5. Click *Browse* next to the filename and locate the CSV file that was created in [Section 4](#), containing the merged header files and AddressBase data.
6. Select the CSV file and click *Open*.
7. Accept the default or create a new layer name for the dataset.
8. Ensure that the *First record has field names* is ticked.
9. For *Field Options*, select *Decimal separator is comma*.
10. For *Geometry Definition*, select *Point Coordinates*.
11. You should now be able to select the *X\_Coordinate field* for the *X Field* dropdown and the *Y\_Coordinate field* for the *Y Field* dropdown if this was not done automatically. Alternatively, if you wish to use the *Latitude and Longitude columns*, the *Longitude column* needs to be inserted into the *X\_COORDINATE field*, and the *Latitude column* needs to be inserted into the *Y\_COORDINATE field*.
12. Click *OK*.

## 6. Loading CSV into a database

This section describes how to load AddressBase products into some common databases.

### 6.1 Considerations

#### 6.1.1 Software dependencies

It should be noted that ArcMap, ArcGIS Desktop and ArcGIS Server software do not support the BIGINT/NUMBER data type as an Object ID. Bear this in mind if the expectation is to use this data type directly with these ESRI products. An alternative method to facilitate using ESRI software is to store this data as a string and add a new Serial ID to act as the Object ID.

If you are loading AddressBase data directly into a database, you may need to increase the column length to accommodate language characters such as ^. Some databases treat this as an additional character and therefore, if you define the column length according to our specification, there is a chance that the load may fail. Please bear in mind such adjustments may be required depending on the database you use to load the data.

#### 6.1.2 UPRN deletions

It is important to note that if a UPRN is deleted and then reinserted, then this does not compromise the integrity of the UPRN and its use as a primary key. If a delete is issued for a UPRN, this does not mean it will not reappear in subsequent supplies. These are the reasons why this may happen:

1. The record has moved in location more than once, moving it out of your Area of Interest (AOI), hence the deletion, but then moved back into your AOI in the future. This would also occur if you altered your AOI.
2. A record has failed data validation upon a change being made. This can result, dependent on the change being made, in the record being deleted and then reintroduced when the error is fixed by the data supplier.

If a UPRN is deleted, it will not be reallocated to a different property and it therefore remains the unique identifier for a property.

## 6.2 Loading CSV into a PostgreSQL database

These steps describe how to load AddressBase into a PostgreSQL database using the text files created by following the instructions in [Section 4](#) to merge the CSV files.

These instructions are based on PostgreSQL version 11.2.3 and assume that you have set-up your database with the PostGIS spatial extension. It is recommended that you have basic understanding of database terminology before following this guide.



1. Prepare the text files as described in [Section 4](#) of this guide.
2. Check that there are no carriage returns (extra rows) at the end of the CSV output file as this will result in errors. To check this, open the CSV file and hit *End* on your keyboard. Your cursor should now be at the end of the last line, and not on any extra line below. If it is on the line below, hit *Delete* to remove the extra empty row.
3. Open the *PGAdmin* tool (this can be found in *Windows Start Menu > PostGreSQL*).
4. Either connect to an existing database or create a new database. It is recommended that the encoding is set to *UTF-8*.
5. Open the public schema (although in a production environment, it is advised to use a different schema) and create the tables using the following steps:
  - a. Open the SQL query tool.
  - b. Depending on the data to be loaded, download the SQL file from either the AddressBase or AddressBase\_Plus\_and\_Island folder on:  
[https://github.com/OrdnanceSurvey/AddressBase/tree/master/Loading\\_Scripts/PostgreSQL](https://github.com/OrdnanceSurvey/AddressBase/tree/master/Loading_Scripts/PostgreSQL).
  - c. This SQL file can be opened in a text editor, and the SQL scripts within it can be copied and pasted into the SQL query tool within PostGreSQL.
6. Once the table has been created, the data can be loaded into each table using the SQL COPY. Adding the CSV option as the first line contains a header record for each table.

Please note that the examples below are for AddressBase, then AddressBase Plus and AddressBase Plus Islands, respectively. The path and filename may need to be changed to reflect your data set-up:


```
COPY addressbase FROM 'C:/Address/AddressBase.csv' DELIMITER ',' CSV HEADER;
```

```
COPY addressbase_plus FROM 'C:/Address/AddressBase_Plus.csv' DELIMITER ',' CSV HEADER;
```

```
COPY addressbase_plus_islands FROM 'C:/Address/AddressBase_Plus_Islands.csv' DELIMITER ',' CSV HEADER;
```

7. Once loaded, you may want to add Primary Keys to the data. However, these can only be added on columns where the data values are unique. Where there are no unique data values, an index may be added which will aid searching. The UPRN provides the only unique value in AddressBase and AddressBase Plus.

Primary Keys are added using the following steps:

- a. Right-click on the table name and select *Properties*.
- b. Select the *Constraints* tab.
- c. Click the + to add a new primary key.
- d. Click the *edit icon* .
- e. Enter a name to call the key under the *general tab* (for example, Key1).

- f. Under the definition tab, select *UPRN* or any other unique value from the dropdown under columns.
  - g. Click *Save*.
8. You can also index the data by following these steps:
- a. Right-click on the table name and select *Create > Index*.
  - b. Under *general*, enter a name (for example, *Idx1*).
  - c. Under the *definition tab > Columns*, click the +.
  - d. Select the *UPRN* for example, or any other unique value.
  - e. Click *Save*.

### 6.2.1 Converting coordinates to geometry

A PostGIS extension is required to create geometries. The AddressBase products contain both British National Grid (BNG) and ETRS89 coordinates. The SQL below shows how to create a column for BNG, but it can be altered to utilise the ETRS89 data.

1. Add a geometry column called *geom* to make the data usable in a GIS:

```
SELECT AddGeometryColumn ('public', 'addressbase_plus', 'geom', 27700, 'POINT', 2);
```

2. Load the data into your new geometry column:

```
UPDATE public.addressbase_plus SET geom = ST_GeomFromText('POINT(' || x_coordinate || ' ' ||  
y_coordinate || ')', 27700);
```

This sets the *geom* column in the table to equal the values from the *X\_coordinate* and *Y\_coordinate* columns, with the spatial reference defined as 27700.

3. Create a spatial index on the data using:

```
CREATE INDEX idx_abp_geom ON public.addressbase_plus USING gist(geom);
```

This adds the index name *idx\_abp\_geom* to the same table on the *geom* column.

## 6.3 Loading CSV into an Oracle database

These instructions assume a basic knowledge of Oracle databases and SQLLDR (the package used to load the CSV files into the database). Other options are available for loading data into Oracle databases.

Using SQLLDR it is not necessary to merge all the AddressBase files into a single file, but it can load the data directly from the file provided as long as it has been unzipped first.

The following steps describe one method for loading a full supply of the data. Sections in *italics* denote where changes will need to be made to accommodate local file naming.

1. Copy the data files from the disk to an appropriate location. It is worth noting that the files will need to be unzipped and therefore you will need in the region of 43Gb of free space.
2. Once the data is copied, the next stage is to unzip the \*.zip files to \*.csv. This can be done using a package such as Winzip or 7Zip. Please see [Section 3](#) for more information.
3. With all the files unzipped, the latter stages are easier if you create a list of all the CSV files to be loaded. This can be done using a batch file that writes all the files out to a text file:

```
dir *.csv /b/s >filelisting.txt pause
```

This file will form the basis for loading the control file in a later step.

