



ORDNANCE SURVEY GB

ADDRESSBASE PREMIUM AND ADDRESSBASE PREMIUM ISLANDS – GETTING STARTED GUIDE

Version history

Version	Date	Description
2.0	04/2020	Revised for AddressBase Premium products. Includes ArcGIS Pro.
2.1	03/2021	Update to Section 7 and removal of OSMA.
2.2	08/2021	Introduction of GeoPackage to AddressBase Premium and AddressBase Premium Islands. Updates and improvements to Sections 5, 6, 7, 8 and 9.

Purpose of this document

This is the Getting Started Guide for the AddressBase Premium and AddressBase Premium Islands products. This Guide provides greater insight into these products and their potential applications. For information on the contents and structure of AddressBase Premium and AddressBase Premium Islands, please refer to the Overview and Technical Specifications. The instructions for AddressBase Premium are also applicable for AddressBase Premium Islands unless otherwise mentioned.

The terms and conditions on which AddressBase Premium and AddressBase Premium 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 Premium and AddressBase Premium 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 Premium and AddressBase Premium 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 Premium contains data created and maintained by English, Welsh and Scottish Local Government as well as Royal Mail and the Valuation Office Agency:

- © Local Government Information House Limited copyright and database rights 2021, or © Hawlfraint a hawlau 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 Premium Islands data are provided by the following authorities:

- Northern Ireland data is 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 Premium and AddressBase Premium 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

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

Contents

1.	Introduction.....	7
1.1	Support documentation available for the products	7
2.	Prerequisites.....	8
2.1	System requirements.....	8
2.2	Backup provision of the products	8
2.3	Typical data volumes	8
3.	Data supply	9
3.1	Supply options.....	9
3.1.1	DVD supply for an area of interest	9
3.1.2	DVD supply of Managed Great Britain Sets.....	9
3.1.3	Secure File Transfer Protocol	9
3.1.4	Download.....	10
3.2	File naming	11
3.2.1	Non-geographic chunks	11
3.2.2	Geographic chunks.....	11
3.3	Unzipping the data	12
3.3.1	Non-geographic chunks	12
3.3.2	Geographic chunks.....	12
4.	Preparing the CSV data	13
4.1	Gawk	13
4.2	Python	15
5.	Loading CSV into GIS software.....	16
5.1	Loading CSV into ArcGIS Pro.....	16
5.2	Loading CSV into ArcGIS Desktop.....	21
5.3	Loading CSV into MapInfo.....	29
5.4	Loading CSV into QGIS	33
5.5	Loading CSV using CadCorp Address Loader	40
5.5.1	Loading CSV into CadCorp SIS Desktop via a database.....	43
6.	Loading CSV into a database	46
6.1	Considerations.....	46
6.1.1	Software dependencies	46
6.1.2	UPRN deletions	46
6.2	Loading CSV into a PostgreSQL database.....	47
6.2.1	Converting coordinates to geometry.....	50
6.3	Loading CSV into an Oracle database	51
6.4	Loading CSV into a Microsoft SQL Server.....	52
6.4.1	Setting Primary and Foreign Keys	54
6.4.2	Creating the point geometry	54
7.	Accessing GeoPackage data via GIS software	55
7.1	Accessing GeoPackage data via ArcGIS Pro.....	55

7.2	Accessing GeoPackage data via ArcGIS Desktop.....	61
7.3	Accessing GeoPackage data via QGIS.....	66
8.	Loading GeoPackage into a database	70
8.1	Loading a GeoPackage into PostgreSQL using GDAL	70
9.	Loading GML	72
9.1	Loading GML via a translator	72
9.2	Loading GML into FME	72
9.3	Using GDAL to load GML into a database	74
10.	COU implementation guidance	76
10.1	Introduction to COU.....	76
10.2	Types of change within a COU.....	76
10.3	High-level COU implementation model	77
10.4	High-level COU implementation model – with archiving	78
10.5	Applying COU to tables	78
10.5.1	Changes to the BLPU table.....	78
10.5.2	Changes to the Classification table	79
10.5.3	Changes to the Organisation table	83
10.5.4	Changes to the Delivery Point Address table	84
10.5.5	Changes to the Land and Property Identifier table.....	85
10.5.6	Changes to the Street table.....	87
10.5.7	Changes to the Street Descriptor table	88
10.5.8	Changes to the Cross Reference table	89
11.	Creating a single-line or multi-line address using AddressBase Premium products	90
11.1	Delivery Point Address vs. Geographic Address.....	90
11.2	Background to single-line and multi-line address labels	91
11.3	Delivery Point Address.....	92
11.3.1	Building a single-line Delivery Point Address	94
11.3.2	Building a multi-line Delivery Point Address	95
11.4	Geographic Address (Local Authority Address)	95
11.4.1	Constructing a single address label from the Geographic Address fields	96
11.4.2	Linking address components	96
11.4.3	Rendering SAOs and PAOs	97
11.4.4	Street description, town, locality and administrative area names	98
11.4.5	Building a single-line Geographic Address.....	100
11.4.6	Building a multi-line Geographic Address	100
11.5	Advice on creating mailing lists using AddressBase Premium products.....	103
11.5.1	Address status.....	103
11.5.2	Other filters available in AddressBase Premium products for use in addressing labels.....	104
12.	Searching for addresses.....	105
12.1	Free text search.....	105
12.2	Structured component search	106
12.3	Understanding the different addresses available.....	107

12.4 The search operation 108

12.5 Generating a search query from structured user input..... 108

12.6 Structuring the query for a structured address search..... 110

12.7 Supporting case-insensitive queries and partial matches 111

12.8 Generating a search query from unstructured user input 111

12.9 Creating a search index for addresses..... 112

12.10 Structuring the query for an unstructured address search..... 113

12.11 Summary of Section 12: Searching for addresses 114

Annex I: Support links..... 115

I. Introduction

AddressBase products are created by bringing together different address sources:

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

The instructions contained in this guide show you how to use AddressBase Premium and AddressBase Premium Islands products. The data is supplied as comma-separated values (CSV), Geography Markup Language (GML) and GeoPackage.

This guide covers the pre-processing steps required to attach the Header files to the data and details how to load the three data formats into various popular geographic information system (GIS) software applications and / or databases, including how to apply a change-only update (COU). It is important to note that this guide does not contain an extensive list of all the applications the products can be used with, and many other GIS software applications and / or databases can be used to load the product.

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. Single line addresses can be used to create a look-up service that customers can use to search for an address.

Please note that the instructions given in this guide for AddressBase Premium are also applicable for AddressBase Premium Islands unless otherwise mentioned.

I.1 Support documentation available for the products

Table 1 lists the support documentation available for the AddressBase Premium and AddressBase Premium Islands products. The documentation can be accessed through the OS website.

Table 1: Support documentation associated with the AddressBase Premium and AddressBase Premium Islands products.

Product name	Associated documents
AddressBase Premium	Overview
	Getting Started Guide
	Technical Specification
AddressBase Premium Islands	Getting Started Guide
	Technical Specification

Throughout this document, we reference files or documents as part of your 'Getting Started' instructions. For ease of reference, we've compiled a list of these in [Annex 1: Support links](#).

2. Prerequisites

2.1 System requirements

AddressBase Premium is an addressing gazetteer offering full property life cycle information that can be used within GIS and database systems. For details of Ordnance Survey's licensed partners, who can incorporate the AddressBase products into their systems, please see the [systems / software page on the Ordnance Survey website](https://www.ordnancesurvey.co.uk/business-government/tools-support/gis/software) (<https://www.ordnancesurvey.co.uk/business-government/tools-support/gis/software>).

Ordnance Survey does not recommend specific suppliers or software products as the most appropriate system will depend on many factors, for example, 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

2.2 Backup provision of the products

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.

Table 2 lists the unzipped file sizes for the GB full supply of the two products.

Table 2: Unzipped file sizes for the GB full supply of the AddressBase Premium and AddressBase Premium Islands products.

Product name	Unzipped CSV file size	Unzipped GML file size	Unzipped GeoPackage file size
AddressBase Premium	40 Gb	186 Gb	28 Gb
AddressBase Premium Islands	1 Gb	4 Gb	0.8 Gb

3. Data supply

3.1 Supply options

3.1.1 DVD supply for an area of interest

When you receive an order via hard media (DVD), the following files will be supplied if the supply is not a Managed Great Britain Set (MGBS):

- Data
- Doc
- Order_Details.txt

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

A text file called Label Information can be found within the Doc directory. This file is a replica of the information on the DVD should you need to reproduce or reprint this.

The Order_Details.txt provides information about the supply, including supply type, format, dates of order, currency and change. It also contains a list of all the zipped folders contained in the Data directory.

3.1.2 DVD supply of Managed Great Britain Sets

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

- Data
- Doc
- Order_Details.txt

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

A text file called Label Information can be found within the Doc directory. This file is a replica of the information on the DVD should you need to reproduce or reprint this.

The Order_Details.txt file provides information about the supply, including supply type, format, dates of order, currency and change. It also contains a list of all the zipped folders contained in the Data directory.

3.1.3 Secure File Transfer Protocol

With a Secure File Transfer Protocol (SFTP) order, the same information is supplied as in Section 3.1.1 (i.e. DVD supply for an area of interest), but the file names will be slightly different, reflecting the SFTP order number.

3.1.4 Download

The AddressBase Premium and AddressBase Premium Islands products are available as online downloads for customers who have signed up to the Public Sector Plan on the OS Data Hub. From October 2021, Premium Plan members will be able to download AddressBase Premium and AddressBase Premium Islands data from the OS Data Hub. Data can be downloaded in various formats.

Public Sector Plan members should use the following instructions to download the products from the OS Data Hub:

1. Sign in to your PSGA organisation account on [OS Data Hub \(https://osdatahub.os.uk/\)](https://osdatahub.os.uk/).
2. Click *Download* on the top navigation bar. You will be taken to the *Premium Data Downloads* catalogue.
3. Search and select the premium product you want.
4. Click the *Add data package* button.
5. Enter a data package¹ name. We suggest using a name that will help you easily identify the data package later. Data package names don't have to be unique.
6. Choose your area. For user-defined areas, draw your area of interest (AOI) using the drawing tools or upload a polygon to the map from the polygon library.
7. Select *file format*. We show only the available file formats for that selection.
8. Select *updates*. We show only the available updates for that selection.
9. Click the *Create data package* button.
10. The confirmation page tells you the data package is being created, with options to *Add another data package* or click to *view the data package status* in the data package list. You will also receive a confirmation email when a data package is created.
11. The data package list displays all the data packages for your organisation. Once a data package has been fulfilled, the status will change from *Available soon* to *Download*. You will also receive an email when the data package is ready to download.
12. Click the *Download* button to open the data package and download the zip file(s).



¹ A data package is a downloadable file or group of files containing the product data you want to receive. There can be more than one version of a data package when new product updates are available to download. To download (order) data, you create a data package in the OS Data Hub for the product you want, and specify the area, format and update cycle. Once the system has fulfilled the data package, you can download it from the data package list, under Data packages on the left menu in your OS Data Hub account.

3.2 File naming

The data is supplied as chunked files that cover your selected area. These files are named according to the convention shown in the following sub-sections.

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

3.2.1 Non-geographic chunks

For a MGBS supply of CSV and GML, you will see a single zip folder when you receive your download data. On opening the data link folder, you will see a series of separate unzipped files.

For example:

AddressBasePremium_FULL_2011-07-29_001_csv.zip (full supply of CSV)

or

AddressBasePremium_COU_2011-07-29_001_gml.zip (COU supply of GML)

For a MGBS supply of GeoPackage, you will see a single zip folder when you receive your download data. On opening the data link folder, you will see a single unzipped file.

For example:

AddressBasePremium_FULL_2011-07-29_001_gpkg.zip (full supply of GeoPackage)

3.2.2 Geographic chunks

When you receive your geo-chunked download data for CSV and GML, you will see a series of zipped folders on opening the data.

For example:

AddressBasePremium_FULL_2011-07-29_TQ2020_csv.zip (full supply of CSV)

or

AddressBasePremium_COU_2011-07-29_TQ2020_gml.zip (COU supply of GML)

When you receive your geo-chunked download data for GeoPackage, you will see a single zipped folder on opening the data.

For example:

AddressBasePremium_FULL_2011-07-29_001_gpkg.zip (full supply of GeoPackage)

3.3 Unzipping the data

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

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

When the files are unzipped, they are ready for use and will appear as follows:

3.3.1 Non-geographic chunks

AddressBasePremium_FULL_2020-02-11_001.csv
AddressBasePremium_ISL_FULL_2020-02-18_001.gml
AddressBasePremium_FULL_2020-02-18_001.gpkg

3.3.2 Geographic chunks

AddressBasePremium_2011-07-29_NC4040.csv

For GeoPackage supply, one GeoPackage will be supplied which will contain all the tiles:

AddressBasePremium_FULL_2020-02-18_001.gpkg

4. Preparing the CSV data

The AddressBase Premium CSV and AddressBase Premium Islands CSV are provided with records for various tables incorporated into a single CSV. Depending on the size of the area of interest (AOI) or if you require a full supply of GB or Islands data, there may be multiple CSV files supplied. These CSV files need to be split into the individual records and tables.

There are multiple methods for splitting AddressBase Premium CSV files by the record identifiers. The following sub-sections step through using either Gawk or Python for splitting the data and appending the Header file.

4.1 Gawk

The following instructions show you how to use Gawk to split the AddressBase Premium CSV files and append the Header files.

1. Group all the AddressBase Premium CSV files into an empty folder. It is very important to ensure that the folder does not contain other CSV files. The folder must contain no spaces in its file directory path name, for example, *C:\AddressBaseData\AddressBase_Premium*.
2. Open the [AddressBase Premium Product Support page on the OS website](https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-premium-support) (<https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-premium-support>).
3. Navigate to the *Technical information* section on the aforementioned web page (as shown in Figure 1) and click on [AddressBase Premium header files \(zip\)](#). This will automatically download the zip folder to your machine.

Note: The header files are also applicable to the AddressBase Premium Islands product.

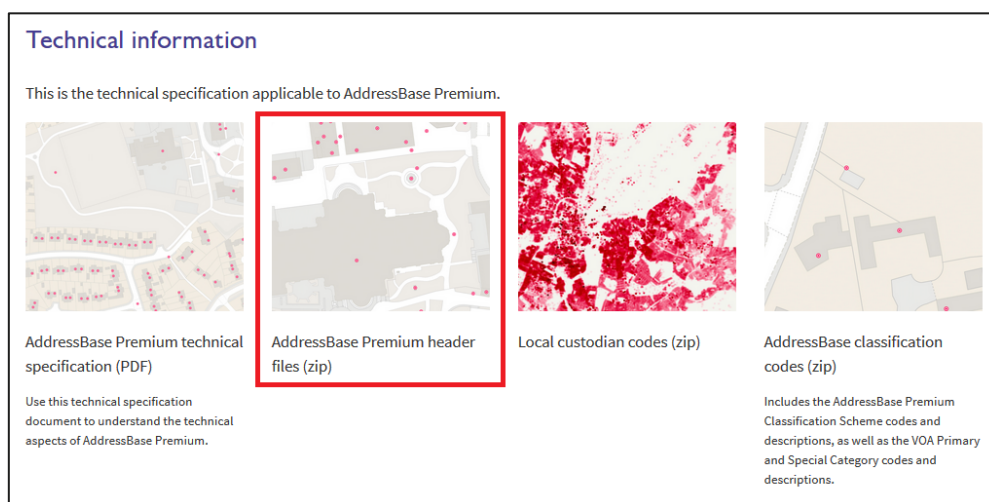


Figure 1: Location of the AddressBase Premium Header files on the AddressBase Premium Product Support page of the OS website

4. Extract the downloaded zip and you should have the following CSV files:

Record_10_HEADER_Header.csv
Record_11_STREET_Header.csv
Record_15_STREETDESCRIPTOR_Header.csv
Record_21_BLPU_Header.csv
Record_23_XREF_Header.csv
Record_24_LPI_Header.csv
Record_28_DELIVERYPOINTADDRESS_Header.csv
Record_29_METADATA_Header.csv
Record_30_SUCCESSOR_Header.csv
Record_31_ORGANISATION_Header.csv
Record_32_CLASSIFICATION_Header.csv
Record_99_TRAILER_Header.csv

5. Place the extracted AddressBase Premium Header files into the folder with the CSV files.
6. Go to the [AddressBase repository on GitHub](https://github.com/OrdnanceSurvey/AddressBase/tree/master>Loading_Scripts/Gawk)
(https://github.com/OrdnanceSurvey/AddressBase/tree/master>Loading_Scripts/Gawk).
7. Open the *AddressBasePremium_GawkSplitScript.bat* file and copy the contents to a text editor such as TextPad or NotePad++. Save this text file as a .bat file in the same folder as your AddressBase Premium data and Header files.
8. In the same link in Step 6, download and extract the zip file called *gawk-4.0.2-bin.zip*.
9. This will extract a file called *Gawk.exe*. Place this file in the same folder as your AddressBase Premium data and Header files.
10. Double-click on the *AddressBasePremium_GawkSplitScript.bat* file and an MS-DOS window will appear. Once the process is complete, the screen will close automatically, or you will have to press any key to continue.

Note: Running the .bat file creates temporary files and requires extra space in the location you are creating your files. These files can be much larger than the original CSV files. They are deleted once the process has finished, but the space is still required.

11. Running this should create additional files which will have a similar naming convention to that of the Header files. These new files should have generated in the same location as the data and headers. For example, looking in the location there should now be an *ID24_LPI_RECORDS* as well as a *Record_24_LPI_Header.csv*.

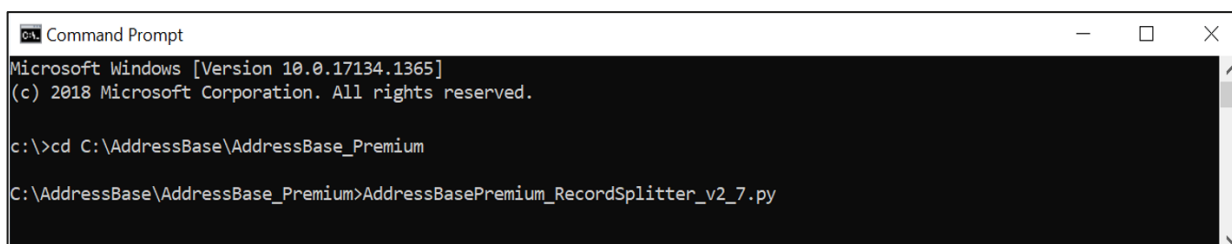
4.2 Python

The following instructions show you how to use Python instead of Gawk to split the AddressBase Premium CSV files and append the Header files. These instructions are based on Python 2.7.

1. Group the AddressBase Premium CSV files into an empty folder. It is very important to ensure that the folder does not contain other CSV files. The folder must contain no spaces in its file directory path name, for example, *C:\AddressBaseData\AddressBase_Premium*.

Note: You do not need to download the Header files if you are using Python to split the AddressBase Premium data.

2. Go to the [AddressBase repository on GitHub](https://github.com/OrdnanceSurvey/AddressBase/tree/master/Loading_Scripts/Python) (https://github.com/OrdnanceSurvey/AddressBase/tree/master/Loading_Scripts/Python).
3. Open the *AddressBasePremium_RecordSplitter_v2_7.py* file and copy the contents to a text editor such as TextPad or NotePad++. Save that text file as a .py file to the same folder as your AddressBase Premium data.
4. Open a Command Prompt window by going to *Windows Start > CMD prompt*.
5. Within the Command Prompt, type *cd* and the directory path where you just placed the *AddressBasePremium_RecordSplitter_v2_7.py* file, then hit *Enter* on your keyboard.
6. Type the name *AddressBasePremium_RecordSplitter_v2_7.py* into the Command Prompt or select it using the *Tab* key, which will display each file within the file directory in turn. Once selected, hit the *Enter* key.



```
Command Prompt
Microsoft Windows [Version 10.0.17134.1365]
(c) 2018 Microsoft Corporation. All rights reserved.

c:\>cd C:\AddressBase\AddressBase_Premium

C:\AddressBase\AddressBase_Premium>AddressBasePremium_RecordSplitter_v2_7.py
```

7. The following message should be displayed:
This program will split OS AddressBase Premium Zip CSV or extracted CSV files by record identifier into new CSV files. Please type in the full path to the directory of OS AddressBase zip files: Directory Path:
8. Enter the full directory path to where you have stored the AddressBase Premium data (for example, *C:\AddressBaseData\AddressBase_Premium*). Hit *Enter* on your keyboard. The process of splitting the files will then begin.
9. When the file splitting process is complete, the following message will be displayed: *The program will close in 10 seconds*. You may still need to close the Command Prompt window afterwards by typing *Exit* and hitting *Enter*.
10. If you navigate to the folder which contained the .py file and the AddressBase Premium CSV files, you should find new files which will have a similar naming convention to that of the Header files. These new files will contain all of your AddressBase Premium data split out by record type.