4. Go to the OS GitHub repository:  
[https://github.com/OrdnanceSurvey/AddressBase/tree/master/Loading\\_Scripts/Oracle](https://github.com/OrdnanceSurvey/AddressBase/tree/master/Loading_Scripts/Oracle).
5. Open the folder of your chosen product and you should see three files. Open the file ending *createtable.sql* in a text editor.
6. Within the provided SQL there are references to *<TablespaceName>*, which need to be changed to the tablespace that is being worked in. When these are changed, copy and paste the SQL into Oracle to create the tables.
7. Next, create a SQLLDR control file. An example of one of these files is *Oracle AddressBase\_Control.ctl*, which is provided in the folder of the GitHub repository in Step 4 above. Open the SQLLDR control file for your chosen product in a text editor.
8. Within the file you will see lines referencing *INFILE*. Populate these *INFILE* lines with the file listing created in Step 3, with one *INFLE* command for each file. This tells the process to open each of the files and carry out the other tasks listed below it.

Note – The last section of the control file creates the Geometry for the X and Y coordinate (British National Grid) if you want to create a Geometry for the Latitude and Longitude values, this will need to be created separately.

9. Once this file is created, it can be called from a .bat file to run it on the box that holds the database rather than a remote machine. If you wish to run it from a remote machine, contact your Oracle Administrator who will be able to advise on the best way to do this within your environment. The contents of the .bat file should be similar to the following:

```
@sqlldr <username>/<password>@<service name> control= <name of ctl file created previously> Pause
```

10. Once the load has completed the relevant indexes need to be built. The SQL statements to create the indexes can be found in the same GitHub repository linked in Step 4 above.

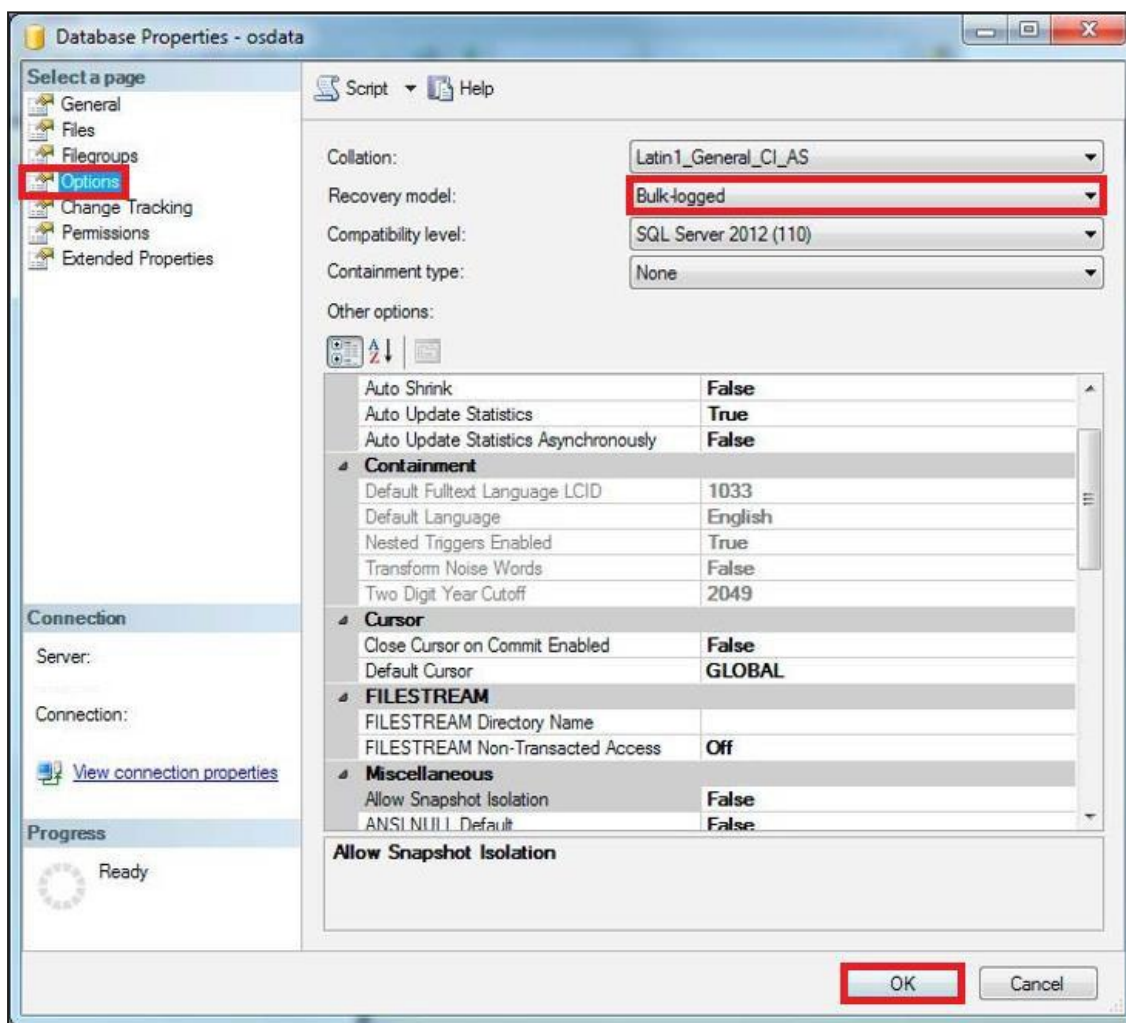
As before, you can copy and paste the SQL statements from a text editor into Oracle to create the indexes. The example table name provided may be different to yours, so check if this needs to be changed before use.

## 6.4 Loading CSV into Microsoft SQL Server

The following instructions assume that users have basic knowledge of Microsoft SQL Server and that the CSV data is already prepared as described in [Section 4](#) of this guide.

Note – There are many ways to load AddressBase products into Microsoft SQL Server; this is just one suggested method for guidance.

1. Open the *SQL Server Management Studio (SSMS)*.
2. Right-click on the database you are loading into and select *Properties*.
3. Select *Options* on the left-hand side.
4. Expand the dropdown box for *Recovery Model* and select *Bulk-logged*. This minimises the logfile size, otherwise the default logging for Microsoft SQL Server can cause logfiles to grow over 20Gb and this can cause issues with loading.



5. Open the *SQL Server Management Studio (SSMS)* and right-click your database from the left-hand panel.
6. Navigate to *Tasks* and click *Import Data*. This will open the *SQL Server Import and Export Wizard*.

7. Click *Next*.
8. On the next screen, change your *Data Source* to *Flat File Source*.
9. Use the *Browse* button to navigate to your CSV file and select it. If you cannot see your files, ensure that the bottom right dropdown box has CSV files (\*.csv) selected.
10. Click *Open*.
11. Your CSV file should have a header row already prepared from [Section 4](#) of this guide. Ensure the *Column names in the first data row* is ticked.
12. Check that the *Text Qualifier* is set to a double quote (“). This is to make sure that the quotations in the raw data supply are removed upon loading but that the data remains intact.
13. On the left-hand side of this screen, select *Columns* and check that the *Column delimiter* is set to *Comma*.
14. On the left-hand side of the screen, select *Advanced*.
15. For each column of data you are loading, you will need to specify a *DataType*. The Microsoft SQL Server loader defaults each column to a *String*. The correct Data Types for each column are given in the Technical Specification, which can be found on the respective Product Support page:
  - AddressBase:  
<https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-support>
  - AddressBase Plus:  
<https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-plus-support>
  - AddressBase Plus Islands:  
<https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-islands-support>
16. Once you have changed the Data Types for each column to match those given in the Technical Specification, click *Next*.
17. Check that your table is going to be imported into the correct database and click *Next*.
18. On this screen, you can edit the default table name that Microsoft SQL Server has chosen by clicking in the destination box. For example, for AddressBase Plus renaming to *[dbo].[ADDRESSBASE\_PLUS]*.
19. Select *Edit Mappings* in the bottom right-hand corner.
20. In the new window, you must remove the tick in the checkbox against the *UPRN column*, which needs to be the Primary Key of the table. Click *OK* once the Primary Key alterations have been completed.
21. Click *Next*. On this screen, you can check that the Source column and Destination columns are correct.
22. Click *Next*. A summary of your import will appear. If you want to continue, click *Finish*.
23. A report will be generated as your data is imported. *Success* should appear at the top once complete.
24. You may need to right-click on your database and click *Refresh* to see your new table listed.

### 6.4.1 Setting Primary Keys

To create a Primary Key field, you can run an SQL statement, such as this example for AddressBase Plus below. Note that the columns you are creating these constraints on cannot be null or allowed to be null.

```
alter table dbo.ADDRESSBASE_PLUS  
add primary key ([UPRN]);
```

### 6.4.2 Creating the point geometry

You can also create point geometry using the X and Y coordinates or the Latitude and Longitude coordinate values. This is achieved by running the following SQL statement:

```
alter table dbo.ADDRESSBASE_PLUS  
add geometry_column as geometry::Point([X_Coordinate],[Y_Coordinate], 27700);
```

Note – This is using British National Grid coordinates, with 27700 representing the spatial reference of the data. To use the Latitude and Longitude coordinate, the spatial reference should be set to 4258 for ETRS89.

## 7. Loading GML

GML is an XML dialect which can be used to model geographic features. It was designed by the Open Geospatial Consortium (OGC) as a means for people to share information regardless of the applications or technology that they use.

In the first instance, GML was used to overcome the differences between different GIS applications by providing a neutral file format as an alternative to proprietary formats. Because it is independent of applications, it can also be moved between databases or other types of application, which allows a wider application than just GIS data transfer.

GML data can be viewed and loaded into a database using software such as Safe FME:

[https://docs.safe.com/fme/html/FME\\_Desktop\\_Documentation/FME\\_ReadersWriters/gml/gml.htm](https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_ReadersWriters/gml/gml.htm)

## 8. COU implementation guidance

### 8.1 Introduction

All the AddressBase products are available as a full supply or a COU. A COU means you will only be supplied with the features which have changed since your last supply. The following sections provide guidance on how you could potentially manage a COU supply of AddressBase and AddressBase Plus data.

Please note: If you receive a tile supply, you will receive Change Chunks. This means if a record within your tile has changed, all of the records in that tile will be provided to you as inserts, and no updates or deletes will be issued.

Tiles are only available for GB supplies, so this does not apply to AddressBase Plus Islands.

### 8.2 Types of change

At a high-level, there are three types of change found within a COU:

- Deletes (CHANGE\_TYPE 'D') are objects that have ceased to exist in your AOI since the last product refresh.
- Inserts (CHANGE\_TYPE 'I') are objects that have been newly inserted into your AOI since the last product refresh.
- Updates (CHANGE\_TYPE 'U') are objects that have been updated in your AOI since the last product refresh.

### 8.3 High-level COU implementation model

The diagram below shows how to implement an AddressBase, AddressBase Plus and AddressBase Plus Islands COU within a database.

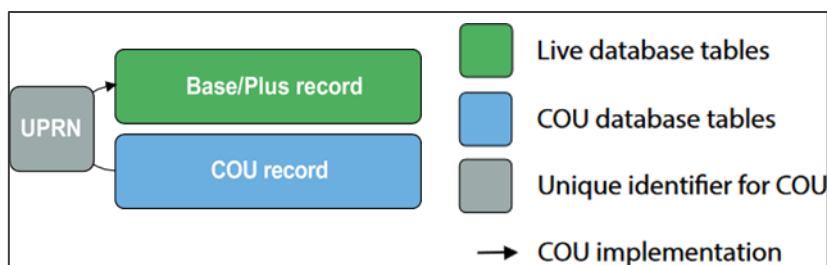


Figure 1: High-level COU implementation model.

## 8.4 High-level COU implementation model – with archiving

Before a COU is applied, there may be a business requirement to archive existing address records. The table below shows how to implement archiving with an AddressBase COU within a database.

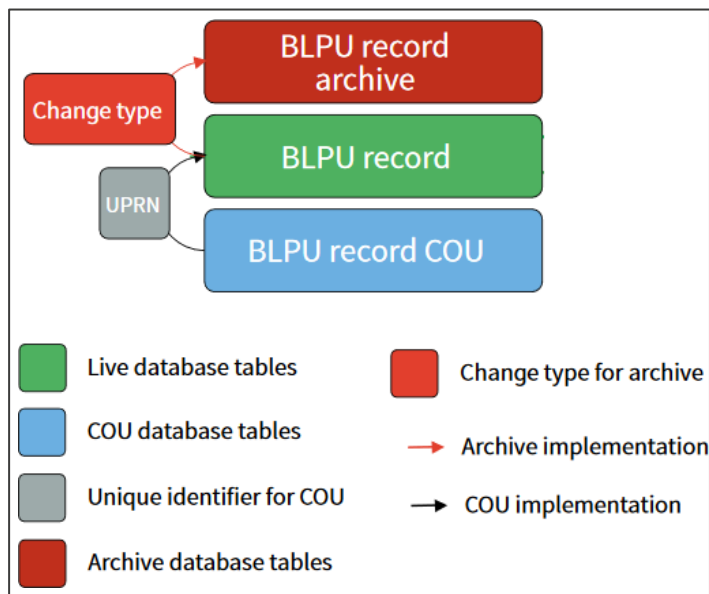


Figure 2: High-level COU implementation model on how to create archive tables and apply a COU.

## 8.5 Applying COU to tables

Within AddressBase and AddressBase Plus there will be no records with the same UPRN. This can be tested by checking the number of records that have the same UPRN. The following SQL code would notify you of any duplicates:

```
SELECT uprn, COUNT(uprn) AS NumOccurrences
FROM addressbase_plus
GROUP BY uprn
HAVING ( COUNT(uprn) > 1 );
```

This query should return 0 rows, and this confirms that there are no duplicates.

As there are no duplicate records, we can use the UPRN to apply the COU.

Once confirmed, the following steps can be taken to apply the COU (without archiving):

#Initially delete the existing records that will be updated and deleted:

```
DELETE FROM addressbaseplus WHERE uprn IN (SELECT uprn FROM addressbaseplus_cou
WHERE change_type != 'I');
```



#Insert the new updated records and the new inserted records:

```
INSERT INTO addressbaseplus SELECT * FROM addressbaseplus _cou  
WHERE change_type != 'D';
```

Where there is a business requirement to keep the records that are being Updated and Deleted in a separate archive table, the following SQL will create an Archive Table. It will populate with records that are being Updated and Deleted from the live AddressBase or AddressBase Plus table.

#The following command creates an archive table of the records that are being updated and deleted from the existing table.