5. Loading CSV into GIS software

This section provides step-by-step instructions on how to load the CSV format of AddressBase Premium products into commonly used GIS software, including ArcGIS Pro, ArcGIS Desktop, MapInfo, QGIS and CadCorp SIS Desktop.

It is assumed that you will have followed the CSV data preparation steps covered in [Section 4](#) before you attempt to load the data. If this pre-processing is not carried out, there may be errors with loading.

5.1 Loading CSV into ArcGIS Pro

There are two methods for loading AddressBase Premium CSV data into ArcGIS Pro. We will only step through the first method in this guide. If you need guidance on the second method, please refer to the ArcGIS Pro instructions in the [Getting Started Guide for AddressBase, AddressBase Plus and AddressBase Plus Islands](#).

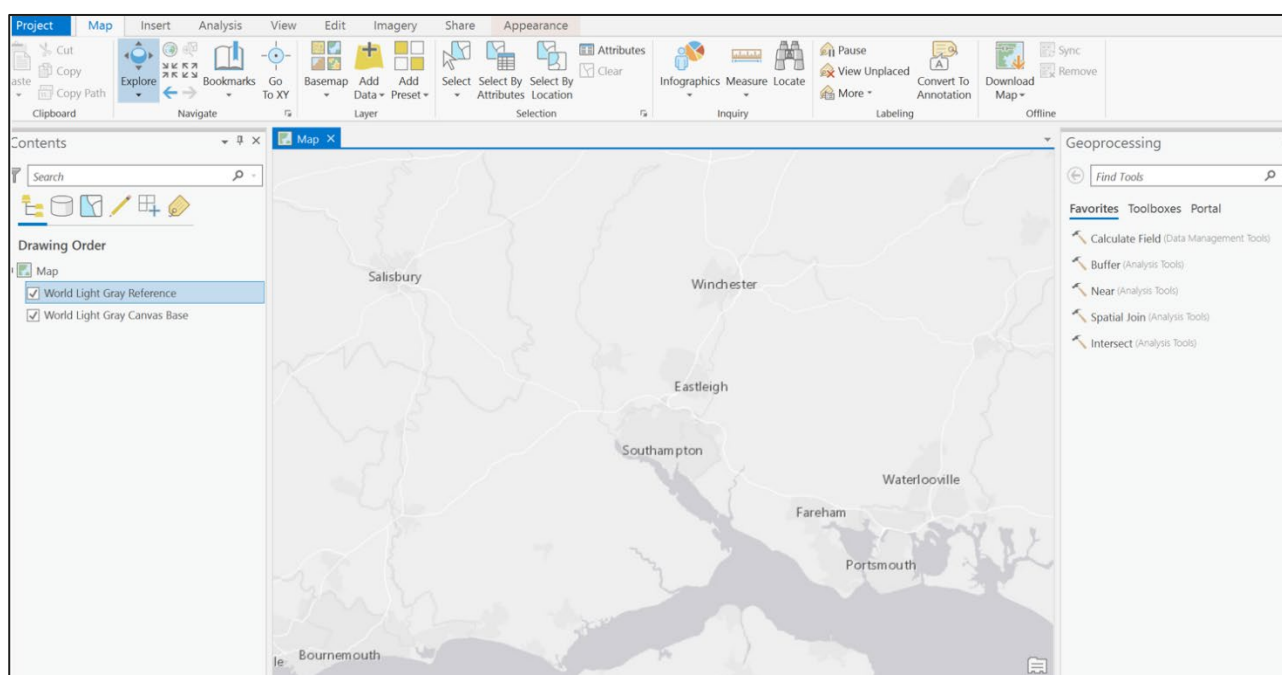
- The first method uses the UK Data Loader published by ESRI, which loads the data into a File Geodatabase and automatically *relates* between the different tables. At the end of the process, you have a fully working dataset ready for use. The following step-by-step instructions in this sub-section use this method.
- The second method must have the CSV data prepared as described in [Section 4](#). These files are imported into a project File Geodatabase within ArcGIS Pro (see the instructions in the [Getting Started Guide for AddressBase, AddressBase Plus and AddressBase Plus Islands](#)). Only the Basic Land and Property Unit (BLPU) table and Streets table will contain geometry which can be mapped by adding XY data. All the other tables will have no geometry, but these tables can be linked together using the Relates function within ArcGIS Pro. Please refer to the Section 2 of the [AddressBase Premium Technical Specification](#) for information on which table fields should be linked together.

The following instructions are based on ArcGIS Pro version 2.3.3.

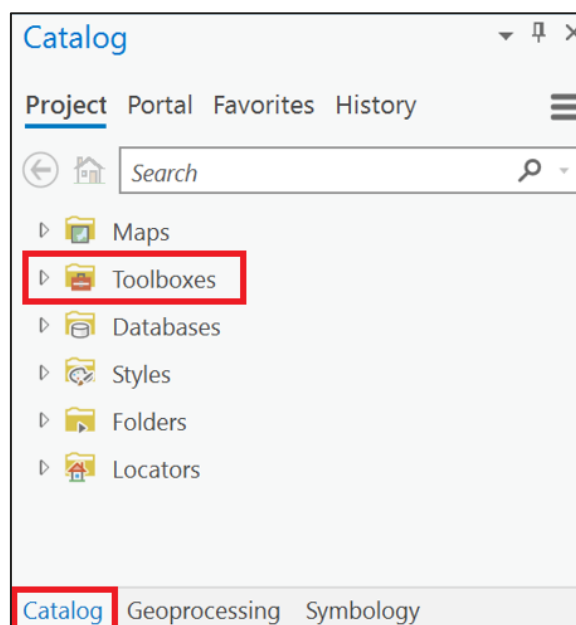
Note: Before following these instructions, it is assumed that you have the full data Interoperability Extension installed and that it matches your ArcGIS Pro version.

1. Launch ArcGIS Pro and create a new project.
2. Select a location to save the project to, name the project and click OK.

If the project is a new map project, it will launch with a default map backdrop supplied by ESRI. You can change this backdrop to another ESRI backdrop or use your own.



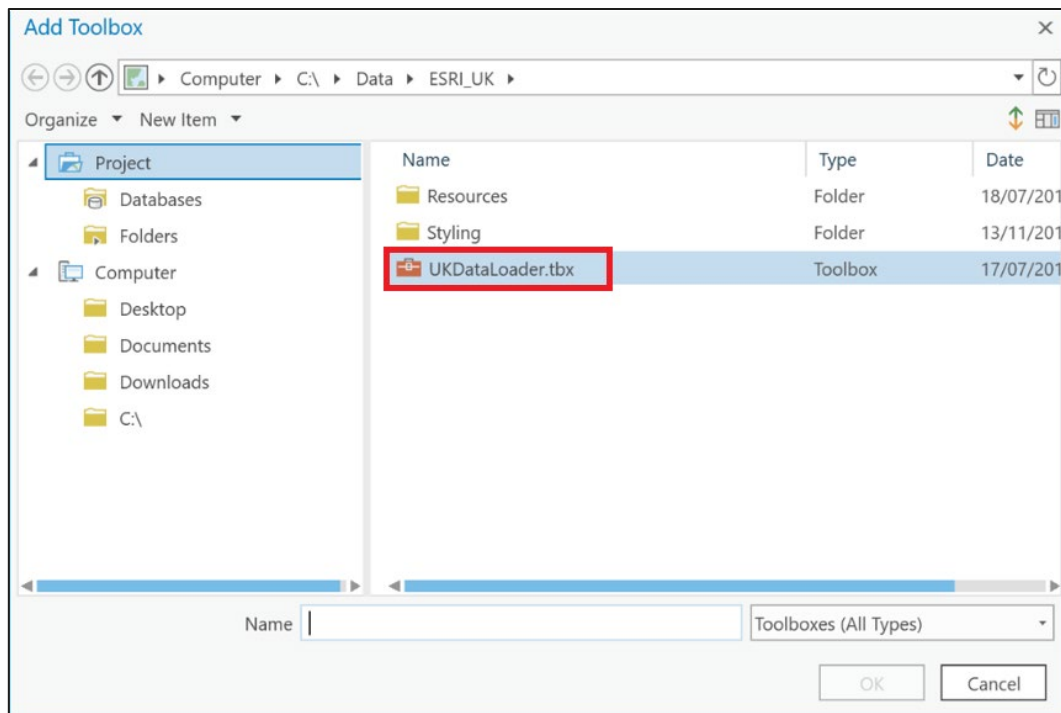
3. Select the *Catalog* pane on the right-hand-side of the screen.
4. Under the Project list, right-click on *Toolboxes* and select *Add Toolbox*.



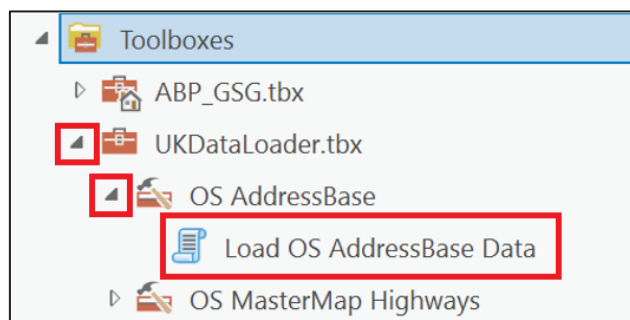
5. Navigate to where the UK Data Loader toolbox is held on your system, for example, C:\Data\ESRI_UK.

Note: The latest version of the UK Data Loader toolbox should be obtained from ESRI.

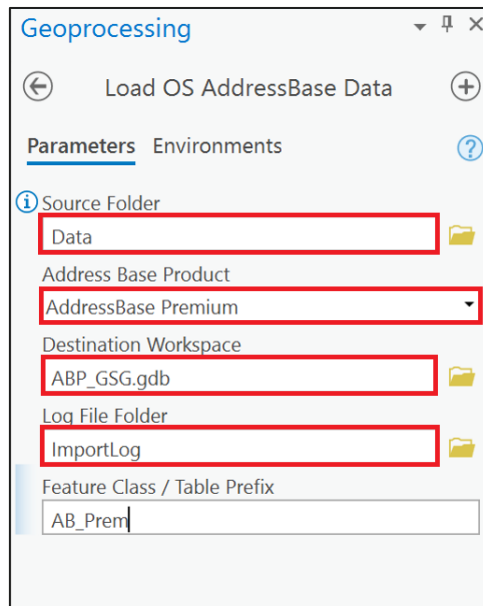
6. Select the *UK Data Loader* toolbox (.tbx) and click *OK*.



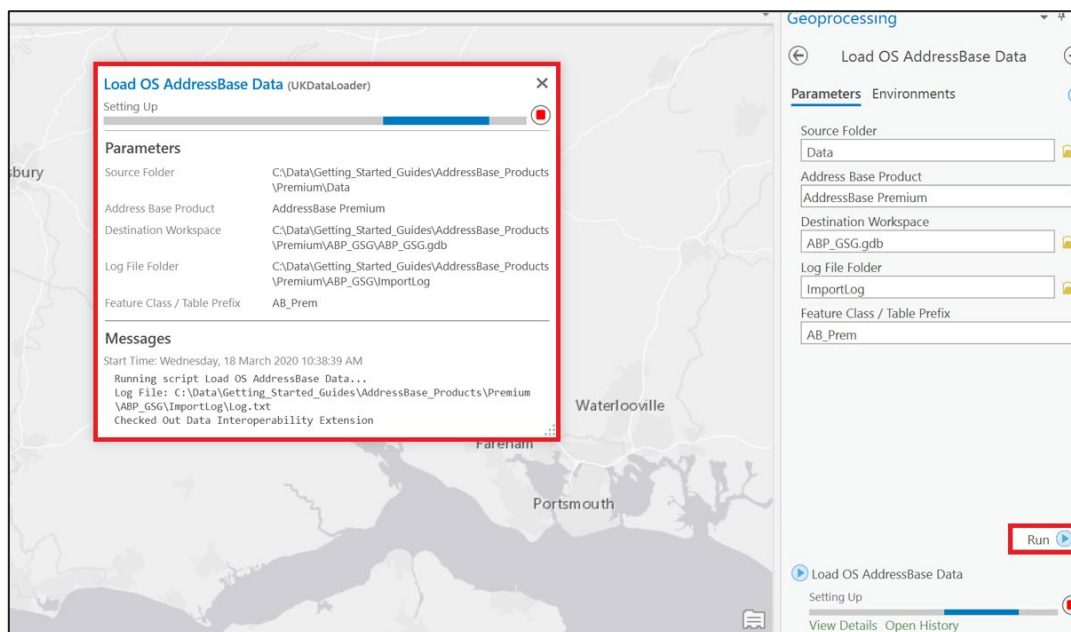
7. The UK Data Loader Toolbox (UKDataLoader.tbx) will now appear in the Catalog list of available toolboxes. Expand it by clicking on the arrow to the left of the entry, then click the arrow again for the *OS AddressBase* entry.
8. Double-click on the *Load OS AddressBase Data* script that should be listed.



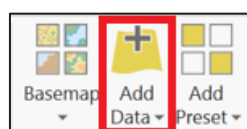
9. A new Geoprocessing window will appear. There are some mandatory blank fields which you need to complete before the script can be run. For the *Source Folder*, enter the directory path where AddressBase Premium data is located. Select your *Address Base Product* from the dropdown menu, then enter a location for the *Destination Workspace* and *Log File Folder*. Other fields are optional.



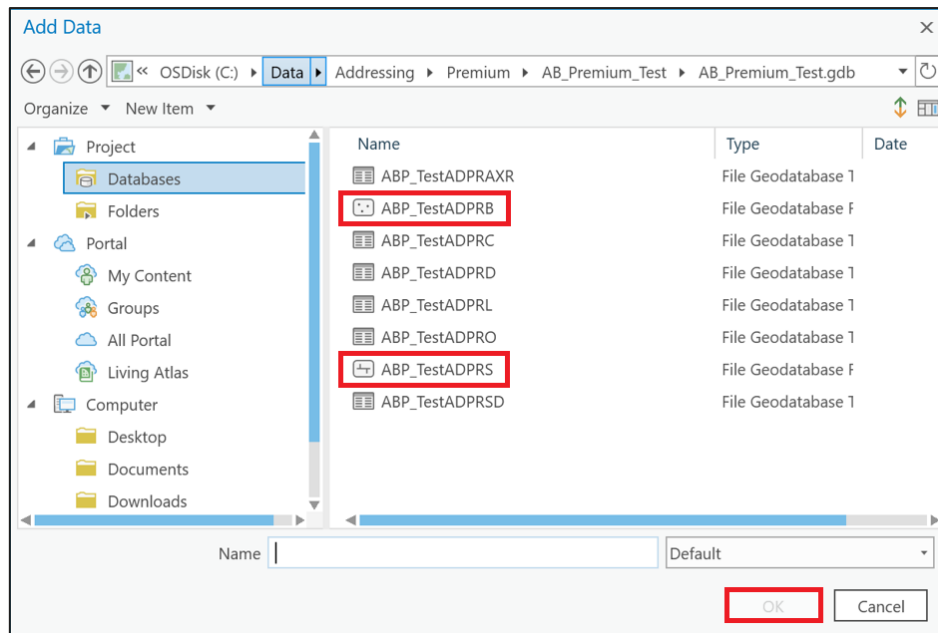
10. When the required fields are completed, click the *Run* button at the bottom of the window.



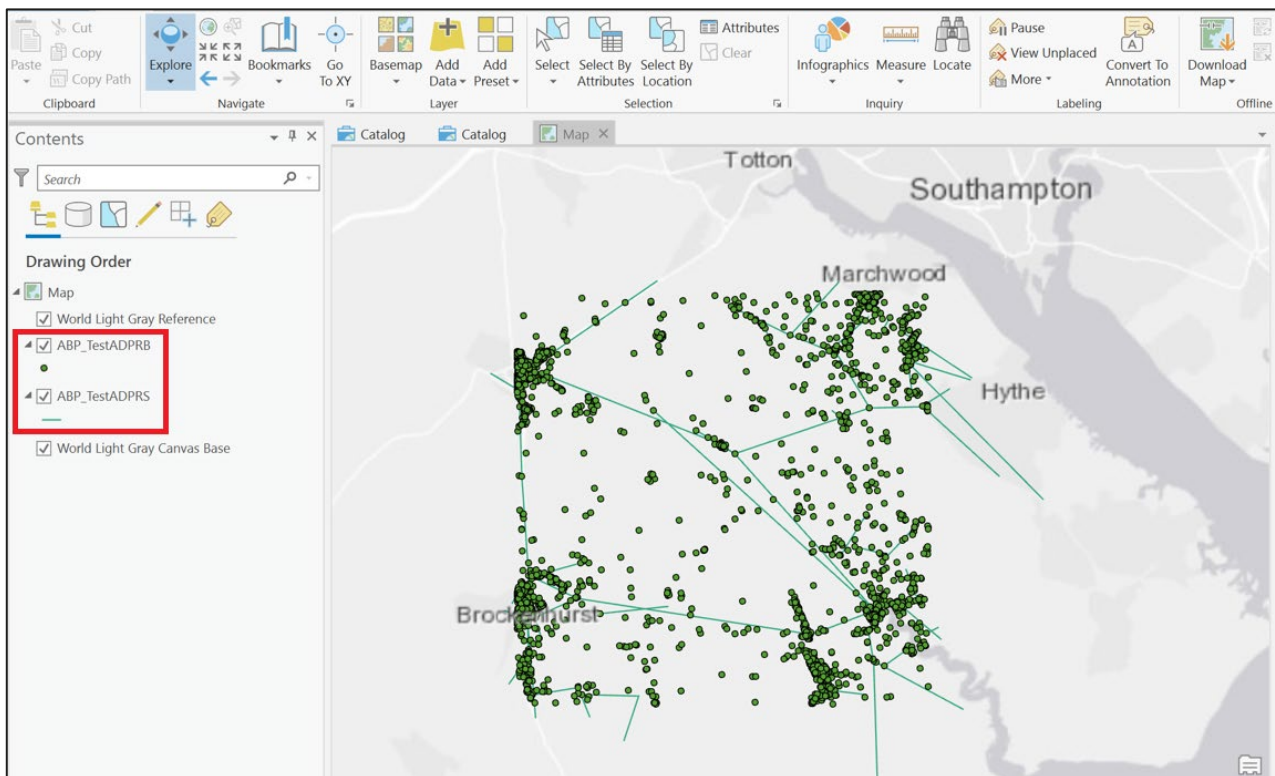
11. This can run for some time, depending upon the amount of data being processed. Wait until a message appears saying the files have been imported successfully.
12. When the import has completed, click the *Add Data* button in the main window menu, then navigate to the project File Geodatabase.



13. You will see that some of the feature classes have no geometry, whilst others do. Select those required and click **OK**. In the example below, **ABP_TestADPRB** refers to geometry created from the **BLPU** table and **ABP_TestADPRS** refers to geometry created from the **Street** table.



14. The data will be added as new features classes in the Contents window. The feature classes containing geometry will be displayed against the map backdrop in the Map window.