#If this table already exists, you can simply use INSERT INTO rather than CREATE TABLE.

```
CREATE TABLE addressbaseplus_archive AS SELECT * FROM addressbaseplus  
WHERE uprn IN (SELECT uprn FROM addressbaseplus _cou WHERE change_type != 'I');
```

#The following command then deletes the records from the existing table, which are either updates or deletions:

```
DELETE FROM addressbaseplus  
WHERE uprn IN (SELECT uprn FROM addressbaseplus _cou WHERE change_type!= 'I');
```

#The following command then inserts the new insert records and the new updated records into the live table:

```
INSERT INTO addressbaseplus SELECT * FROM addressbaseplus _cou  
WHERE change_type != 'D';
```

## 9. Creating a Single-Line or Multi-Line Address

### 9.1 Introduction

The AddressBase products provide a variety of data fields, allowing you to construct different forms of an address for a given addressable object, dependent on how the address is to be used.

AddressBase contains the Delivery Point Address which is sourced from Royal Mail's Postcode Address File (PAF) – a non-geocoded list of addresses. These addresses are used primarily as a 'mailing list' for postal purposes.

There are two types of address contained in the AddressBase Plus products:

- Delivery Point Address
- Geographic Address

These two address types come from different sources and are matched together by GeoPlace.

As noted above, the Delivery Point Address is sourced from Royal Mail's PAF data. Geographic Addresses are maintained by contributing Local Authorities. The structure of a Geographic Address is based on the British Standard BS7666. These addresses are used to provide an accurate geographic locator for an object to support, for example, service delivery, asset management, or command and control operations. They also represent the legal form of addresses as created under street naming and numbering legislation.

Each UPRN in AddressBase Plus provides the Geographic Address and, where matched, the Delivery Point Address in a one-to-one relationship. If there is no match, then the following fields will be left empty:

DEPARTMENT\_NAME

ORGANISATION\_NAME

SUB\_BUILDING\_NAME

BUILDING\_NAME

BUILDING\_NUMBER

PO\_BOX\_NUMBER

DEPENDENT\_THOROUGHFARE (and WELSH\_DEPENDENT\_THOROUGHFARE)

THOROUGHFARE (and WELSH\_THOROUGHFARE)

DOUBLE\_DEPENDENT\_LOCALITY (and WELSH\_DOUBLE\_DEPENDENT\_LOCALITY)

DEPENDENT\_LOCALITY (and WELSH\_DEPENDENT\_LOCALITY)

POST\_TOWN (and WELSH\_POST\_TOWN)

POSTCODE

## 9.2 Background

A common requirement for customers using the AddressBase products is to build a single address label from core address elements.

There are two types of address label. The simplest is a full address on a single line with different elements separated by commas and spaces. This type of label is suited for displaying a full address within a tabular display, such as within an on-screen data grid or spreadsheet, or where a single-line printed address is most appropriate (such as within the text, header or footer of a letter):

ROSE COTTAGE, 5 MAIN STREET, ADDRESSVILLE, LONDON, SE99 9EX

The other type of formatted address is a multi-line address label. These are most often used on envelopes or at the tops of letters, where different parts of an address are separated onto different lines:

ROSE COTTAGE  
5 MAIN STREET  
ADDRESSVILLE  
LONDON  
SE99 9EX

This guide outlines a methodology for structuring and layering a single address label, providing suggested logic to build both the Delivery Point and Geographic Address. The logic in [Section 9.3](#) is applicable to both AddressBase, AddressBase Plus and AddressBase Plus Islands. The logic in [Section 9.4](#) is only applicable to AddressBase Plus and AddressBase Plus Islands.

The rules in this guide are suggestions only and can be used for visual display of full addresses. It is strongly recommended that address components are stored in the format in which they are provided in order to allow maximum flexibility of use and derived value.

## 9.3 Delivery Point Address (PAF Address)

A Delivery Point Address contains information sourced from Royal Mail (PAF). Stringent rules are used to match these addresses to the Geographic Address and assign a common UPRN to link addresses from the two addressing sources together in the data model.

To construct a single address label based purely on the Royal Mail PAF address fields, the following attributes can be used to build a Delivery Point Address label.

Delivery Point Address Component	Type
DEPARTMENT_NAME	Character
ORGANISATION_NAME	Character
SUB_BUILDING_NAME	Character

Table 1: Delivery Point Address Components.

Delivery Point Address Component	Type
BUILDING_NAME	Character
BUILDING_NUMBER	Integer
PO_BOX_NUMBER	Integer
DEPENDENT_THOROUGHFARE (or WELSH_DEPENDENT_THOROUGHFARE)	Character
THOROUGHFARE (or WELSH_THOROUGHFARE)	Character
DOUBLE_DEPENDENT_LOCALITY (or WELSH_DOUBLE_DEPENDENT_LOCALITY)	Character
DEPENDENT_LOCALITY (or WELSH_DEPENDENT_LOCALITY)	Character
POST_TOWN (or WELSH_POST_TOWN)	Character
POSTCODE	Character

These address components are listed in the correct order in which they should appear on an address label. There may be a business need to replace the thoroughfare, locality and post\_town attributes with the Welsh equivalent. The following examples will use the English version of these attributes only.

It should be noted that most of the PAF fields are optional and may contain null values (or zero, in the cases of 'BUILDING NUMBER' and 'PO BOX NUMBER'). In these cases, those fields should be omitted.

The following (entirely fictional) example shows all of the PAF fields filled in (apart from the PO Box number) and indicates how they should be ordered in a single address label.

Delivery Point Address Component	Example
DEPARTMENT_NAME ORGANISATION_NAME	CUSTOMER SERVICE DEPARTMENT JW SIMPSON LTD.
SUB_BUILDING_NAME BUILDING_NAME BUILDING_NUMBER PO_BOX_NUMBER	UNIT 3 THE OLD FORGE 7
DEPENDENT_THOROUGHFARE THOROUGHFARE DOUBLE_DEPENDENT_LOCALITY DEPENDENT_LOCALITY POST_TOWN POSTCODE	RICHMOND TERRACE MAIN STREET HOOK WARSASH SOUTHAMPTON SO99 9ZZ

In cases where a PO BOX NUMBER is present, it will only be described in the data as an integer. In order to properly format these addresses when generating an address label, these integers should be prefixed with the text 'PO BOX', as shown in the following example:

Delivery Point Address Component	Data Content	Formatted output
ORGANISATION_NAME	'JWS CONSULTING'	JWS CONSULTING
PO_BOX_NUMBER	5422	PO BOX 5422
THOROUGHFARE	'HIGH STREET'	HIGH STREET
POST_TOWN	'SPRINGFIELD'	SPRINGFIELD
POSTCODE	'SP77 0SF'	SP77 0SF

Where null or empty, string values exist (for character fields) or zeros or nulls (for integer fields), those fields should be entirely omitted from the output. However, the order in which the fields should be concatenated always remains the same.

Delivery Point Address Component	Data content	Formatted output
DEPARTMENT_NAME	Null	TM MOTORS
ORGANISATION_NAME	'TM MOTORS'	
SUB_BUILDING_NAME	Null	
BUILDING_NAME	'THE OLD BARN'	
BUILDING_NUMBER	0 (or null)	
PO_BOX_NUMBER	0 (or null)	THE OLD BARN
DEPENDENT_THOROUGHFARE	Null	
THOROUGHFARE	'HORSHAM LANE'	
DOUBLE_DEPENDENT_LOCALITY	Null	
DEPENDENT_LOCALITY	Null	
POST_TOWN	'HORSHAM'	HORSHAM
POSTCODE	'RH12 1EQ'	RH12 1EQ

### 9.3.1 Building a single-line Delivery Point Address

Building a single-line, formatted address for a Delivery Point is relatively straightforward. All the fields should be checked in the order shown previously in Table 1, and those that have values should be concatenated together into a single line. Generally, address components should be separated by a comma followed by a single space (', '), although sometimes only a space is used between a building number and a thoroughfare name. You can use your preference.