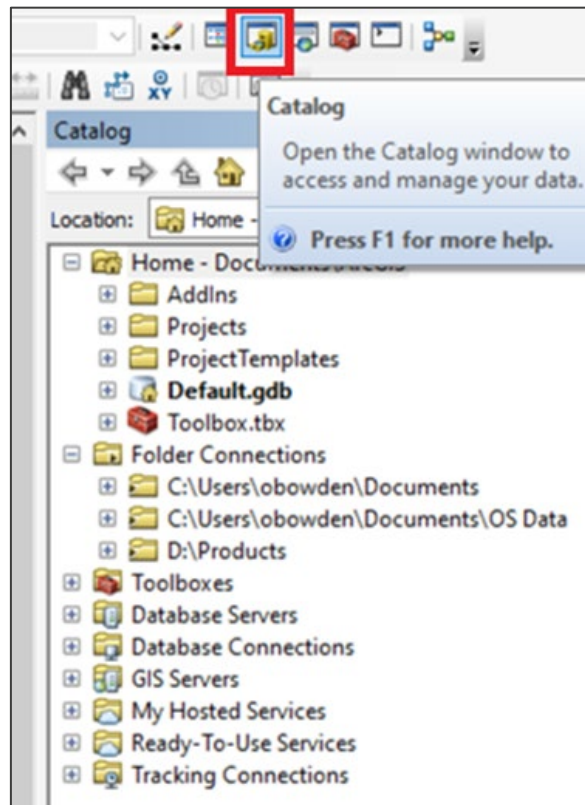
You have now successfully loaded AddressBase Premium data into ArcGIS Pro.

5.2 Loading CSV into ArcGIS Desktop

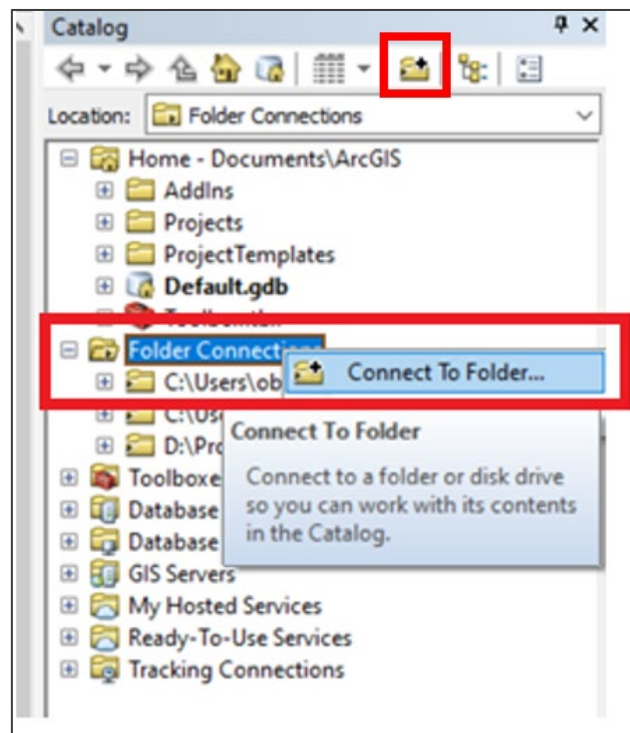
The following step-by-step instructions are based on ArcGIS Desktop version 10.

When using CSV data in ArcGIS Desktop, it is necessary to have split out the individual record types from the original single CSV file and to append column Headings. Before following these instructions, ensure the splitting and Headings have already been prepared as described in [Section 4](#).

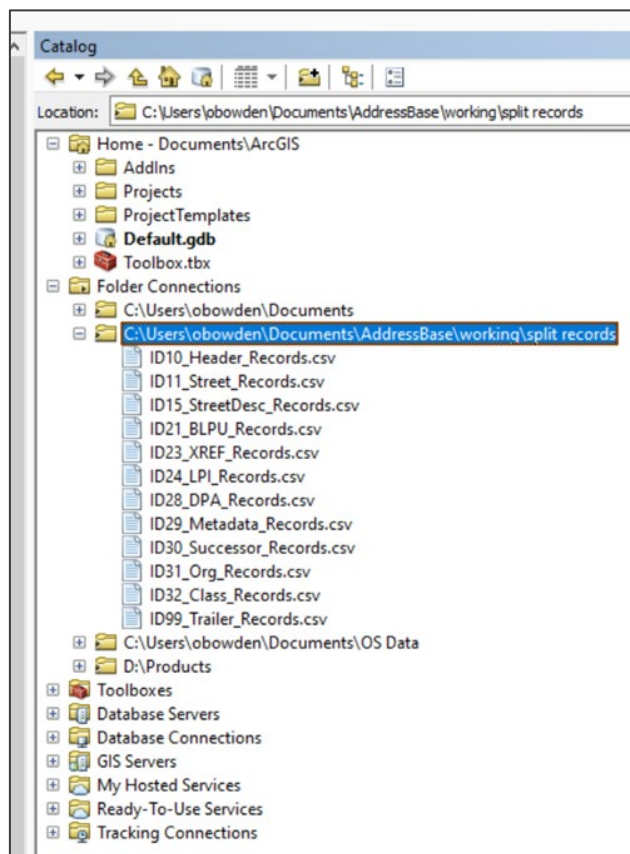
1. Start ArcCatalog as a separate program, or within ArcMap if you are using version 10.



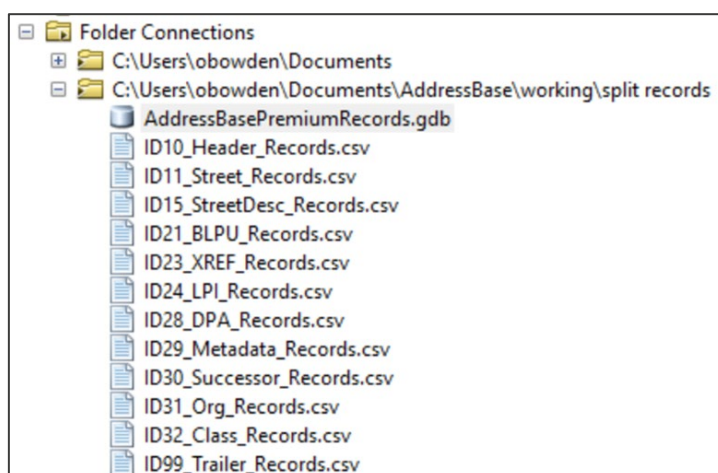
2. Connect to the folder where the AddressBase Premium data is located (for example, *C:\AddressBasePremium_Data*):
 - a. Click *File*, or in version 10, select *Folder Connections*.
 - b. Click *Connect Folder* in the top ribbon or right-click on *Folder Connections* > *Connect To Folder*, then navigate to the folder containing the data.



- 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.

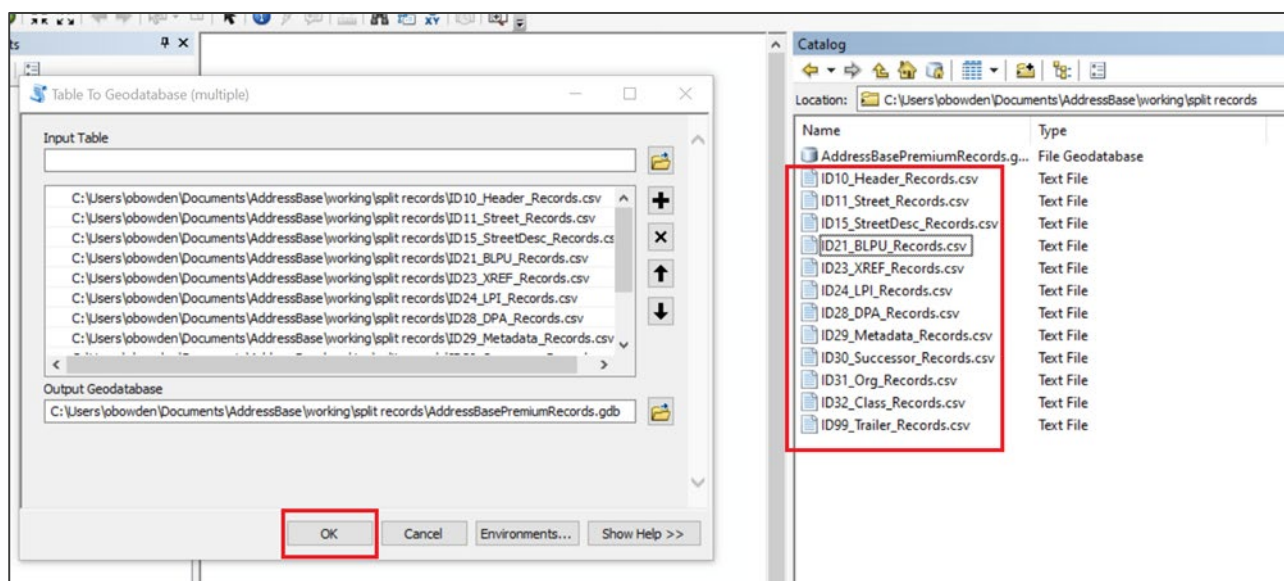


4. Create a File Geodatabase (.gdb) to store the address data. Using the file tree, go to *Folder Connections* and navigate to the directory where you wish to create the File Geodatabase, for example, *C:\AddressBase_Geodatabase\AddressBase_Premium*. This may need to be set up as a new connection.
5. Right-click on the folder where you wish to store the File Geodatabase and select *New > File Geodatabase*.
6. It will be created and named by default as *New File Geodatabase*. Rename it to a name of your choice.

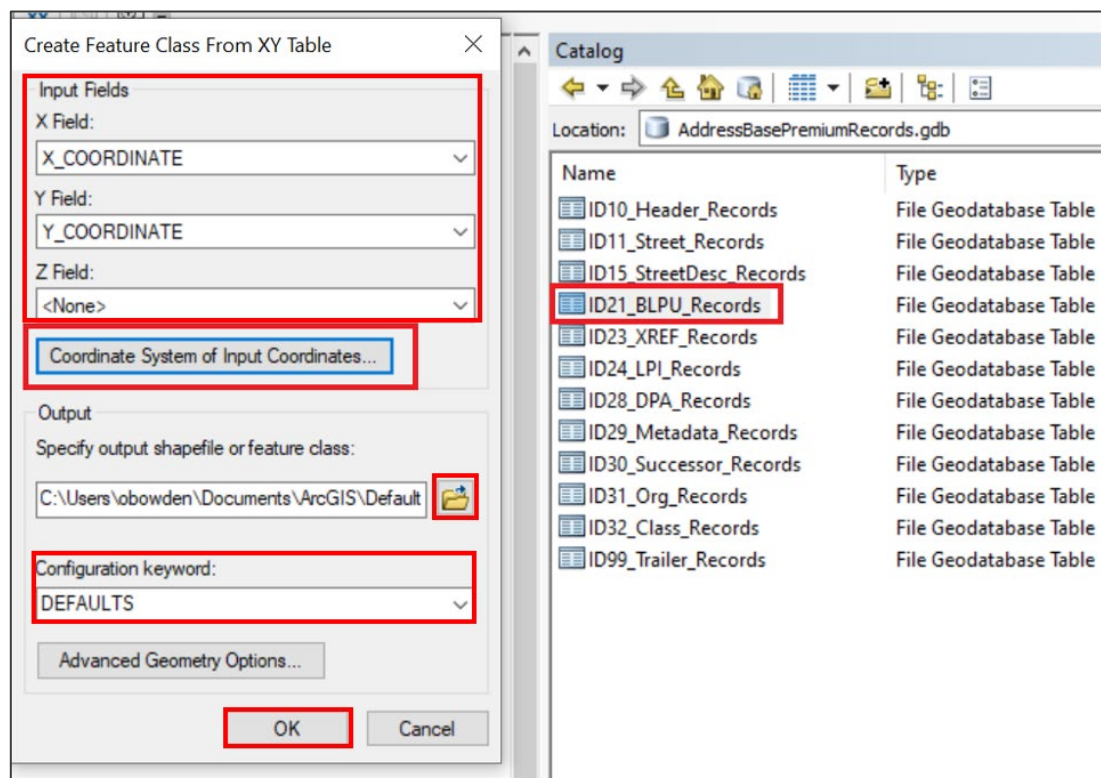


7. Right-click on your new File Geodatabase and select *Import > Table (multiple)...*
8. For *Input Table*, navigate to the location of the CSV files that you wish to open, i.e. the folder that contains the AddressBase Premium data split into individual files by record type.
9. Select the files that you wish to add. Make sure you add the data files (which have the following naming convention: *ID21_BLPU_Records.csv*) and not the Header files (which have the following naming convention: *Record_21_BLPU_Header.csv*).
10. Click *Add*.
11. The *Output Geodatabase* option should automatically be populated by the location of the File Geodatabase that is to be updated; this should be the File Geodatabase you created in steps 4 to 6.
12. Click *OK*, and once the process is complete, click *Close*.

Note: The process may take some time, depending on the number of files and the amount of data being added. The window may close before the operation is complete, so if you cannot see all of your expected files under your File Geodatabase, this may mean that the data is still being loaded.



13. To create a map of the locations of the AddressBase Premium records, they need to be geocoded. To do this, double-click on the Geodatabase that the data was just imported into. The BLP table and the Streets table are the only tables in AddressBase Premium that carry geocoded information.
14. Right-click on the Geodatabase table that was created from AddressBase Premium records with a record type of 21 or 15 (for example, ID21_BLP_Records.csv or ID15_StreetDesc_Records.csv) and select *Create Feature Class*.
15. In the *Create Feature Class From XY Table* dialog box, 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*. Leave the Z Field as *<None>*.



16. For *Input Coordinates*, click on *Coordinate System of Input Coordinates > Select* and then 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 before, then you need to select 'ETRS89 [EPSG: 4258]' for the coordinate system instead.

17. Double-click on the selected coordinate system, then click *Apply* and *OK*.

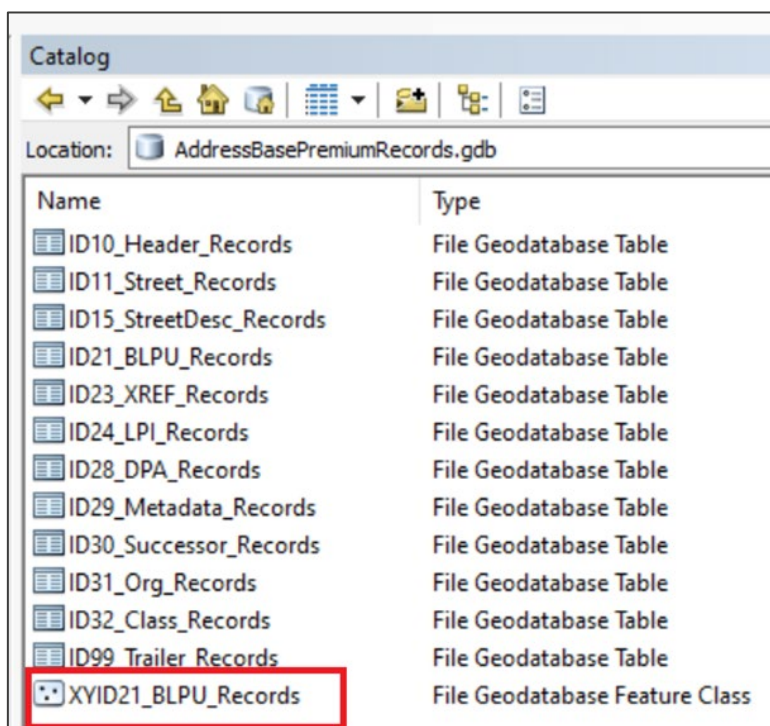
18. Click on the folder icon alongside the *Output* field and navigate to the location where you wish to save the output shapefile or feature class (we recommend that this be within the Geodatabase you created in steps 4 to 6).

Note: If you can't see your Geodatabase, ensure the 'Save as type' box at the bottom of the dialog box is set to 'File and Personal Geodatabase feature classes'.

19. Give the file a suitable name (for example, *XYID21_BLPV_Records*).

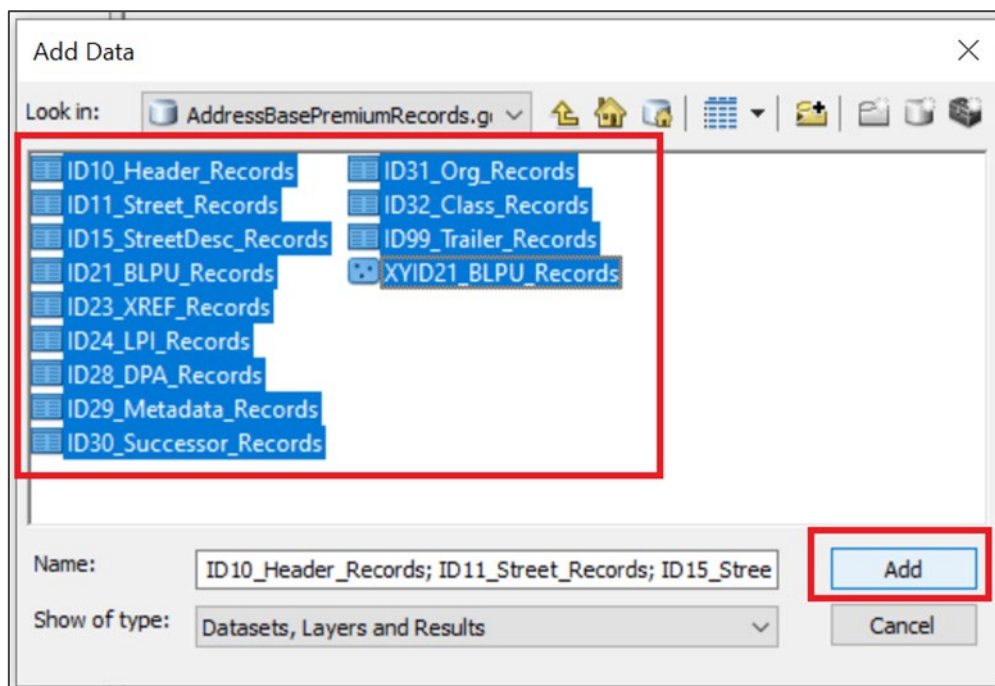
20. Click *Save*, then leave the *Configuration* keyword as *DEFAULTS* and click *OK*.

Note: You may need to right-click on the Personal Geodatabase where it was saved and select 'Refresh' in order to see your new feature class.



Now the processing has been done, the data needs to be loaded into ArcMap so that the individual tables can be *related*.

21. Start ArcMap if you have not been working within ArcMap version 10 already.
22. Select *File > Add Data...*
23. Navigate to the folder where the Geodatabase was created.
24. Double-click on the Geodatabase and select all the files inside.
25. Click *Add*.



26. You need to create the following *joins / relates* between the tables, as stated in the [AddressBase Premium Technical Specification](#):

BLPU (ID21_BLP_Records):

- UPRN – Application Cross Reference (ID23_XREF_Records) UPRN
- UPRN – LPI (ID24_LPI_Records) UPRN
- UPRN – Delivery Point Address (ID28_DPA_Records) UPRN
- UPRN – Organisation (ID31_Org_Records) UPRN
- UPRN – Classification (ID32_Class_Records) UPRN

LPI (ID24_LPI_Records):

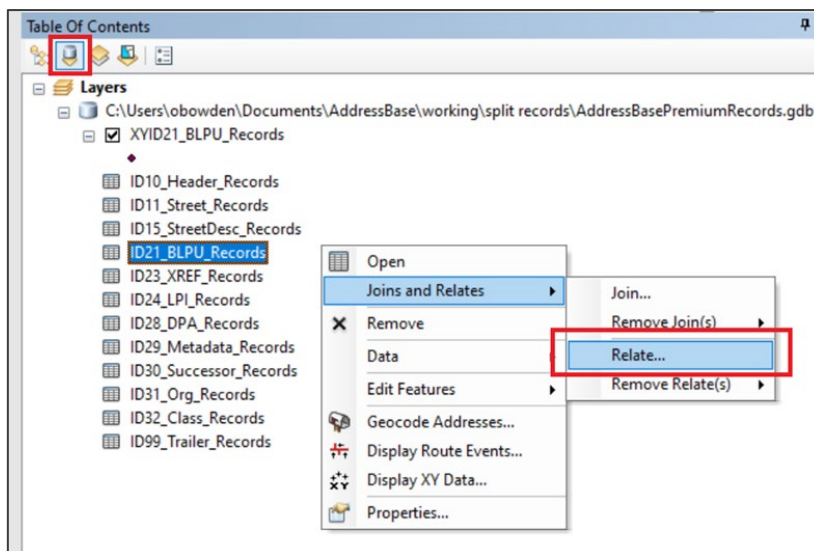
- USRN – Street (ID11_Street_Records) USRN

Street (ID11_Street_Records):

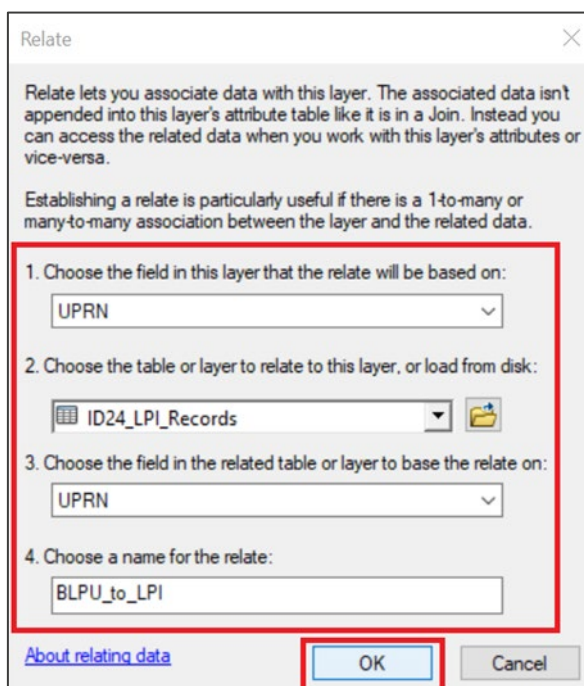
- USRN – Street Descriptor (ID15_StreetDesc_Records) USRN

27. To do this, select the *Source* tab in the left-hand navigation window and right-click on the first table that you wish to relate to another. To create the relevant *relates*, use the following instructions:

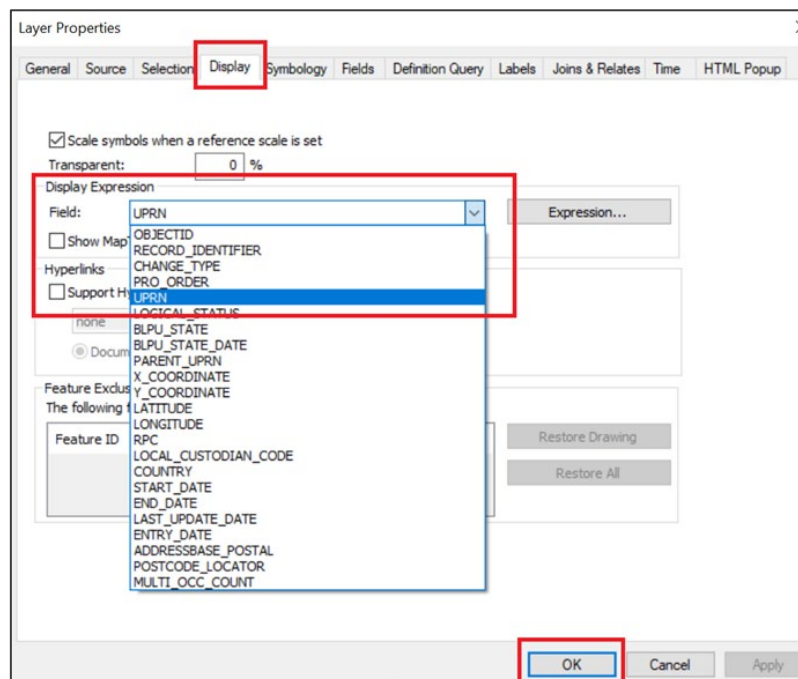
- a. Click *Joins and Relates > Relate...*



- b. From the first dropdown menu, select the attribute from the first table that will be used to create the *relate* between the two tables. (Apply the relationships as listed in step 26.)
- c. From the second dropdown, select the table that is going to be related to. (Apply the relationships as listed in step 26.)
- d. From the third dropdown, select the attribute from the table that is being related to. (Apply the relationships as listed in step 26.)
- e. In the fourth dropdown, input a relevant name for the relate (for example, *BLPU_to_LPI*).



- f. Click OK.
 - g. Repeat this process for all of the *joins / relates* listed in step 26.
28. Once the data has been loaded into ArcMap, you may wish to display more relevant information in the *Info* tool than the Esri-defined *Object ID*. To change this, use the following instructions:
- a. Double-click on the spatial dataset that you wish to change the *Display Expression* of.
 - b. Select the *Display* tab.
 - c. Change the *Field* to the desired field (for example, UPRN).
 - d. Click OK.



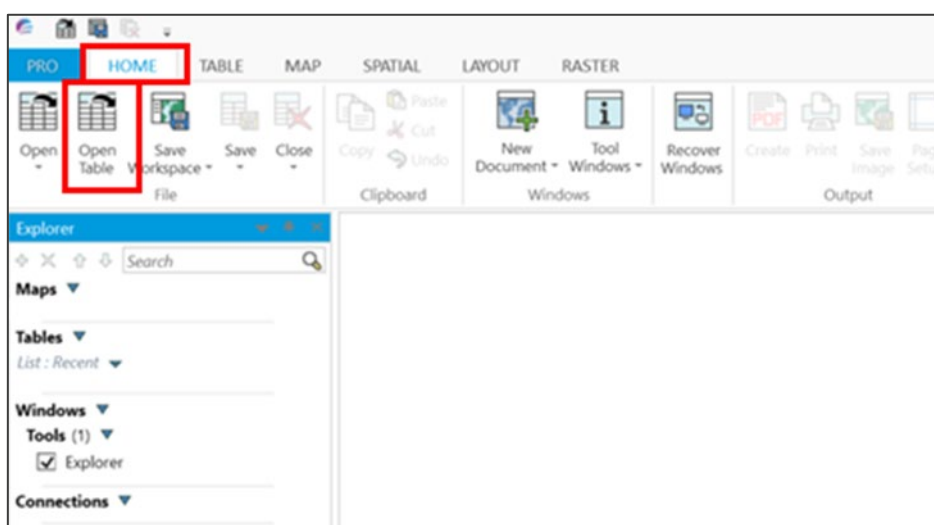
5.3 Loading CSV into MapInfo

The following step-by-step instructions are based on MapInfo Professional version 16.0.

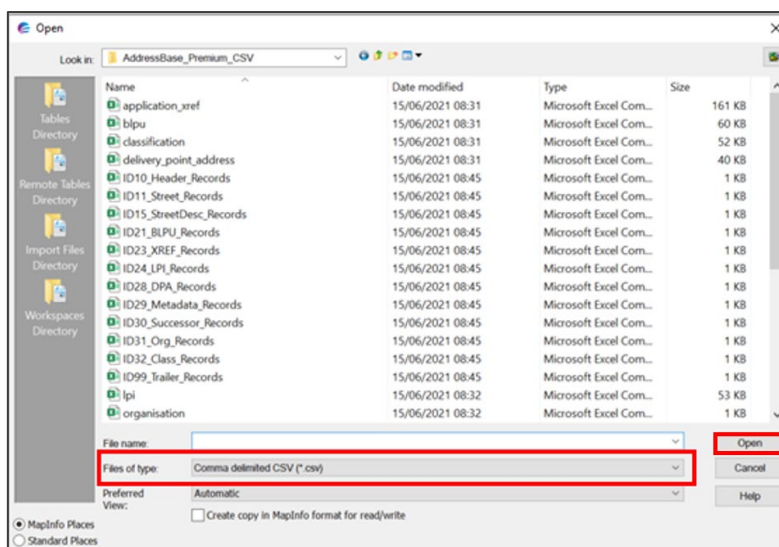
Note: MapInfo has a size limit of 2 Gb on each table. This equates to a maximum number of approximately 4 million AddressBase Premium records.

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 append the Header files can be found in [Section 4](#) of this guide.

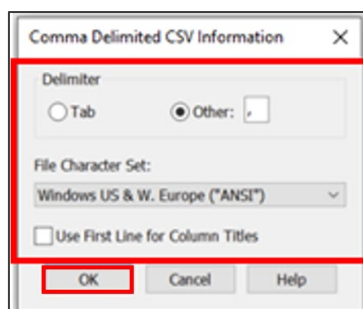
1. Launch MapInfo.
2. Click *Home > Open Table* and navigate to the folder that contains the AddressBase Premium data.



3. In the *Files of Type* dropdown menu, select *Comma delimited CSV (*.csv)*, then click on the AddressBase Premium data to be loaded. Click *Open*.

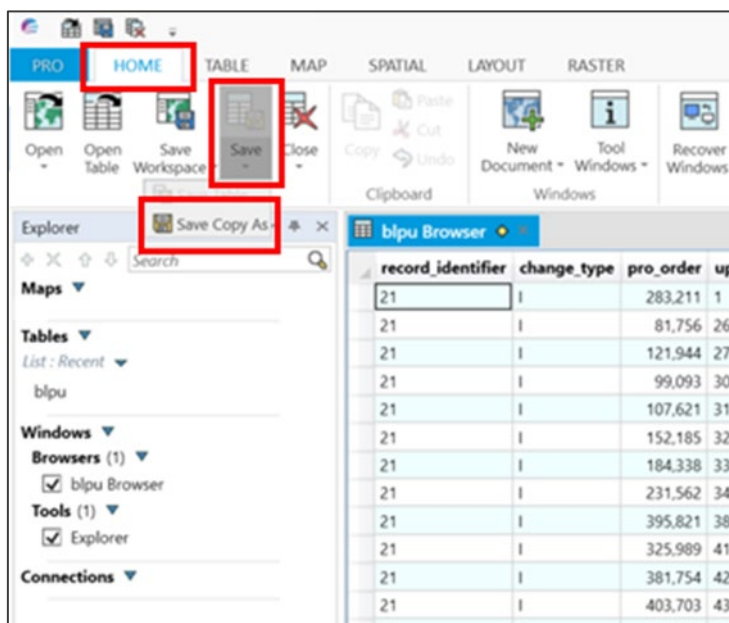


4. In the next window, tick the box *Use First Line for Column Titles* and select *Windows US & W. Europe* (“ANSI”) from the dropdown menu for File Character Set. Click OK.



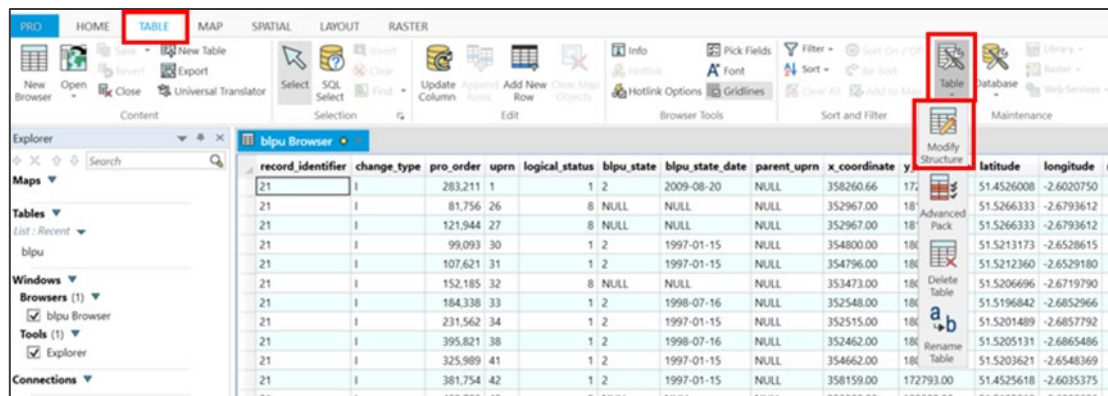
Note: When adding data this way, the field type classifications and field sizes of each column automatically try to fit the type of data that MapInfo believes is contained within the column and the largest value of that classification found within that column. This means the classifications and field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

5. Select *Home > Save > Save Copy As* and select the AddressBase table that was loaded in. Click *Save As...* and name the table to be created, then click *Save*.



6. Open the table that was just created via *Home > Open*. Navigate to and select the copy of the table you just named. Click *Open*.

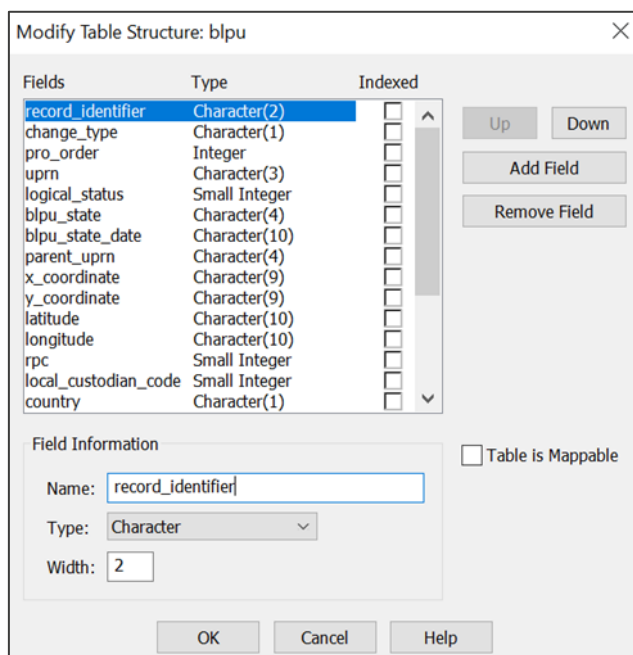
7. Navigate to **Table > Maintenance > Table > Modify Structure** and select the table to be edited. Click OK.



8. Here, you can change the *Type* and *Width* of each attribute to match the ones stated in the [AddressBase Premium Technical Specification](#).

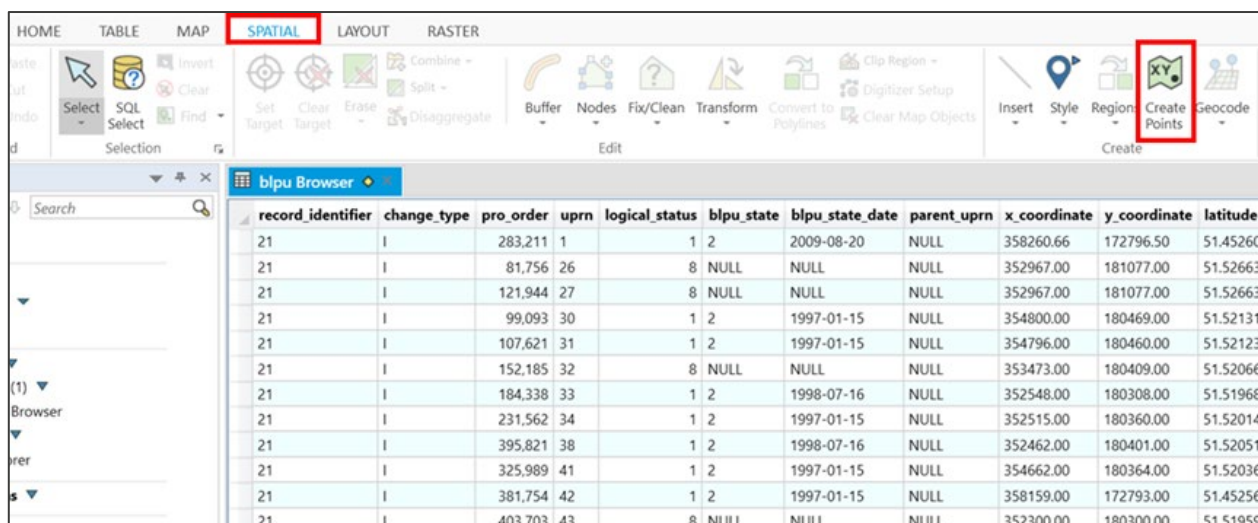
9. Type and Width should be changed for all attributes, apart from the following due to software-specific dependencies:

- *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*

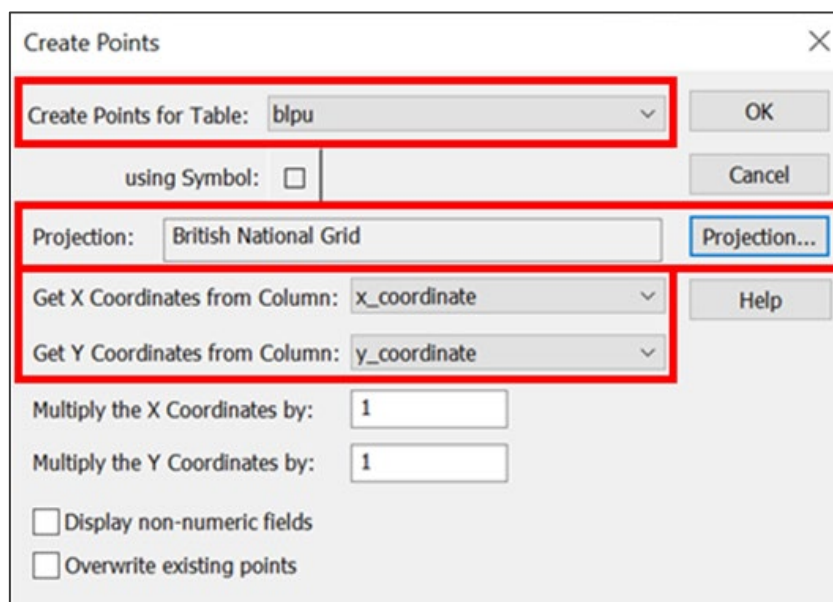


10. After all changes have been made, click OK.

11. To create a map of the location of the AddressBase Premium records, they need to be geocoded. Ensure the table of records that you wish to geocode is open, then navigate to *Spatial > Create Points*.

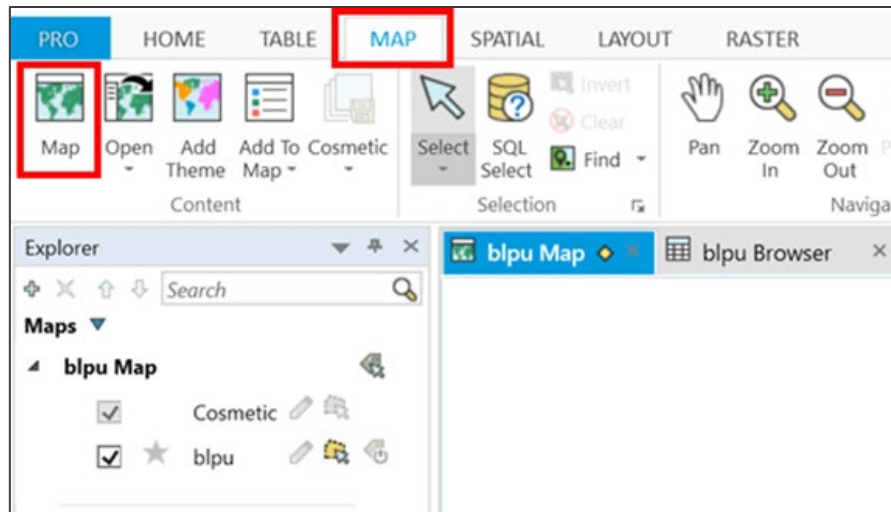


12. Select the table you wish to geocode from the *Create Points for Table* dropdown menu.
13. Expand the *Get X Coordinates from Column* dropdown menu and select either *X_Coordinate* or *Longitude*.
14. Expand the *Get Y Coordinates from Column* dropdown menu and select either *Y_Coordinate* or *Latitude*.
15. Click on the *Projection* button, then select the *British Coordinate Systems* option from the *Category* dropdown menu. Select *British National Grid [EPSG: 27700]*, or if you selected Longitude and Latitude in the steps above, then you should select *ETRS89 [EPSG: 4258]*.