An example of SQL logic to create a single-line Delivery Point Address is on our GitHub repository <https://github.com/OrdnanceSurvey/AddressBase/tree/master/SQL>, which should be used under the following considerations:

- The SQL operator for concatenating text is a double pipe ('||').
- CASE blocks have been used to test each of the fields for null values before concatenating its contents (along with a suitable separator – either ', ' or ' ').
- The field names and table names used are illustrative and may vary between databases.
- Depending on the database schema and data loading method used, it may be necessary to test some fields for empty strings ('') or zero values (for integer fields) instead of, or as well as, testing for NULLs.

- If you are using PostGres (PostGIS), it might be beneficial to substitute the 'IS NOT NULL' with != ". This should improve the overall appearance of the output.

### 9.3.2 Building a multi-line Delivery Point Address

Splitting a Delivery Point Address into multiple lines is more complicated. There are several rules to consider in order to avoid having very short lines (for example, just a building number) or very long lines within the formatted address. A summary of these rules is as follows:

1. Generally, if there is a building number, it should appear on the same line as the thoroughfare (or dependent thoroughfare) name. If there is no thoroughfare name information, it should appear on the same line as the first locality name.
2. In cases where building numbers have been placed in the building name field due to the presence of a letter suffix (for example, '11A') or a number range separator (for example, '3-5'), these should be detected and placed on the same line as the thoroughfare name in the same way as a building number (or on the first locality line if no thoroughfare name is present).
3. In most other cases, the building name, if present, should appear on a separate line above the thoroughfare (or dependent thoroughfare) name. If there is no thoroughfare name present, it should appear on the same line as the first locality name.
4. Similar tests should be applied to the SUB\_BUILDING\_NAME field: if this field contains a number, a number with a suffix, or a numeric range, it should precede the building name on the same line. In most other cases, it should appear on a separate line above the building name.

For more information on how to build a multi-line Delivery Point Address, please consult pages 27–42 of the [PAF Programmers' Guide](#).

## 9.4 Geographic Address (Local Authority Address)

The structure of a Geographic Address is based on the British Standard BS7666 and is split into several components. This means that in order to construct a complete address label (for example, on an envelope, database form or GIS display), the components need to be constructed according to a set of rules.

Within the AddressBase Plus products, the core property level address information is stored within the Primary Addressable Object (PAO) and Secondary Addressable Object (SAO) fields. The additional attribution required to build a full address label are the la\_organisation, street\_description, locality, town\_name, administrative\_area and postcode\_locator.

For a full description of PAOs and SAOs, and the complete set of AddressBase Plus fields, please refer to the Technical Specification on your respective Product Support page:

- AddressBase:  
<https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-support>
- AddressBase Plus:  
<https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-plus-support>
- AddressBase Plus Islands:  
<https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-islands-support>

## 9.4.1 Constructing a single address label from the Geographic Address fields

To construct a single address label based purely on the BS7666 address fields, the following attributes should be used to build a Geographic Address label.

Table 2: Geographic Address Components.

Geographic Address Component	Type
LA_ORGANISATION	Character
SAO_TEXT (or ALT_LANGUAGE_SAO_TEXT)	Character
SAO_START_NUMBER	Integer
SAO_START_SUFFIX	Character
SAO_END_NUMBER	Integer
SAO_END_SUFFIX	Character
PAO_TEXT (or ALT_LANGUAGE_PAO_TEXT)	Character
PAO_START_NUMBER	Integer
PAO_START_SUFFIX	Character
PAO_END_NUMBER	Integer
PAO_END_SUFFIX	Character
STREET_DESCRIPTION (or ALT_LANGUAGE_STREET_DESCRIPTION)	Character
LOCALITY	Character
TOWN_NAME	Character
ADMINISTRATIVE_AREA*	Character
POSTCODE_LOCATOR	Character

\*ADMINISTRATIVE\_AREA is optional because it is common for this field to be the same as the TOWN\_NAME. Sometimes, however, this field will help users construct a more complete address.

These address components are listed in the correct order in which they should appear on an address label. There may be a business need to use to alternate language fields for the SAO\_TEXT, PAO\_TEXT and STREET\_DESCRIPTION, which are also listed in the correct order above.

## 9.4.2 Rendering SAOs and PAOs

When building a single address label, it may be necessary to concatenate the various SAO fields and PAO fields together respectively. These fields contain any property names, numbers, number ranges or suffixes that apply to an address.

A PAO number/range string should be constructed from the PAO\_START\_NUMBER, PAO\_START\_SUFFIX, PAO\_END\_NUMBER and PAO\_END\_SUFFIX fields, as illustrated in the following table.

Attribute	Example 1	Example 2	Example 3	Example 4
PAO_START_NUMBER	I	I	I	I
PAO_START_SUFFIX		A		A
PAO_END_NUMBER			5	5
PAO_END_SUFFIX				C
Rendered PAO range	I	IA	I-5	IA-5C

Similarly, a SAO number/range string should be constructed from the SAO\_START\_NUMBER, SAO\_START\_SUFFIX, SAO\_END\_NUMBER and SAO\_END\_SUFFIX fields, as illustrated in the following table.

In addition to the numeric range fields described above, there are also PAO\_text and SAO\_text fields. These fields may be populated instead of, or as well as, the numeric range fields. In both cases, if both text and a numeric range string are present, the text should appear before the numeric range in any formatted address.

Attribute	Example 1	Example 2	Example 3	Example 4
PAO (number string)	I	IA	IA	
PAO (text)			Rose Cottage	Rose Cottage
Rendered PAO (showing street name location)	I <street>	IA <street>	Rose Cottage, IA <street>	Rose Cottage, <street>

Note – For PAOs, there will always be either a text entry or a numeric/range entry, or both. This is not the case for SAOs, which may be entirely absent for a given address.



### 9.4.3 Street description, town, locality and administrative area names

The street description and administrative area names are always present, while the locality name and town name may be empty.

The ADMINISTRATIVE\_AREA field always contains a value; however, this value will not always enhance an address, but in some cases it will. In particular, check that it is not the same as the value in the TOWN\_NAME field, as this is often the case.

Administrative area not included	vs.	Administrative area included (BURY)
34, CROW LANE, RAMSBOTTOM, BL0 9BR		34, CROW LANE, RAMSBOTTOM, BURY, BL0 9BR

In other cases, the administrative area name will simply contain the local authority name, which would not traditionally form part of a single or multi-line address but can be included to add additional information to an address label. Its inclusion is largely down to business requirements or personal preference; however, it may also be useful to 'de-duplicate' some Geographic Addresses.

The following (entirely fictional) example shows all of the BS7666 Geographic Address fields filled in and how they should be ordered in a single address label.

Geographic Address Component	Example
LA_ORGANISATION	JW SIMPSON LTD
SAO_TEXT	THE ANNEXE
SAO (number/range string)*	1A
PAO_TEXT	THE OLD MILL
PAO (number/range string)*	7-9
STREET_DESCRIPTION	MAIN STREET
LOCALITY	HOOK
TOWN_NAME	WARSASH
ADMINISTRATIVE_AREA	SOUTHAMPTON
POSTCODE_LOCATOR	SO99 9ZZ

\*The number/range strings are built from the relevant PAO/SAO START\_NUMBER, START\_SUFFIX, END\_NUMBER and END\_SUFFIX fields, as described above, and formatted as character strings.