16. Click OK to close that window and OK again to close the next window.

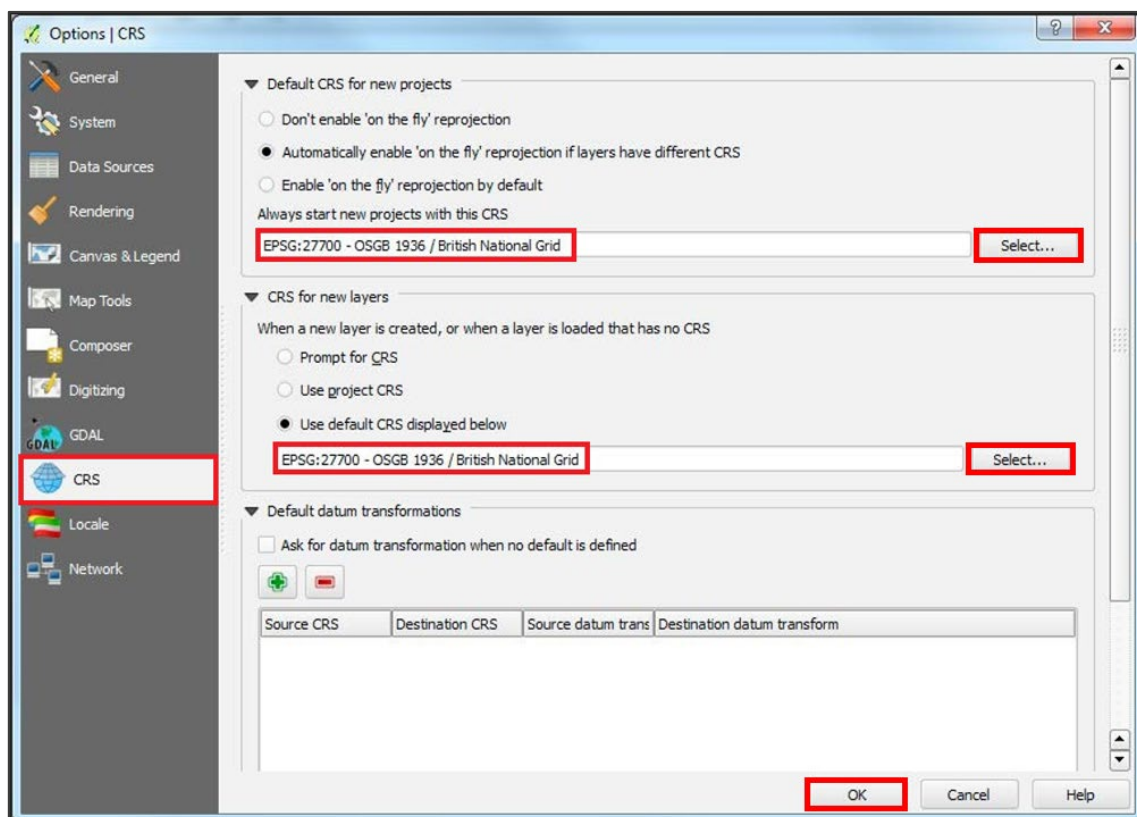
17. Finally, click *Map > Map (New Map Window)* to view the loaded geocoded points.



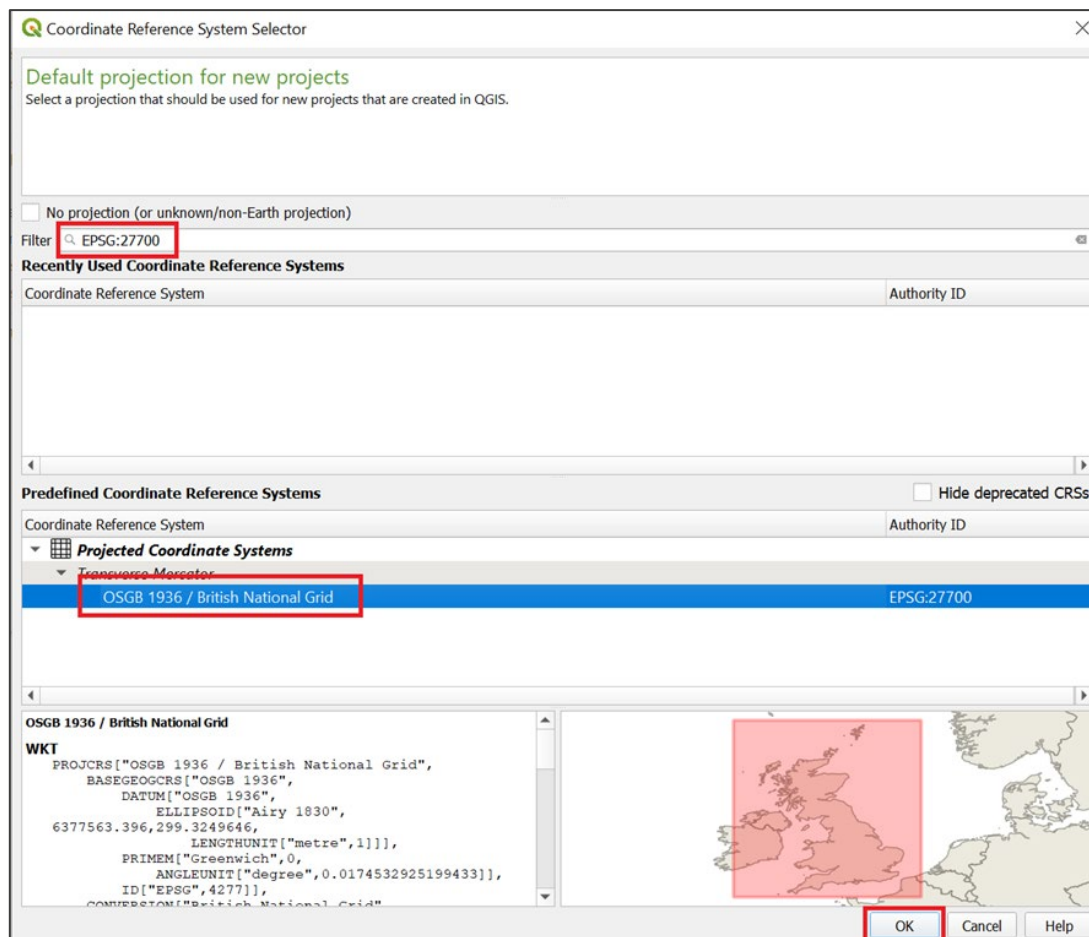
5.4 Loading CSV into QGIS

The following step-by-step instructions are based on QGIS version 3.1.

1. Launch QGIS and click *Settings > Options*.
2. Select *CRS* from the left-hand menu and check the coordinate reference system (CRS) is set to *British National Grid* within the *Default CRS for new projects* section and the *CRS for new layers* section.



If the coordinate reference system is not already set, click the *Select...* button in the bottom right-hand corner of each section. A new dialog box will appear. In the Coordinate Reference System Selector dialog box, 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]*. Click *OK* to close the Coordinate Reference System Selector dialog box.



3. Click *OK* to close the Options CRS dialog box.
4. As AddressBase Premium is made up of many record types, we need to make *joins* to view all the available data. The *joins* you need to make are given in the [AddressBase Premium Technical Specification](#), but for reference:

BLPU (ID21_BLPUs_Records):

- UPRN – Application Cross Reference (ID23_XREF_Records) UPRN
- UPRN – LPI (ID24_LPI_Records) UPRN
- UPRN – Delivery Point Address (ID28_DPA_Records) UPRN
- UPRN – Organisation (ID31_Org_Records) UPRN
- UPRN – Classification (ID32_Class_Records) UPRN

LPI (ID24_LPI_Records):

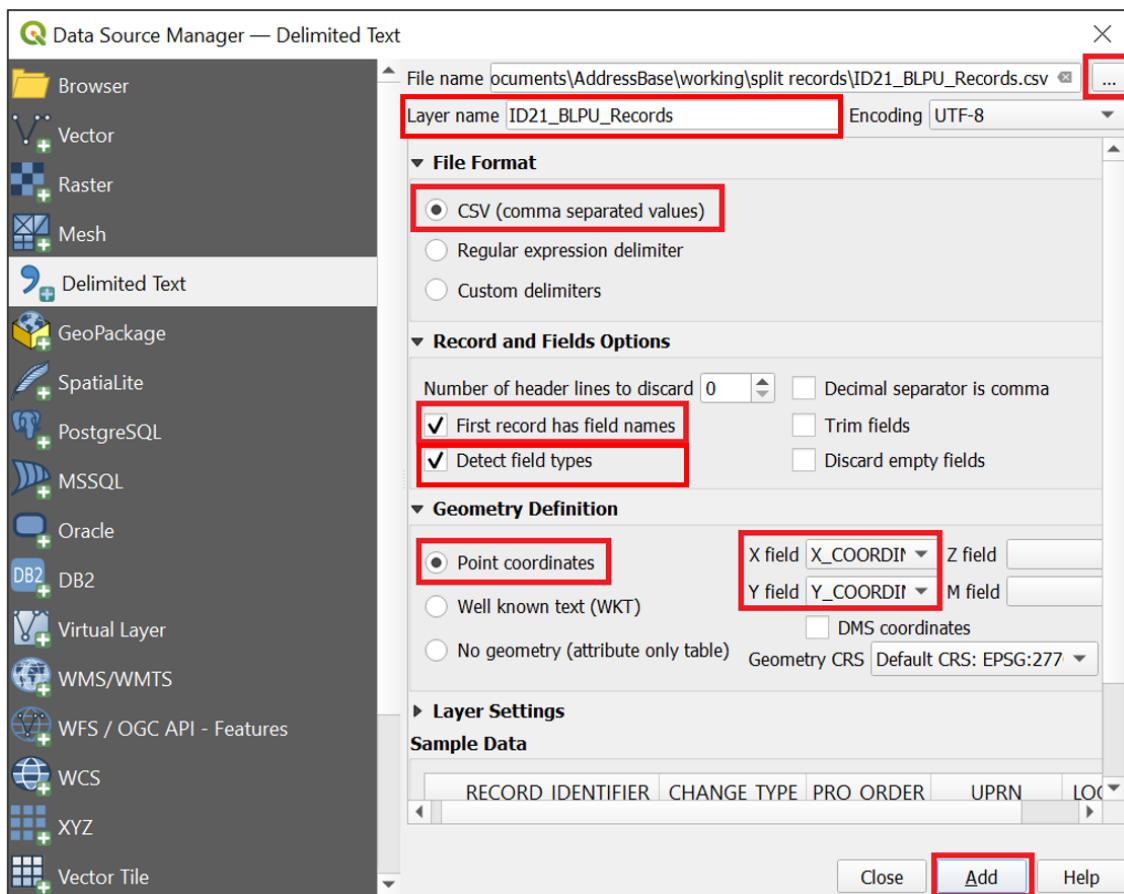
- USRN – Street (ID11_Street_Records) USRN

Street (ID11_Street_Records):

- USRN – Street Descriptor (ID15_StreetDesc_Records) USRN

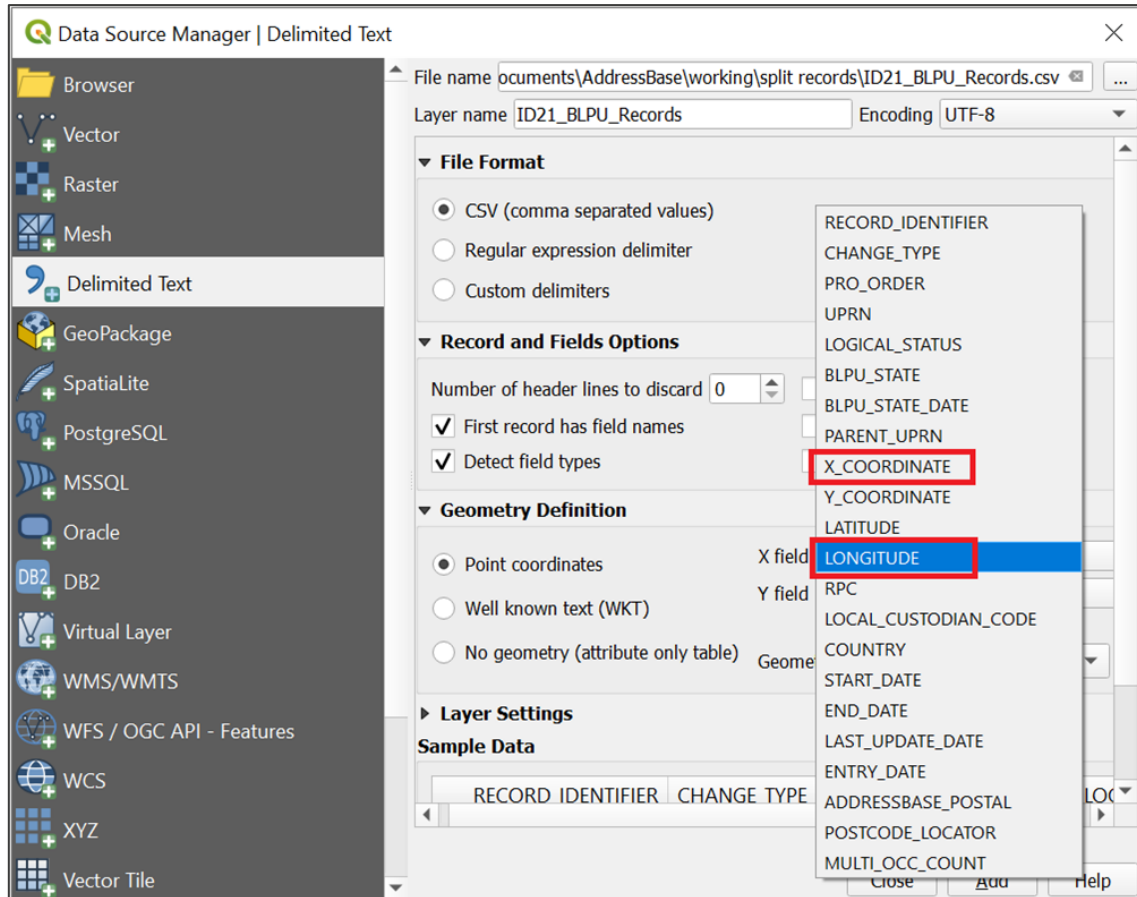
Note: The following steps explain how the joins are made for the first relationship given in the list above, i.e. between the BLPU (ID21_BLPUs_Records) and the Application Cross Reference (ID23_XREF_Records). This process will need to be repeated for all subsequent joins you make.

5. Back in the QGIS window, in the top ribbon, select **Layer > Add Layer > Add Delimited Text Layer**. The Data Source Manager – Delimited Text dialog box will appear.



6. Click the *three dots* button next to the *file name* box and locate the CSV file that was created in [Section 4](#), named (*ID21_BLPUs_Records*). Do not select any files with *Record* at the start of their name. Select the CSV file and click *Open*.
7. In the *Layer name* box, you can keep the default layer name or change it to a new one.
8. Ensure the *First record has field names* option is ticked.
9. Ensure the *Detect field types* option is ticked.
10. For *Geometry Definition*, select *Point coordinates*.

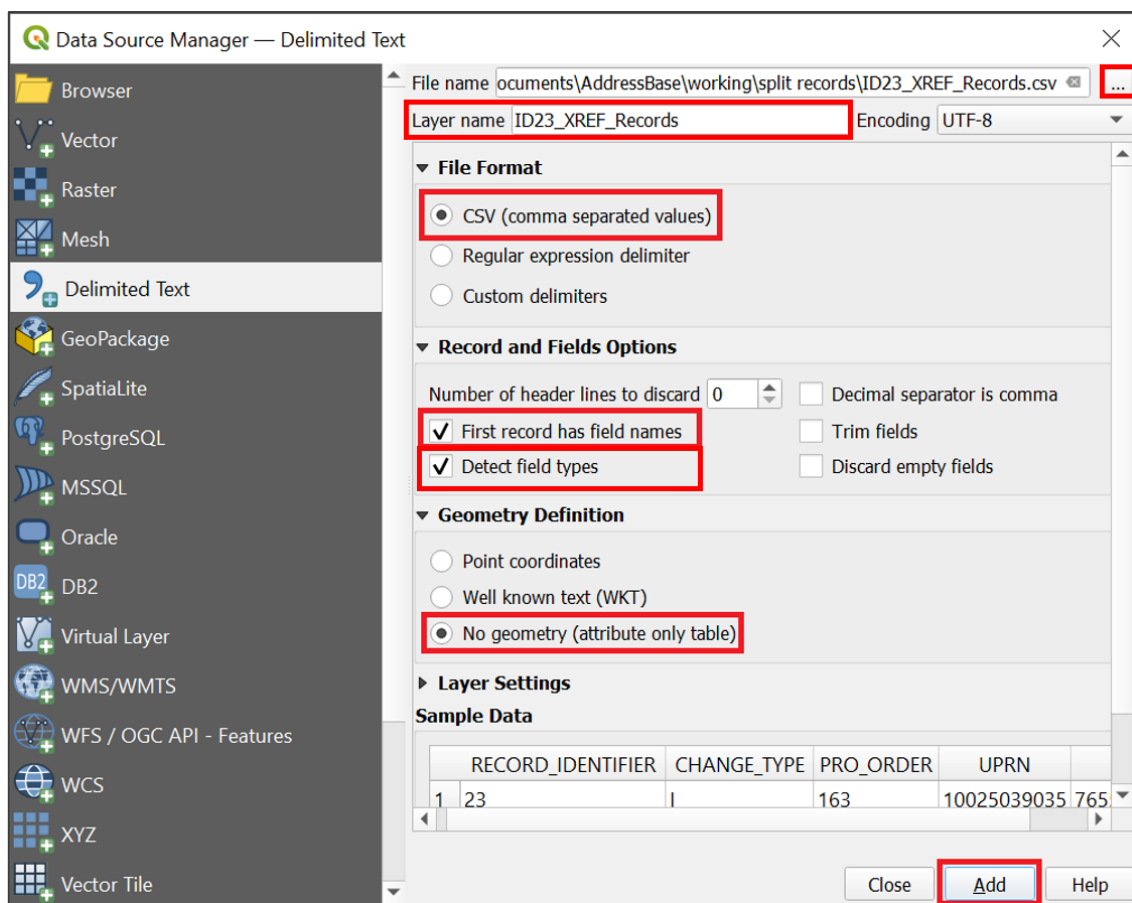
11. You should now be able to select the *X_Coordinate* field for the X Field dropdown and the *Y_Coordinate* for the Y Field dropdown if this was not done automatically. Alternatively, if you wish to use the Latitude and Longitude columns, *Longitude* needs to be inserted into X field dropdown, and *Latitude* needs to be inserted into the Y field dropdown.



12. Click **Add**.

So far, you have loaded the BLP_U record information.

13. In the QGIS window, in the top ribbon, select *Layer > Add Layer > Add Delimited Text Layer*. The Data Source Manager – Delimited Text dialog box will appear.

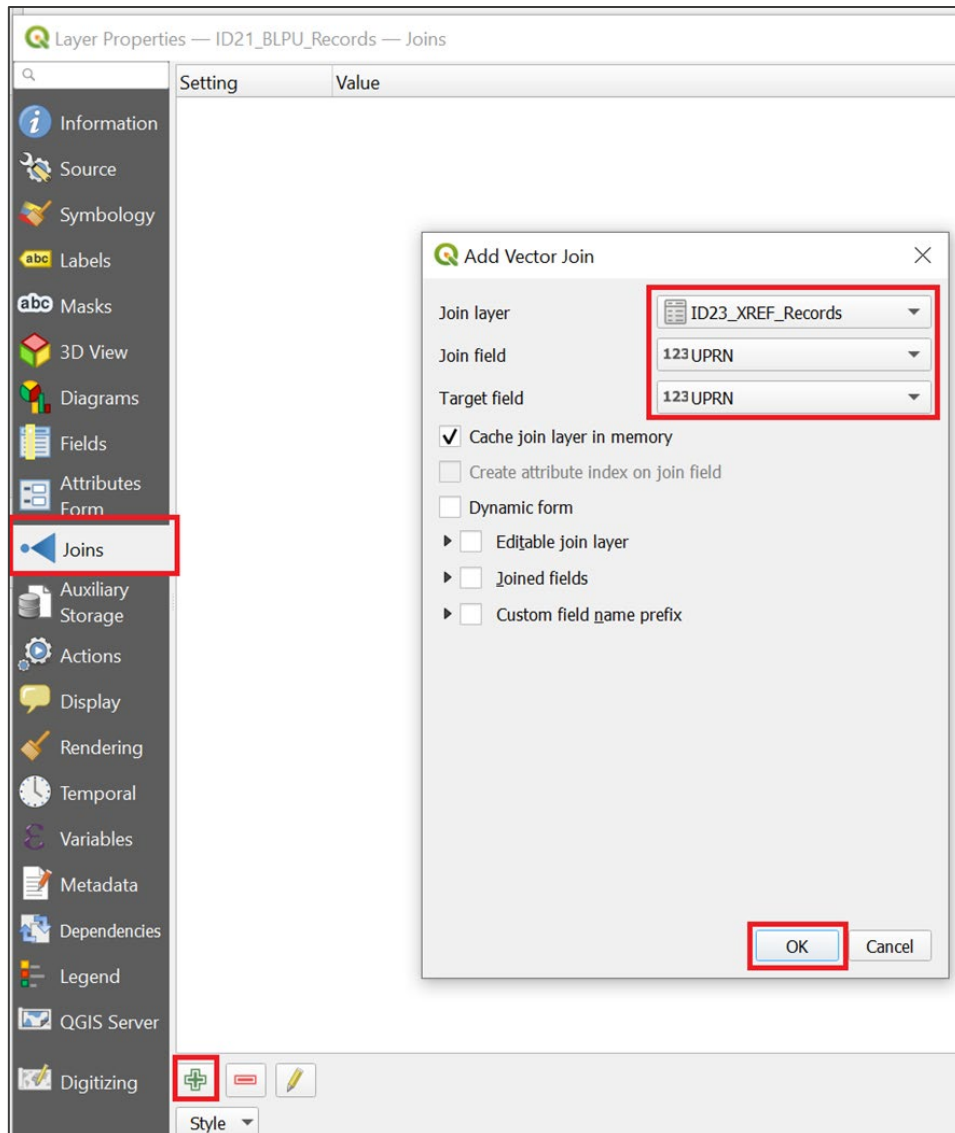


14. Click the *three dots* button next to the *file name* box and locate the CSV file that was created in [Section 4](#) named (*ID23_XREF_Records*). Do not select any files with *Record* at the start of their name. Select the CSV file and click *Open*.
15. In the *Layer name* box, you can keep the default layer name or change it to a new one.
16. Ensure the *First record has field names* option is ticked.
17. Ensure the *Detect field types* option is ticked.
18. For Geometry definition, choose *No Geometry (attribute only table)*.
19. Click *Add*.

This table will now be added to QGIS, but it will not be viewable as we have added it as an attribute table only.

20. Next, right-click on your BLPV layer and select *Properties*.

21. Go to the *Joins* tab found on the left-hand side.



22. Click the *green plus button* in the bottom left-hand corner.

23. Select your *Join layer*. For this example, it is: *Application Cross reference (ID23_XREF_RECORDS)*.

24. Select the *Join field*. For all BLPD links, this will be the *UPRN* (see the joins listed in step 4).

25. The *Target field* will also be the *UPRN* for this example.

26. Click *OK*.

You should now have a *join* listed in your *Joins* window.

27. Click *OK* to return to your main QGIS mapping screen.

28. If you now select one of your BLPU records in the main mapping window, you will see the BLPU attributes and the relevant Application Cross References for that record.

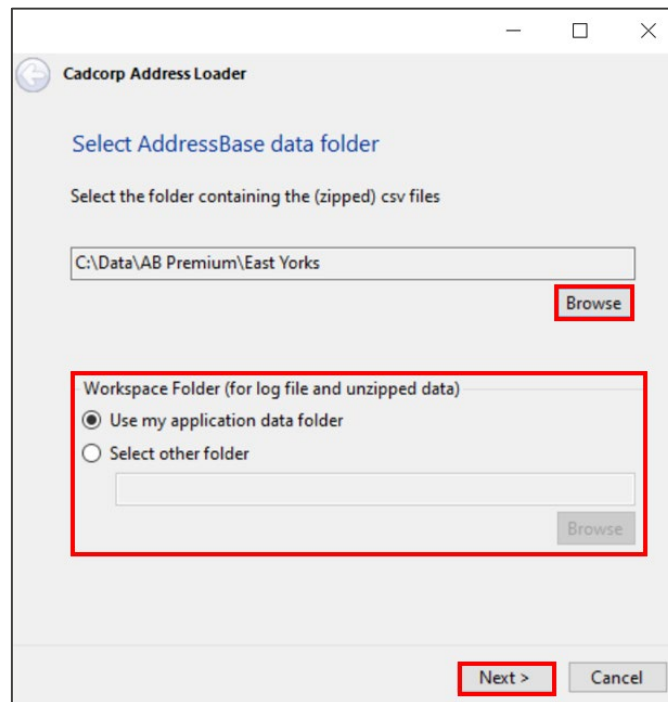
Identify Results	
Feature	Value
▼ ID21_BLPV_Records	
▼ RECORD_IDENTIFIER	21
▶ (Derived)	
▶ (Actions)	
RECORD_IDENTIFIER	21
CHANGE_TYPE	I
PRO_ORDER	48
UPRN	10025364089
LOGICAL_STATUS	1
BLPU_STATE	NULL
BLPU_STATE_DATE	NULL
PARENT_UPRN	NULL
X_COORDINATE	449994.8
Y_COORDINATE	1203624
LATITUDE	60.7126524
LONGITUDE	-1.0857674
RPC	1
LOCAL_CUSTODIAN_CODE	7655
COUNTRY	S
START_DATE	2011-07-29
END_DATE	NULL
LAST_UPDATE_DATE	2018-12-04
ENTRY_DATE	2011-07-29
ADDRESSBASE_POSTAL	N
POSTCODE_LOCATOR	ZE2 9DD
MULTI_OCC_COUNT	0
ID23_XREF_Records_RECORD...	23
ID23_XREF_Records_CHANGE...	I
ID23_XREF_Records_PRO_OR...	349
ID23_XREF_Records_XREF_KEY	7655X610380485
ID23_XREF_Records_CROSS_R...	osgb4000000003210879
ID23_XREF_Records_VERSION	3
ID23_XREF_Records_SOURCE	7666MI
ID23_XREF_Records_START_D...	2016-02-07
ID23_XREF_Records_END_DATE	NULL
ID23_XREF_Records_LAST_UP...	2018-12-05
ID23_XREF_Records_ENTRY_D...	2014-01-15

29. You can now repeat steps 4 to 28 for all the additional *joins* (listed in step 4) you wish to make.

5.5 Loading CSV using CadCorp Address Loader

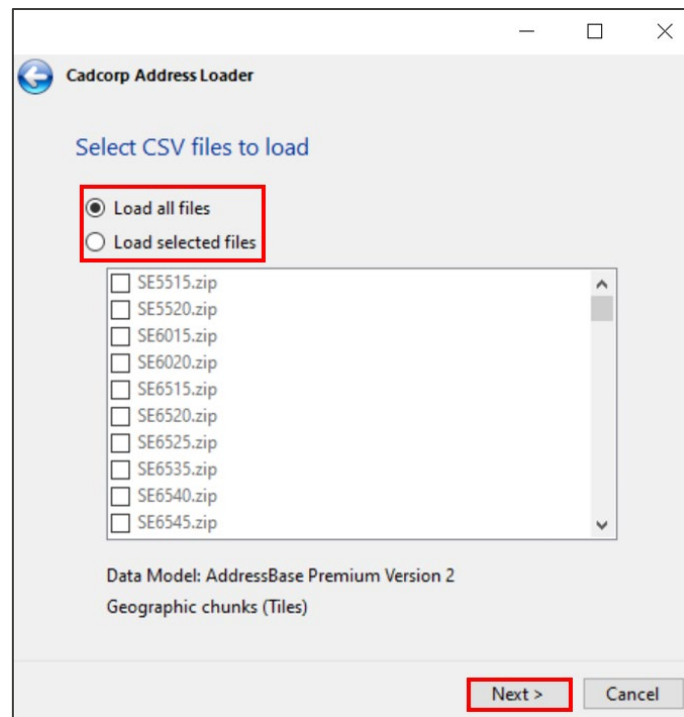
This sub-section details how to load AddressBase Premium CSV into a database using the CadCorp Address Loader. The CadCorp Address Loader requires a separate license to the CadCorp SIS Desktop version.

1. Open CadCorp Address Loader. A dialog box will appear:

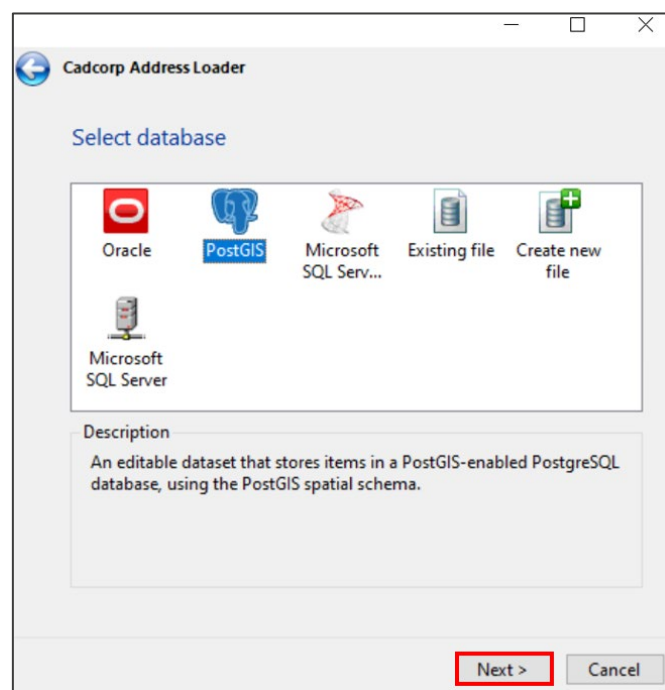


2. Click the *Browse* button and navigate to the location of the AddressBase Premium CSV file (zipped or unzipped) that you want to load into CadCorp.
3. In the Workspace Folder section, select a folder in which to store the log file and unzipped data. This can either be a folder of your choosing or the application data folder.
4. Click *Next*.

- Another dialog box will appear where you can select which address files to load. You can choose to load all or selected files.



- Click Next.
- The next dialog box to appear will allow you to select a database into which you can save the AddressBase Premium data. Select your database of choice, then click Next. (Note: The following instructions will relate to a PostGIS database).



8. In the next dialog box, connect to an existing database by entering information into the following fields:

- Host
- Port Number
- Database
- User name
- Password

Note: These details can be found within the parameters of your database. It is not possible to create a database via this method, and one will have had to have been created prior to this stage.

Cadcorp Address Loader

Connect to PostGIS

Host: localhost

Port number: 5432

Database: Cadcorp

User name: postgres

Password:

☐ Save password (encrypted)

Transaction size: 10000

Next > Cancel

9. Click Next.

10. A schema dialog box will appear showing the tables that the data will be loaded into:

Cadcorp Address Loader

AddressBase Premium Schema

AddressBase data will be loaded into the tables below:

☐ Prefix table names with:

application_xref
blpu
classification
dpaddress
lpi
metadata
organisation
street

(Append to existing tables)

☐ Drop and rebuild indexes ☐ Build interface table Options

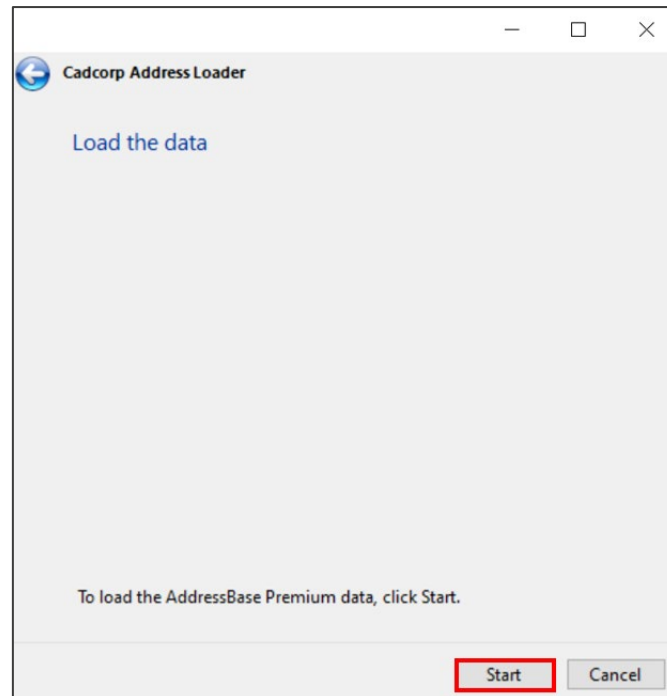
☒ Create PAF gazetteer view: view_paf_gaz

☒ Create LPI gazetteer view: view_lpi_gaz

☐ Truncate tables before load

Next > Cancel

11. At this stage, it is possible to create a gazetteer view or an interface table where a concatenated LPI address field can be built.
12. Click *Next* in that dialog box, then *Start* in the following dialog box to begin loading the data:



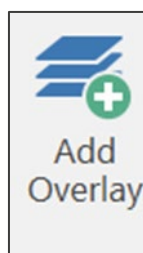
13. Once finished, the tables will be loaded into the database.

At this stage, it is possible to load the spatial tables (BLPU and Streets) into CadCorp. You can join these tables together within your chosen database to then add all data into CadCorp. Please refer to the [AddressBase Premium Technical Specification](#) to gain a better understanding of the relationships between AddressBase Premium tables.

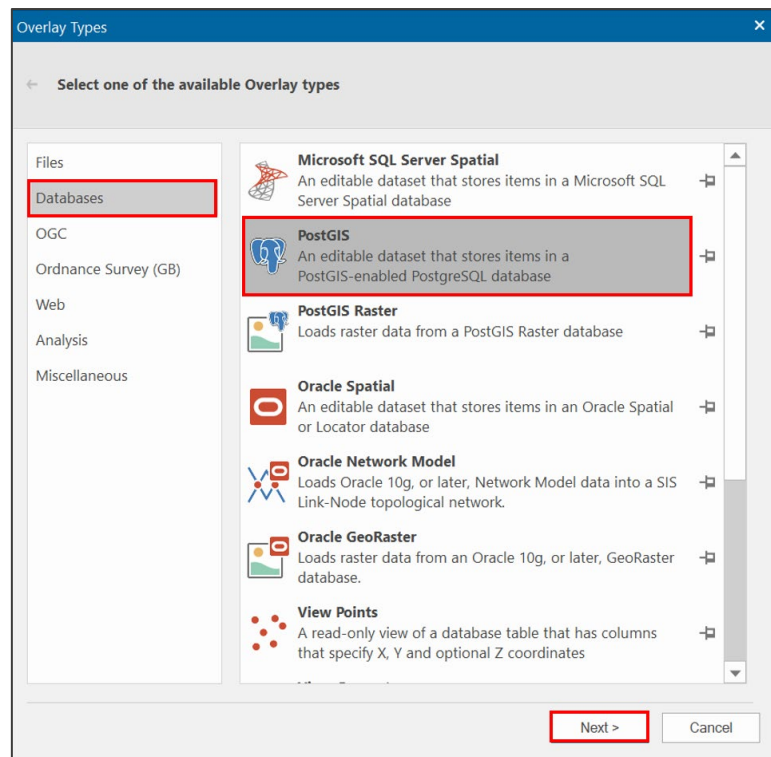
5.5.1 Loading CSV into CadCorp SIS Desktop via a database

Following on from the instructions in sub-section 5.5, use the following instructions to load AddressBase Premium into CadCorp SIS Desktop via a database:

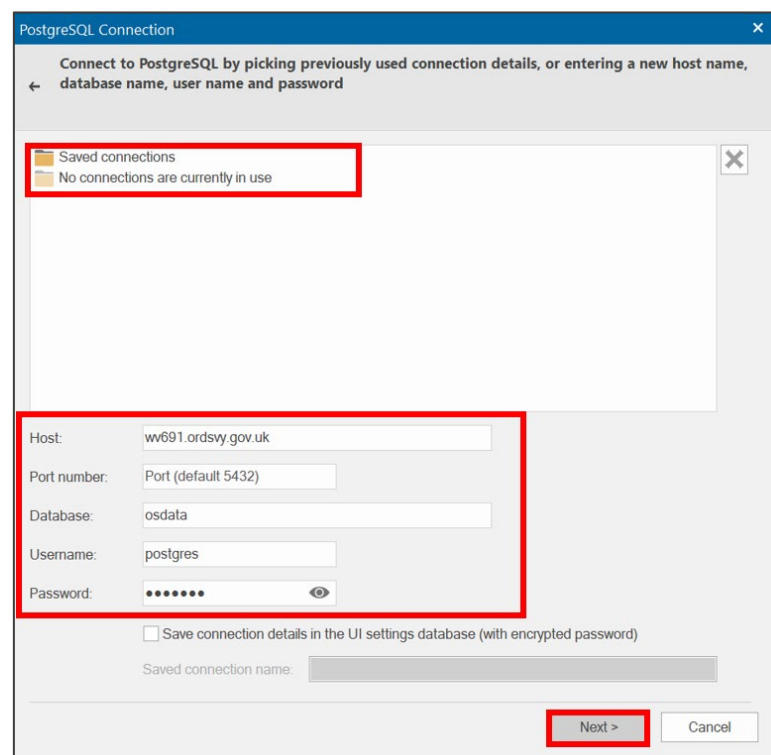
1. Open a new or existing CadCorp SIS Desktop project.
2. In the top ribbon, select *Add Overlay*.



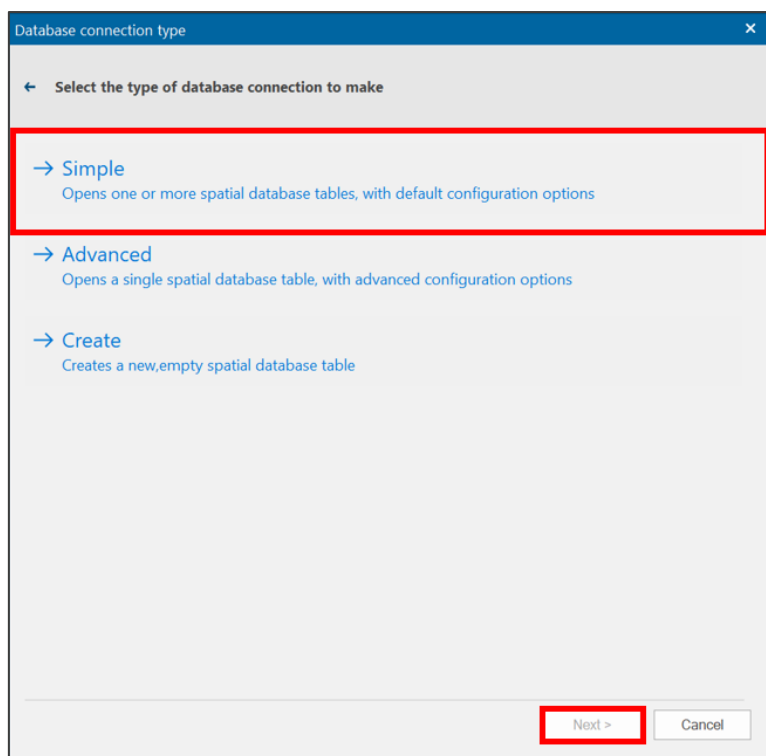
- On the left, select *Databases*, then select your database of choice from the list on the right (Note: A *PostGIS* database will be referred to in the instructions from this point on).



- Click *Next*.
- Connect to your existing database. To connect to a saved database, select a database under *Saved connections*. Alternatively, you can manually enter the database details. Click *Next*.



6. In the next dialog box, select the type of database connection you want to make. The *Simple* connection option is effective in this instance. Click *Next*.



7. Select the tables to load into CadCorp, then click *Finish*.

6. Loading CSV into a database

This section provides step-by-step instructions on how to load the CSV format of AddressBase Premium products into some commonly used databases, including PostgreSQL, Oracle and Microsoft SQL Server 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 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 Primary Keys on all tables (for example, UPRN on the BLPU table) are valid upon a data load. If a Delete is issued for a Primary Key, this doesn't mean that Primary Key will not reappear in subsequent supplies.