Where an administrative area matches the town name, it should always be omitted.

Delivery Point Address Component	Data content	Formatted output
PAO_TEXT	'Highbury House'	Highbury House
STREET_DESCRIPTION	'High Street'	High Street
TOWN_NAME	'Southampton'	Southampton

Delivery Point Address Component	Data content	Formatted output
ADMINISTRATIVE_AREA POSTCODE_LOCATOR	'SOUTHAMPTON' 'SO77 0SF'	SO77 0SF

Where null or empty string values exist (for character fields) or zeros or nulls (for integer fields), those fields should be entirely omitted from the output; however, the order in which the fields should be concatenated always remains the same.

Delivery Point Address Component	Data content	Formatted output
ORGANISATION	'TM MOTORS'	TM MOTORS
SAO_TEXT	null	
SAO (number/range string)*	null	
PAO_TEXT	'THE OLD BARN'	THE OLD BARN
PAO (number/range string)*	'1'	1
STREET_DESCRIPTION	'HORSHAM LANE'	HORSHAM LANE
LOCALITY	null	
TOWN_NAME	'HORSHAM'	HORSHAM
ADMINISTRATIVE_AREA	'HORSHAM'	* Duplicate name omitted
POSTCODE_LOCATOR	'RH12 1EQ'	RH12 1EQ

#### 9.4.4 Building a single-line Geographic Address

Building a single-line, formatted address for a Geographic Address is slightly more complicated than for a Delivery Point Address due to the need to preformat the SAO and PAO number/range strings. However, once this is done, the process is largely the same as before: the calculated fields should be checked in the order shown previously in Table 2, and those that have values should be concatenated together into a single line. Generally, address components should be separated by a comma followed by a single space (', '), although sometimes only a space is used between a PAO number/range string and a street description. This is down to personal preference.

Example SQL logic to create a single-line Geographic Address can be found on our GitHub repository <https://github.com/OrdnanceSurvey/AddressBase/tree/master/SQL>, which should be used under the following considerations:

- The SQL operator for concatenating text is a double pipe ('||').
- CASE blocks have been used to test each of the fields for null values before concatenating its contents (along with a suitable separator – either ', ' or ' ').
- The field names and table names used are illustrative and may vary between databases.
- Depending on the database schema and data loading method used, it may be necessary to test some fields for empty strings ('') or zero values (for integer fields) instead of or as well as testing for NULLs.

## 9.4.5 Building a multi-line Geographic Address

Splitting a Geographic Address into multiple lines is more complex. As with Delivery Point Addresses, there are several rules to consider in order to avoid having very short lines (for example, just a building number) or very long lines within the formatted address.

A summary of these rules is as follows:

1. Generally, if there is a PAO number/range string, it should appear on the same line as the Street Description.

### Example

1 1A MAIN STREET

2. If there is a PAO\_text value, it should always appear on the line above the Street Name (or on the line above the <PAO number string> + <Street Name> where there is a PAO number/range).

PAO_text only	PAO_text and PAO number or range
ROSE COTTAGE, MAIN STREET	ROSE COTTAGE, 1 1A MAIN STREET

3. If there is a SAO\_text value, it should appear on a separate line above the PAO\_text line (or the PAO number/range + Street Name where there is no PAO\_text value).

SAO_text value only, with PAO_text value only	SAO_text value only, with PAO number/range only
THE ANNEXE, ROSE COURT, MAIN STREET	THE ANNEXE, 1 1A MAIN STREET

4. If there is a SAO number/range value, it should be inserted either on the same line as the PAO\_text (if there is a PAO\_text value), or on the same line as the PAO number/range + Street Name (if there is only a PAO number/range value and no PAO\_text value). If there are both PAO\_text and a PAO number/range, then the SAO number/range should appear on the same line as the PAO\_text, and the PAO number/range should appear on the street line.

SAO number/range value only, and PAO_text value only	SAO number/range value only, and PAO number/range value only	SAO number/range value only, and both PAO_text and PAO number/range values
1A ROSE COURT, MAIN STREET	1-3, 1 1A MAIN STREET	1A ROSE COURT, 1 1A MAIN STREET

5. If there is a SAO\_text value, it should always appear on its own line.

SAO_text value only with PAO_text only	SAO_text and SAO number/range and PAO_text and PAO number/range
THE ANNEXE, ROSE COTTAGE, MAIN STREET	WARDEN'S FLAT, 1A ROSE COURT, 1 1A MAIN STREET

6. If there is an Organisation Name, it should always appear alone as the top line of the address.

Organisation Name along with all PAO + SAO fields
COTTAGE INDUSTRY LTD, THE ANNEXE, 1A ROSE COURT, 1 1A MAIN STREET

7. The Locality (if present) should appear on a separate line beneath the Street Description, followed by the Town Name on the line below it. If there is no Locality, the Town Name should appear alone on the line beneath the Street Description.

Locality and Town Name present	Town Name only
[first part of address, formatted as described above] MAIN STREET, HIGHFIELD, SOUTHAMPTON	[first part of address, formatted as described above] HIGH STREET, SOUTHAMPTON

8. If the Administrative Area name is required and it is not a duplicate of the Town Name, it can optionally be included on a separate line beneath the Town Name.

Administrative Area name included
[first part of address, formatted as described above] MAIN STREET, WINDSOR, ROYAL BOROUGH OF WINDSOR AND MAIDENHEAD

9. Finally, the Postcode Locator should be inserted on the final line of the address.

#### **With Postcode\_Locator on final line**

[first part of address, formatted as described above]

HIGH STREET,  
MILTON,  
ML99 0VWW

## 9.5 Creating mailing lists

Given that AddressBase Plus contains two different types of address, a decision needs to be made as to whether to use the Geographic or Delivery Point Addresses, or a mixture.

The following two options should be considered:

1. Use Delivery Point Addresses whenever they are available, and when they are not, use a Geographic Address.
2. Use Geographic Addresses in all cases.

Depending on business requirements, in some user interfaces it may be worth considering displaying both forms of an address where possible, since this will provide the maximum information available about a given UPRN.

‘Mixing and matching’ components from the two different forms of address into a single address label is not recommended as this is likely to cause confusion in some instances.

### 9.5.1 Other filters

AddressBase Plus offers other attributes that could be used in conjunction with address labels. For example, classification can be used to target certain types of property, or OS MasterMap Topography TOID cross references can be used to link address labels to Topographic objects and viewed in a GIS.

Note – TOID cross references are not available in AddressBase Plus Islands.

## 10. Searching for addresses

A common requirement for customers using the AddressBase products is to search for properties using full or partial addresses. Address searches may return a large number of addresses, a short list of possibilities, a single match or no results, depending on the search criteria.

There are many methods of implementing an address search, from free text queries through to structured address component searches. This guide will step through two such approaches that may be used when working with AddressBase and/or AddressBase Plus.

Note – These methods are not intended as recommendations; they are merely examples of how to get maximum value out of the product when implementing an address search function.

### 10.1 Free text search

One type of search implementation involves a single ‘search engine’ style text box, into which a user can type all or some of an address. For example:

Find address	Results
CLOVER AVENUE, SW99 9ZZ	1, Clover Avenue, Fieldtown, Addressville, SW99 9ZZ 2, Clover Avenue, Fieldtown, Addressville, SW99 9ZZ 3, Clover Avenue, Fieldtown, Addressville, SW99 9ZZ 4, Clover Avenue, Fieldtown, Addressville, SW99 9ZZ 5, Clover Avenue, Fieldtown, Addressville, SW99 9ZZ 6, Clover Avenue, Fieldtown, Addressville, SW99 9ZZ 7, Clover Avenue, Fieldtown, Addressville, SW99 9ZZ

In this scenario, the user can choose to type anything in *Find address*, which may be just one component of an address (for example, a postcode, street name or building name), several parts of an address (for example, street name + town name, house name + postcode, etc.) or even (rarely) a complete address.