There are a number of reasons this may happen:

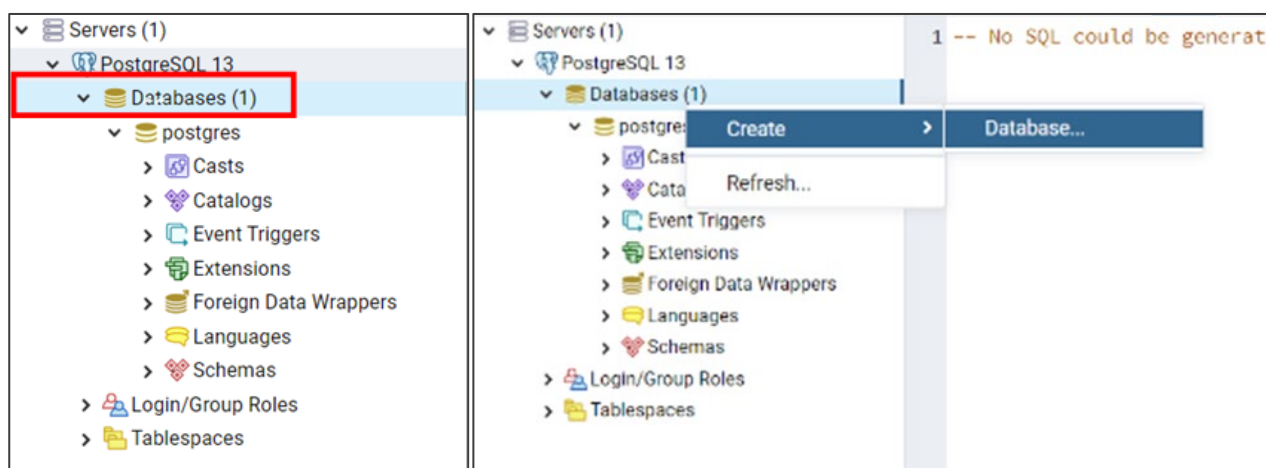
- The record has moved in location more than once, moving it out of your AOI (therefore, the record is deleted) but then back into your AOI in the future. This would also occur if you altered your AOI.
- A record has failed data validation upon a change being made. This can result dependent on the change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

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

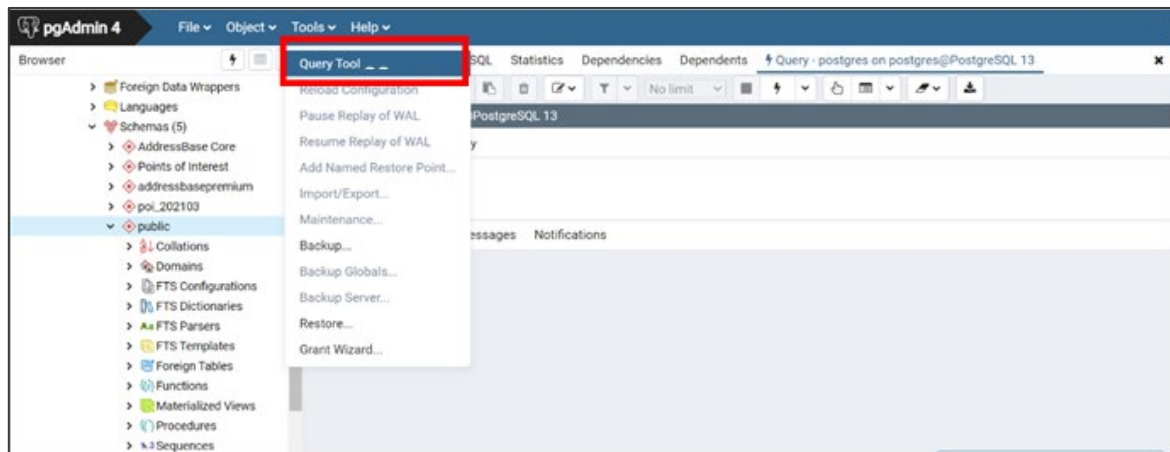
6.2 Loading CSV into a PostgreSQL database

The following step-by-step instructions describe how to load AddressBase Premium into a PostgreSQL database using the text files created using the CSV file merge utility, as described in [Section 4](#). The instructions are based on PostgreSQL version 1.12.3 and assume that you have set-up your database with the PostGIS spatial extension. It is recommended that you have a basic understanding of database terminology before following these instructions.

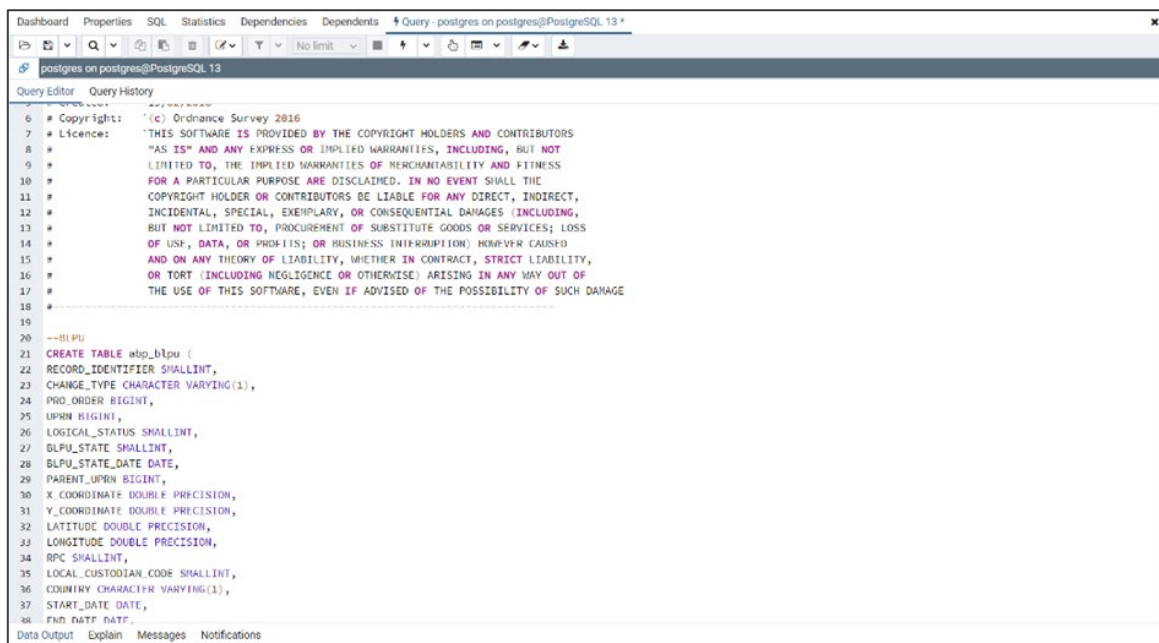
1. Prepare the text files as described in [Section 4](#) of this guide.
2. Check that there are no carriage returns (extra rows) at the end of the CSV output file as this will result in errors. To check this, open the CSV file and hit *End* on your keyboard. Your cursor should now be at the end of the 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 on the *Windows Start Menu* > *pgAdmin*).
4. On the left-hand side, under *Browser*, you have the option to connect to either your existing databases or a new one. To connect to a new database, right-click on *Databases*, then select *Create* > *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. Download the SQL file in the [GitHub AddressBase_Premium_and_Islands folder](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 can be copied and pasted into the SQL query tool within PostgreSQL. To open a new SQL file, select *Tools* in the ribbon, then select *Query Tool* from the dropdown.



6. Copy and paste the SQL code from the GitHub link given in step 5b into the SQL Query Tool (as shown in the following screenshot).



7. The following tables should be created (you can alter table names as you wish):

- BLP
- Classification
- Cross reference
- Delivery Point Address
- LPI
- Organisation
- Streets
- Street Descriptor
- Successor Records

8. 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.

Note: The path and file name may need to be changed to reflect your data:

```
COPY abp_blpu FROM 'C:/Address/ID21_BLP_Records.csv' DELIMITER ';' CSV HEADER;  
COPY abp_delivery_point FROM 'C:/Address/ID28_DPA_Records.csv' DELIMITER ';' CSV  
HEADER;  
COPY abp_lpi FROM 'C:/Address/ID24_LPI_Records.csv' DELIMITER ';' CSV HEADER;  
COPY abp_crossref FROM 'C:/Address/ID23_XREF_Records.csv' DELIMITER ';' CSV HEADER;  
COPY abp_classification FROM 'C:/Address/ID32_Class_Records.csv' DELIMITER ';' CSV  
HEADER;  
COPY abp_street FROM 'C:/Address/ID11_Street_Records.csv' DELIMITER ';' CSV HEADER;  
COPY abp_street_descriptor FROM 'C:/Address/ID15_StreetDesc_Records.csv' DELIMITER ';' CSV  
HEADER;  
COPY abp_organisation FROM 'C:/Address/ID31_Org_Records.csv' DELIMITER ';' CSV HEADER;  
COPY abp_successor FROM 'C:/Address/ID30_Successor_Records.csv' DELIMITER ';' CSV  
HEADER;
```

9. Once loaded, you may want to add Primary and Foreign Keys to the data. 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. For the BLP table, the UPRN provides a unique value. 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 + symbol to add a new Primary Key.




- d. Click the *edit button*.
- 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*.

10. Repeat the procedure for the Streets table and USRN.

However, in the other tables these columns may contain duplicate values. In this case, use the table key (for example, *LPI_Key* as the primary identifier). Alternative Object Identifiers (OID) can be added to each table (these are also required to use the data in some GIS, including QGIS and MapInfo.) The following SQL can be used for this:

```
ALTER TABLE insert_table_name SET WITH OIDS
```

11. To help performance when querying across multiple tables, a Foreign Key may be added. A list of the Foreign Keys with AddressBase Premium can be found in Section 2 of the [AddressBase Premium Technical Specification](#). However, as with a Primary Key, only unique data columns can be used.
 - a. Right-click on the table name and select *Properties*.
 - b. Select the *Constraints* tab.
 - c. Click the + symbol to add a new Foreign Key.
 - d. Click the edit button  .
 - 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*.
12. You can also index the data by using the following steps:
 - a. Right-click on the table name and select *Create > Index*.
 - b. Under the General tab, enter a name (for example, *Idx1*).
 - c. Under *Definition tab > Columns*, click the + symbol.
 - 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 Premium 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 to make the data usable in a GIS:

```
ALTER TABLE public.abp_blpu ADD COLUMN geom geometry(Point, 27700);
```

2. Load the data into your new geometry column:

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

This sets the geom column in the BLPU 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.abp_blpu 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

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

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

The following steps describe one method for loading a full supply of the data. Sections of text within <...> (a pair of less than and greater than symbols) 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; therefore, you will need in the region of 40 Gb of free space to load AddressBase Premium and 1 Gb of free space to load AddressBase Premium Islands.
2. Once the data is copied, the next stage is to unzip the *.zip files to extract the *.csv files. This can be done using a package such as Winzip or 7Zip. Please see [Section 3.3](#) for more information.
3. Now that all the files are unzipped, the latter stages will be easier if you create a file listing 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. Copy the following into a text editor and save it as a .bat file in the same directory as the AddressBase Premium data:

```
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 *AddressBase_Premium_and_Islands* folder. Open the file ending *CreateTables.sql* in a text editor.
6. Within that SQL there are references to <TablespaceName>, which need to be changed to your tablespace name that is being worked in. When these are changed, copy and paste the SQL into Oracle to create the tables.
7. Next, a SQLLDR control file needs to be created. An example of one of these files is *Oracle_AddressBasePremium_Control.ctf*, provided in the *AddressBase_Premium_and_Islands* folder in the OS GitHub repository linked in step 4. Open this file in a text editor.
8. Populate the INFILE lines with the file listing that was created in step 3; use one INFILE command for each file. This tells the process to open each of the files and carry out the other tasks listed below it.
9. The rest of the file tells the tool how to interpret the files that it is reading in. The INTO statement at the top of each of the tables tests the first column (01) of the row in the file that it is looking at. If it meets the criteria, the structure of the table that the line is to be loaded into is described below it. Save the completed file with the extension *.ctl.
10. 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
```

11. When the .bat file has been run, the data is loaded. Errors with the load and / or any records that do not meet the expected structure are recorded in the and *.log and *.bad files, respectively. These are written out to the same drive location as the control file that is being used to load the data.

It is strongly recommended that the log file is checked once the load is completed to verify that all of the data has loaded correctly before continuing.

12. After loading, the indexes need to be built in order to be able to carry out spatial queries and other queries where the relationship between the tables need to be built.

For example, in order to return all the Delivery Point Addresses within a county, there needs to be a spatial index on the BLPU table which contains the geometry, as well as the UPRN in both the BLPU and Delivery Point Address tables. The SQL statements to create the indexes can be found in the GitHub repository link that was referenced in step 4.

13. Again, you can copy and paste the SQL statements from a text editor into Oracle in order to create the rest of the indexes.

Note: The table name provided may be different to yours and therefore might need changing before use.

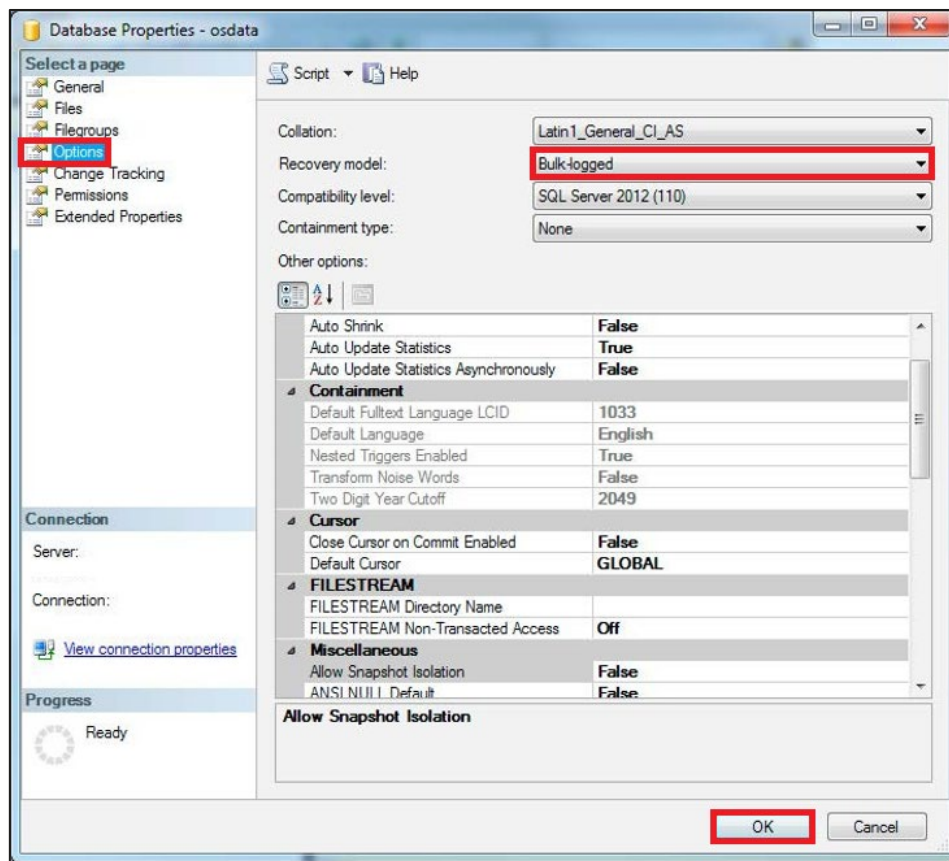
14. Once the indexes are complete, the data loading process is complete and the data is ready to use.

6.4 Loading CSV into a Microsoft SQL Server

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

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 menu.
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 20 Gb which, in turn, can cause issues with loading. Click *OK*.



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 select your CSV file. Each table needs to be added separately (for example, *ID21_BLPU_Records.csv*).

Note: 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 document. Ensure that *Column names in the first data row* is ticked.
12. Check that the *Text Qualifier* is set to a double quote ("). This ensures 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 that 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 AddressBase Premium [Technical Specification](#).
16. Once you have changed the column types to match 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 Premium, renaming to *[dbo].[BLPU_TABLE]*.
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 column which needs to be the Primary Key of the table. The Primary Keys for each table can be found in the AddressBase Premium [Technical Specification](#).
21. Click *Next*. On this screen, you can check that the *Source column* and *Destination column* 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 and Foreign Keys