There may or may not be commas between search items, or address components can be entered with or without capitalised letters, etc. In short, with this search method, there is no structure to the user input and the search methodology must be designed with this in mind.

## 10.2 Structured component search

The other common type of implementation for address searches involves entering search criteria in a structured way (for example, with a different text box for each major address component).

Find Address:

House name:	<input type="text" value="Rose Cottage"/>	House number:	<input type="text"/>
Street name:	<input type="text" value="Main Street"/>		
Locality:	<input type="text"/>	Town:	<input type="text"/>
Postcode:	<input type="text"/>		<input type="button" value="Search"/>

### Results

Rose Cottage, Main Street, Fieldtown, Addressville, SW99 9ZZ  
Rose Cottage, Main Street, Ashford, AS45 9PP  
Rose Cottage, Main Street, Buxtew, Monley, MO88 4TY  
And so on...

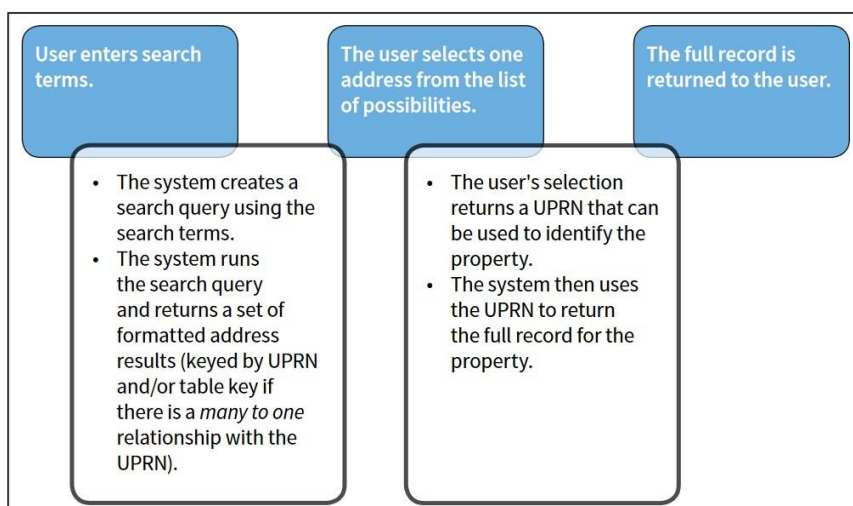
This method guides the user to enter known components of an address and creates a predictable user input structure around which to build a search function. While generally simpler to use and implement, it can be less user-friendly, particularly in cases where it is not obvious which box to type an address component into, for example, is *Richmond Terrace* a building name or a street?

## 10.3 The search operation

This guide suggests how to implement the two search methods described above. Both should be used alongside the instructions on formatting single address labels.

Note – The methods described here may be adapted to work with both AddressBase Plus, AddressBase Plus Islands and AddressBase; however, in the case of AddressBase, only Delivery Point Addresses are searchable, so the geographic guidance will not apply to this product.

An address search operation typically requires two stages of interaction from a user and several processing steps from the underlying IT system. These steps can be summarised in the following diagram:



The second user interaction can be omitted if there is only one result returned from the query. In almost all cases, there should be an option to 'search again' at the second and third stages in case no results are returned, or if none of the options shown is the required address.

Of course, different applications require different approaches; however, the general principles of the above process apply in all cases where an address is searched for based on user-entered criteria.

## 10.4 Generating a search query from structured user input

Within an interface that accepts structured user input for an address search, it is necessary to 'map' the fields presented to the user with those found within AddressBase or AddressBase Plus. In particular, any query will need to test multiple fields for a given input and will need to combine result sets from the two different address formats of AddressBase Plus (or the single address format of AddressBase) in order to produce the most complete result set.

Generally, a search form will describe a simplified view of an address in order to keep the user interface tidy and intuitive. Users may be given a set of text boxes to fill in, generally including building name, building number, street name, locality name, town name and postcode. The relationships between some common search fields and the fields found in AddressBase Plus are as follows:

Search Box	Mapped Delivery Point fields	Mapped geographic fields
<b>Business Name</b>	Organisation_Name	Organisation PAO_Text SAO_Text
<b>Flat/Subdivision Name</b>	Sub_Building_Name Department_Name	SAO_Text
<b>Flat/Subdivision Number</b>	Sub_Building_Name	SAO_StartNumber SAO_StartSuffix SAO_EndNumber SAO_EndSuffix



Search Box	Mapped Delivery Point fields	Mapped geographic fields
Building Name	Building_Name	PAO_Text
Building Number	Building_Number Building_Name (in cases where a suffix or range is present)	PAO_StartNumber PAO_StartSuffix PAO_EndNumber PAO_EndSuffix
Street	Thoroughfare Dependent_Thoroughfare	Street PAO_Text
Locality	Dependent_Locality Double_Dependent_Locality	Locality Town Street
Town	Dependent_Locality Post_Town	Town Locality
Postcode	Postcode	Postcode_Locator

The above mapping is an example only, and it is possible to breakdown the search fields differently, in which case, a different mapping would be required. The important thing is to consider all possibilities for how data might be recorded. For example, a business name can sometimes appear as an organisation name or a building/PAO name depending on circumstances, so both must be checked when creating a search query.

Numbers need to be handled very carefully due to the presence of suffixes and ranges. There are two options for structuring the search input in these cases:

1. A single 'number' box can be used (as shown above in Flat/Subdivision Number and Building Number), which will then require some string manipulation to split the input into the appropriate numeric range and suffix components in order to search the geographic addresses; or
2. Four boxes can be provided for each number (start number, start suffix, end number and end suffix), which would then need to be combined into an appropriate string to search the Delivery Point Addresses.

## 10.5 Structuring the query for a structured address search

The basic rules to adhere to when generating a search query from structured input are as follows:

1. Ignore any search boxes that are not filled in with values.
2. Where a value is entered, assume that a match on at least one of the mapped fields is essential.

In SQL query terms, this means that each search term should generate a sub-query that searches each of the mapped fields (using OR), and that these sub-queries should then be combined together (using AND) into a single search query. The following SQL code illustrates this (for the Delivery Point Address search only) for an example where a street, locality and town name have been entered by the user:

```
SELECT dp.UPRN, GetFormattedAddress(dp.*) FROM abp dp  
WHERE (dp.thoroughfare = streetsearchtext OR dp.dependent_thoroughfare = streetsearchtext) AND  
(dp.dependent_locality = localitysearchtext OR dp.double_dependent_locality = localitysearchtext) AND  
(dp.dependent_locality = townsearchtext OR dp.post_town = townsearchtext)
```

In the above example, *streetsearchtext*, *localitysearchtext*, and *townsearchtext* (shown in blue) represent user-entered search terms (which could be parameters within an SQL function) and the *GetFormattedAddress(\*)* function is a hypothetical user-defined function that returns the formatted address as a single string (suitable for display in the user interface). For more information on formatting addresses, please see [Section 9](#) of this guide.

On top of this, for a complete query, the two different types of addresses should be queried separately (Geographic and Delivery Point Addresses), and the two result sets should be amalgamated into a single set using a UNION. The following example builds upon the previous example to include Geographic Addresses as well as Delivery Point Addresses.

```
SELECT UPRN, GetFormattedAddress(*) FROM abp  
WHERE (thoroughfare = streetsearchtext OR dependent_thoroughfare = streetsearchtext) AND  
(dependent_locality = localitysearchtext OR double_dependent_locality = localitysearchtext) AND  
(dependent_locality = townsearchtext OR post_town = townsearchtext)  
  
UNION  
  
SELECT uprn, GetFormattedAddress(*) FROM abp WHERE  
(.street_name = streetsearchtext OR .pao_text = streetsearchtext) AND  
(.locality = localitysearchtext OR .town = localitysearchtext OR .street_name = localitysearchtext) AND  
(.town = townsearchtext OR .locality = townsearchtext)
```

The SQL UNION operator will combine the two result sets, discarding any exact duplicates. (Retaining the exact duplicates requires the use of UNION ALL, but that is not desirable in this example.)

The resulting output from this query will be a set of search results as formatted addresses along with their UPRN. Exact duplicates will be omitted, but all ‘variations’ of the same address will be output (one row for each variation, with the same UPRN repeated more than once potentially). It may be wise to return the *Postal Address Flag* values against each to enable further filtering, for example, to restrict the results to postal addresses only. Note that the *Postal Address Flag* is only available in AddressBase Plus. All records in AddressBase are deemed postal as they are from Royal Mail’s PAF data.

## 10.6 Supporting case-insensitive queries and partial matches

A flaw in the above examples is the use of equality operators. In practice, because people do not tend to be consistent with capitalisation of letters, the SQL ‘LIKE’ operator might work better, and depending on the nature of the application, a ‘%’ wildcard could be appended to the end of each search term to allow only the first few letters of an address component to be entered. For example:

post_town LIKE townsearchtext	<i>Case insensitive search in some databases</i>
post_town LIKE (townsearchtext    '%')	<i>Matches post towns that start with the search text</i>
post_town LIKE ('%'    townsearchtext    '%')	<i>Matches post towns that contain the search text</i>

Alternatively, if exact matches are required but case sensitivity is not, then the UPPER() or LOWER() SQL functions can be used on each side of the equals sign in comparisons (a solution that should work in all databases):

UPPER(post_town) = UPPER(townsearchtext)	<i>Case insensitive equality</i>
--	----------------------------------

Finally, to combine all of the approaches, the following would work for maximum flexibility:

```
UPPER(post_town) LIKE ('%' || UPPER(townsearchtext) || '%')
```

## 10.7 Generating a search query from unstructured user input

When offering a ‘search engine’ style search feature with just a single text box to enter search terms, a wholly different approach is required. No assumptions can be made about the order, format or style of the user input, and the data will need to be ‘indexed’ in a way that facilitates searches of this type.

### 10.7.1 Creating a search index for addresses

Search engine style searches are likely to require the creation of an additional index/lookup table for addresses. Such a table is likely to consist of just two main columns: a key value (UPRN) and a formatted address string. Additional columns may be required to allow filtering of results (such as the *AddressBase Postal flag* values from AddressBase Plus, which would allow the results to be filtered by *different address statuses*).

The following table shows a possible address index table structure:

UPRN	Address Text	Statuses (multiple fields)
123456789012	4 THE MEADOWS HIGH STREET WALTHAMSDALE BURRIDGE BU27 9UB	Local Authority
123456789012	FLAT 4 THE MEADOWS HIGH STREET WALTHAMSDALE BURRIDGE BU27 9UB	PAF
123456789013	4 HIGH STREET WALTHAMSDALE BURRIDGE BU27 9UB	Non-postal

Note how the addresses have been formatted as a single text string with a single space between each word (although leaving commas in would do no harm). All forms of each address (both PAF and geographic) have been added to the index, so there can be several rows with the same UPRN. To speed up complex searching, an appropriate index could be added to the Address Text field, such as a full text search index.

### 10.7.2 Structuring the query for an unstructured address search

Once a suitable search index is in place, the query itself can be put together. The basic idea is to split the user input into search terms by removing commas, double spaces, and other unnecessary whitespace and then splitting it at each single space, as follows:

- User input: 4, High Street, westville, wv17

Capitalised, with commas and double-spaces removed:

- 4 HIGH STREET WESTVILLE WV17

Split into separate search terms:

- 4
- HIGH
- STREET
- WESTVILLE
- WV17

Once the user input has been pre-processed into separate search terms, a query can be generated. The key assumption in this example will be that ALL search terms must be matched against the index table to be considered as a result. This implies a query where each value is matched using an 'AND' operator. In order to search the whole index, the 'LIKE' operator will need to be used along with a '%' wildcard on either side of the search text. A suitable search query for the above example would be as follows:

```
SELECT UPRN, AddressText FROM AddressSearchIndex
```

```
WHERE
```

```
AddressText LIKE '%4%' AND  
AddressText LIKE '%HIGH%' AND  
AddressText LIKE '%STREET%' AND  
AddressText LIKE '%WESTVILLE%'  
AND AddressText LIKE '%WV17%';
```

This query would return all rows from the index table that contain all of the search terms, along with the appropriate UPRNs. The following table shows how the index table would be used in the above example to return relevant results:

UPRN	Address text	Statuses (multiple fields)
894756389092	4 HIGH STREET WESTVILLE SUNNYTOWN WV17 7HL	Geographic + PAF
894756389132	ROSE COTTAGE 4 HIGH STREET WESTVILLE SUNNYTOWN WV17 7HL	Geographic

UPRN	Address text	Statuses (multiple fields)
274859037849	FLAT 4 Highbury Court High Street Westville Sunnytown WV17 7HL	Geographic + PAF
482974769830	MAPS4U LTD High Street Westville Sunnytown WV17 7HL	Geographic + PAF

This result set can then be presented to the user, who can select the most appropriate record, which can then be retrieved in full using the UPRN.

Of course, in a practical implementation, the above query would need to be dynamically generated, with a separate condition added for each search term. This example is quite a strict search query that requires all search terms to be present. Many layers of complexity could be added to allow partial and 'fuzzy' matches, and to return confidence scores, for example, but such enhancements are beyond the scope of this guide.

## 10.8 Summary

This guide is intended as an introduction to implementing address search functionality using AddressBase, AddressBase Plus and AddressBase Plus Islands. The following list is a summary of the main points:

1. A user front-end for an address search may contain a single, search engine style text box or multiple text boxes representing different parts of an address.
2. A typical address search function takes place in three stages:
  - a. A user enters search text.
  - b. A query is run, returning a set of possible matches.
  - c. The user selects the address of interest and the full record is then returned.
3. With a structured search interface, the addresses can be queried directly by mapping the various address fields to the text boxes supplied.
4. For an unstructured (single text box) interface, it is necessary to create an index table with fully formatted address strings against each UPRN. Queries can then be run against this index table by splitting the user input into individual search terms and requiring them all to be present.
5. It is possible to filter results by status in AddressBase Plus (for example, *postal* or *non-postal*).
6. Any search function should search all forms of an address (both Geographic and Delivery Point Addresses).
7. Careful consideration should be given to the use of ‘fuzzy’ search algorithms (such as using wildcard or sound-alike searches).

## Annex I – Support links

Filename	Link
AddressBase technical support documents	<a href="https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-support">https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-support</a>
AddressBase Plus technical support documents	<a href="https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-plus-support">https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-plus-support</a>
AddressBase Plus Islands technical support documents	<a href="https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-islands-support">https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-islands-support</a>