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

Primary Key

```
alter table dbo.ID21_BLPU_Records  
add primary key (UPRN);
```

Foreign Key

```
alter table dbo.ID32_Class_Records  
add foreign 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.ID21_BLPU_Records  
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. Accessing GeoPackage data via GIS software

This section provides step-by-step instructions on how to access the GeoPackage format of AddressBase Premium products via commonly used GIS software, including ArcGIS Pro, ArcGIS Desktop, and QGIS.

GeoPackage (.gpkg) is an open, non-proprietary, platform-independent and standards-based data format for geographic information systems (GIS), as defined by the Open Geospatial Consortium (OGC). It is designed to be a lightweight format that can contain large amounts of varied and complex data in a single, easy to distribute and ready to use file. GeoPackage is natively supported by numerous software applications.

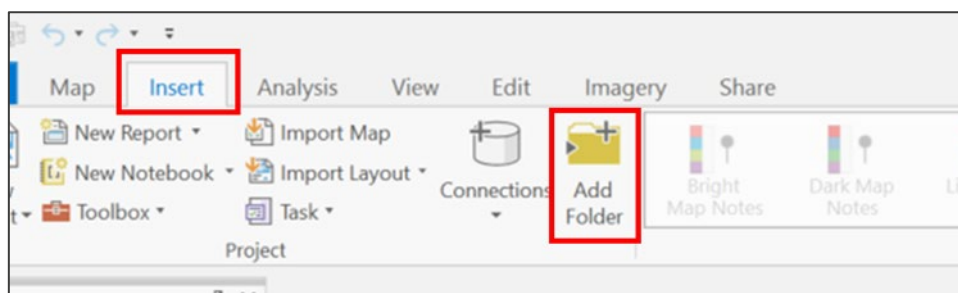
Note: The relational nature of AddressBase Premium has meant that loading GeoPackage into certain GIS is not possible at the time of this document's release.

7.1 Accessing GeoPackage data via ArcGIS Pro

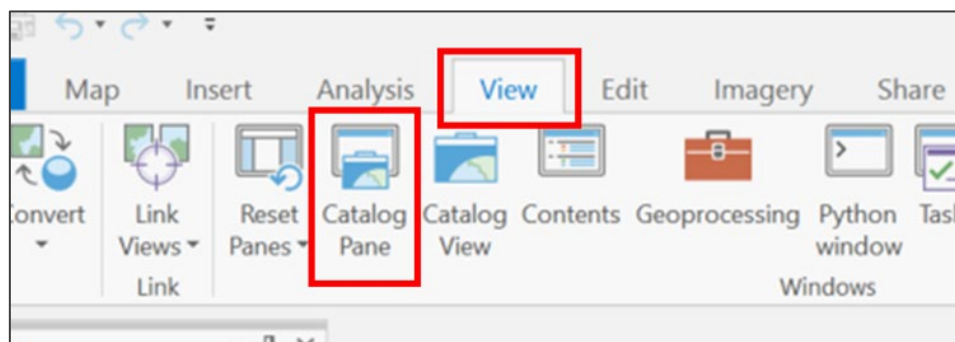
The following instructions are based on ArcGIS Pro version 2.7.1. It is recommended that you have a basic understanding of ArcGIS Pro tools and terminology before following these instructions.

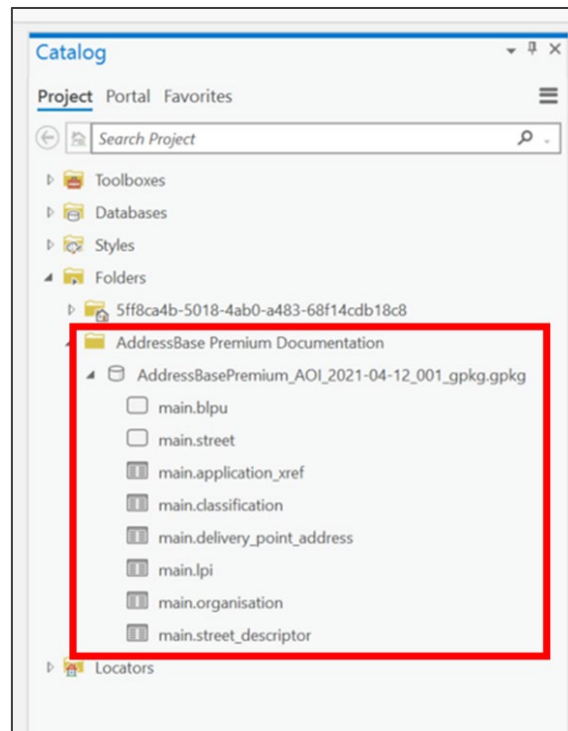
Note: The file must have the file extension .gpkg (not case sensitive) to recognise it as an OGC GeoPackage. Make sure the AddressBase Premium GeoPackage is stored in a suitable location and the file is unzipped.

1. Launch ArcGIS Pro.
2. To establish a connection to the folder where the GeoPackage file is stored, navigate to the *Insert* tab and click *Add Folder*. Find and select the folder, then click OK.

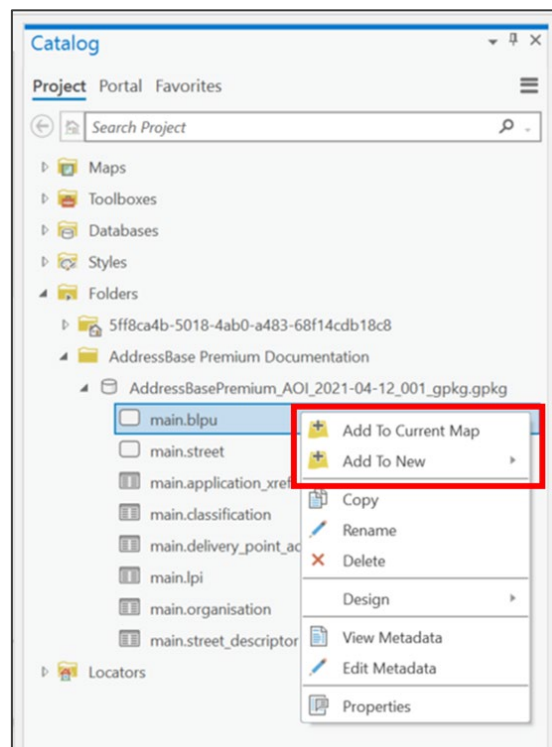


3. Navigate to the *View* tab and open the *Catalog Pane*. In the pane that appears, find the now-connected folder where the GeoPackage is stored, then expand the GeoPackage to show the available layers.

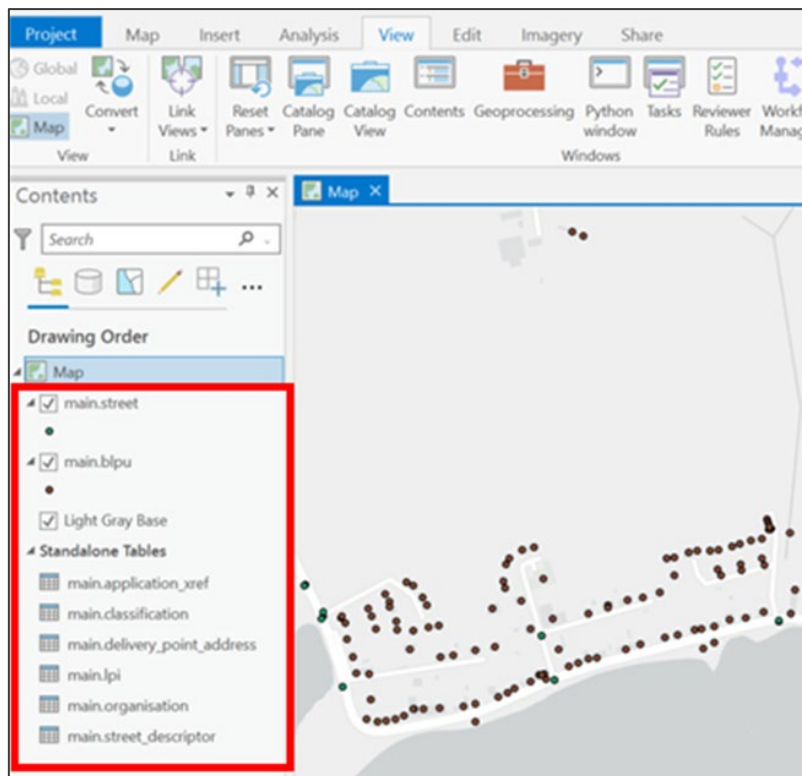




4. Right-click the *main.blpu* features class and click *Add To Current Map* (or *Add To New* if a map frame has not yet been created for this project). Repeat this step for all of the layers in the GeoPackage.



5. Note that the other tables can be added to the map frame, but they will only act as standalone tables as they have no spatial element to visualise. The BLPU and Streets tables will be visible in the Current Map view as they contain geometry. Once all of the tables have been added to the map frame, it should look similar to the following screenshot:



6. These standalone tables can be related to the spatial tables in order to give them a spatial component. The following relates are required:

BLPU (ID21_BLPUR_Records):

- UPRN – Application Cross Reference (ID23_XREF_Records) UPRN
- UPRN – LPI (ID24_LPI_Records) UPRN
- UPRN – Delivery Point Address (ID28_DPA_Records) UPRN
- UPRN – Organisation (ID31_Org_Records) UPRN
- UPRN – Classification (ID32_Class_Records) UPRN

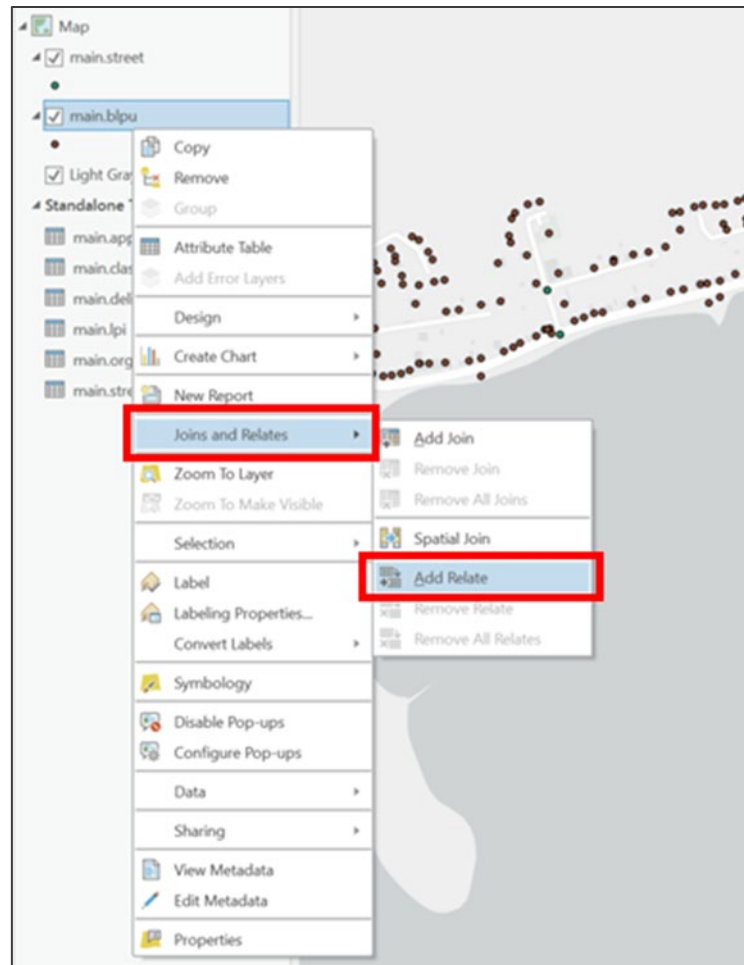
LPI (ID24_LPI_Records):

- USRN – Street (ID11_Street_Records) USRN

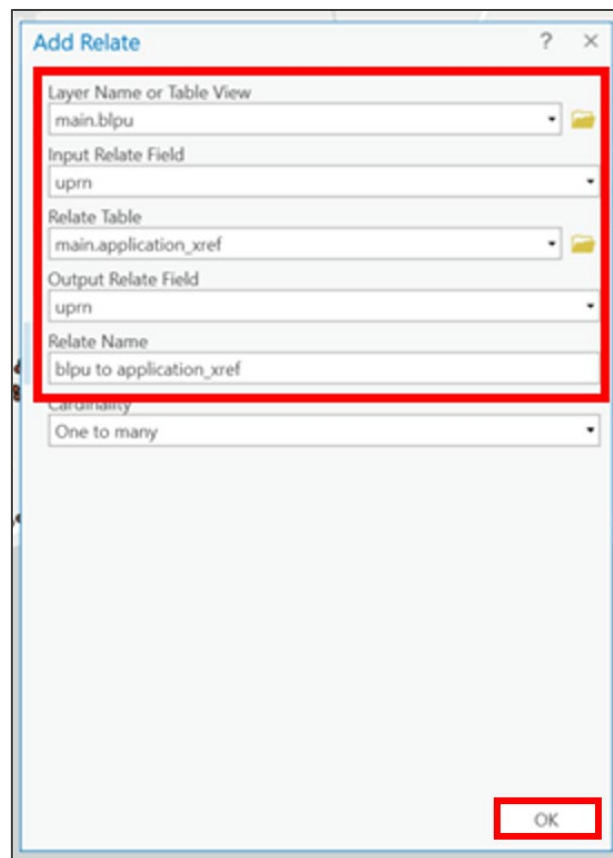
Street (ID11_Street_Records):

- USRN – Street Descriptor (ID15_StreetDesc_Records) USRN

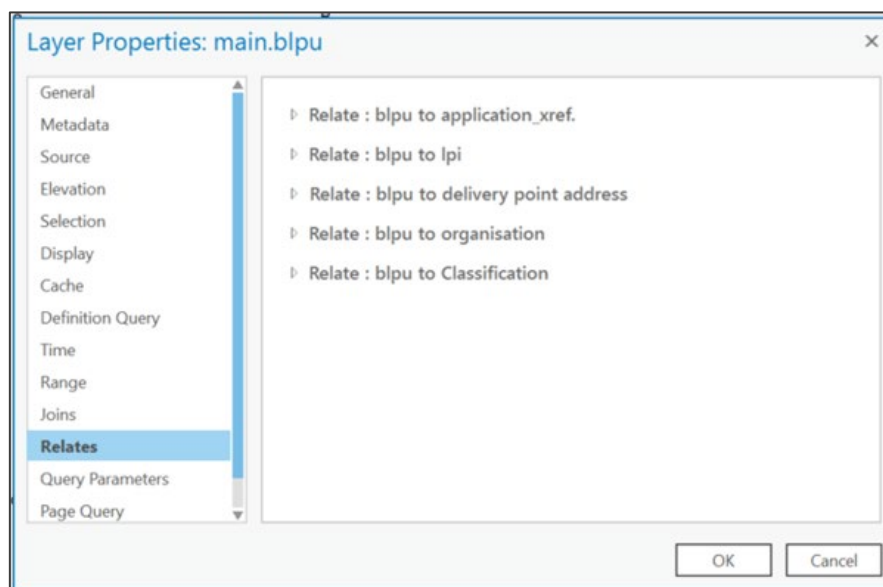
7. To perform a relate, right-click on the *main.blpu* feature class, then click *Joins and Relates > Add Relate*.



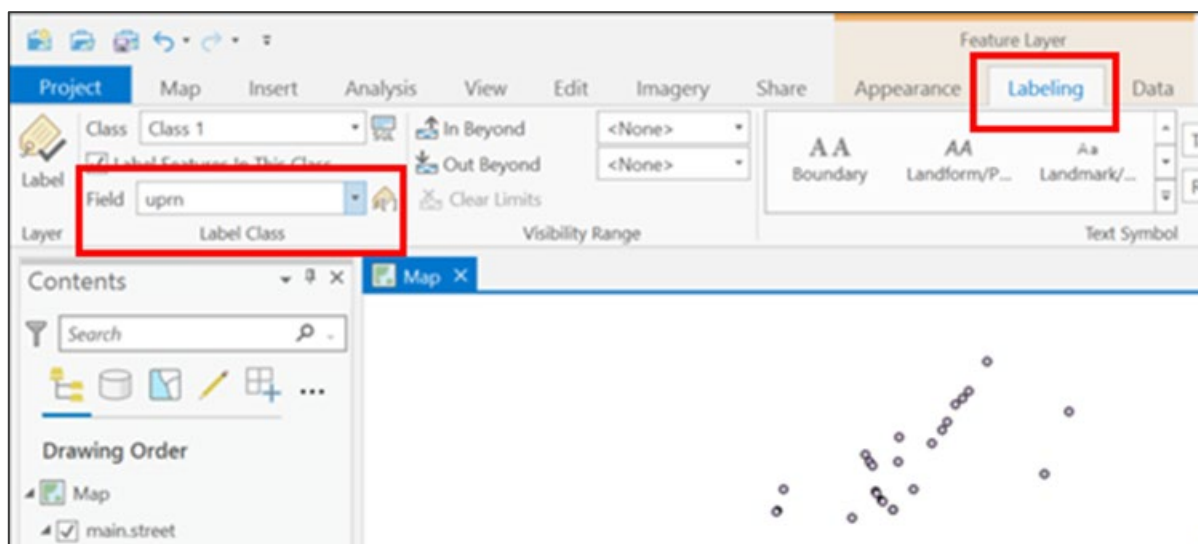
8. In the Add Relate dialog box, change the parameters as below:
- **Layer Name or Table View:** The table the relate is being added to (for example, *main.blpu*).
 - **Relate Table:** The table the relate is being taken from (for example, *main.application.xref*).
 - **Input Relate Field and Output Relate Field:** The field to relate the two tables on (for example, *uprn*).
 - **Relate Name:** Something simple that reflects the tables being related (for example, *blpu to application_xref*).



9. Once the parameters are set, click OK.
10. Repeat steps 7, 8 and 9 for all the relates listed in step 6. You may wish to confirm the relates have been successful by right-clicking on a layer and navigating to *Properties > Relates*.



11. Once the data has been loaded into ArcGIS, you may wish to display more relevant information in the *Explore* tool than the Esri-defined *Object ID*. To change this, use the following instructions:
 - a. Click on the layer that you wish to change the label field of.
 - b. Select the *Labeling* tab (under *Feature Layer*)
 - c. Change the *Field* to the desired field (for example, *UPRN*).

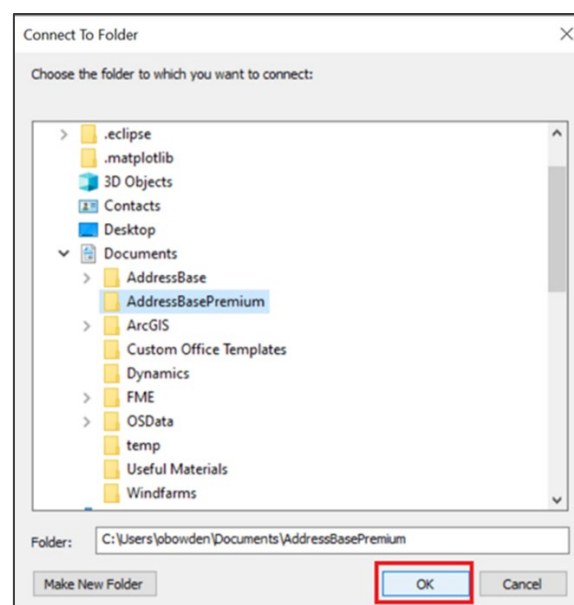
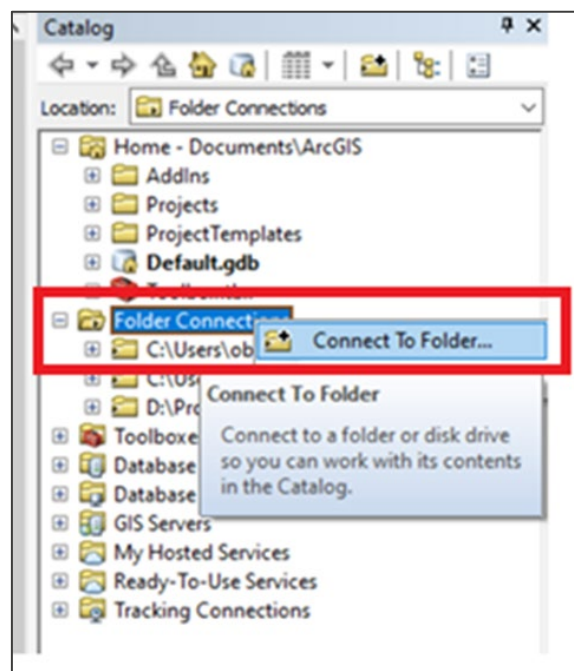


7.2 Accessing GeoPackage data via ArcGIS Desktop

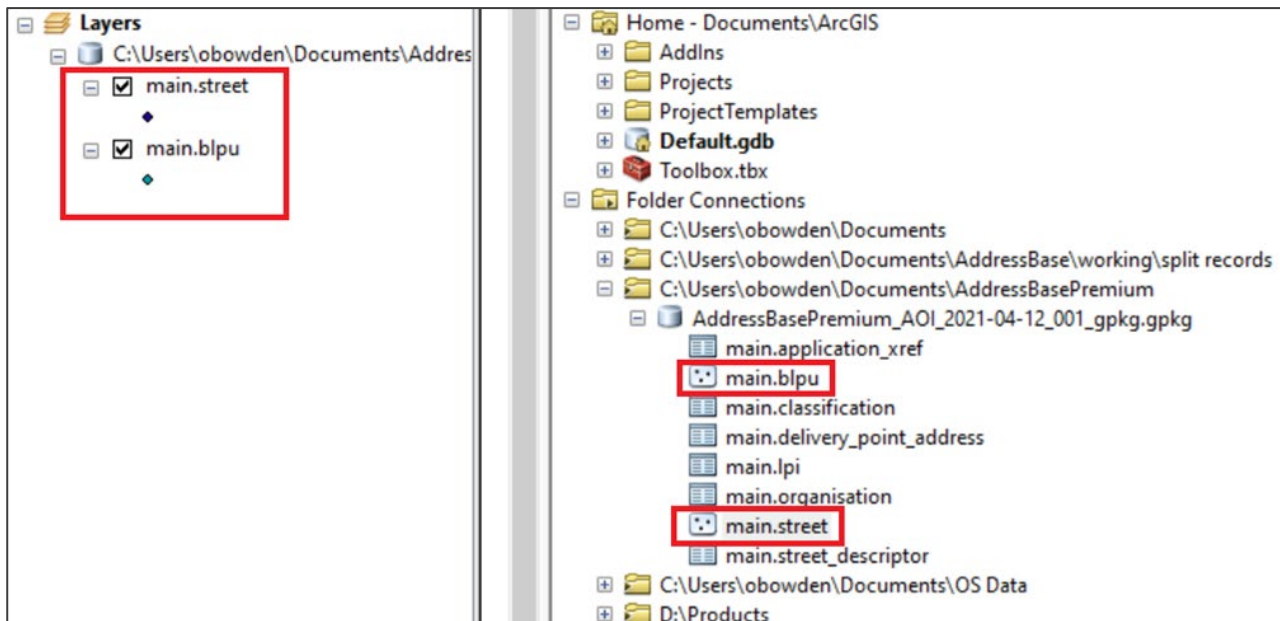
The following instructions are based on ArcGIS Desktop version 10.

Note: The file must have the file extension .gpkg (not case sensitive) to recognise it as an OGC GeoPackage. Make sure the AddressBase Premium GeoPackage is stored in a suitable location and the file is unzipped.

1. Start ArcCatalog as a separate program, or within ArcMap if you are using version 10.
2. To establish a connection to the folder where the GeoPackage file is stored, right-click on *Folder Connections > Connect To Folder*, then navigate to the folder containing the data. Select that folder and click OK.

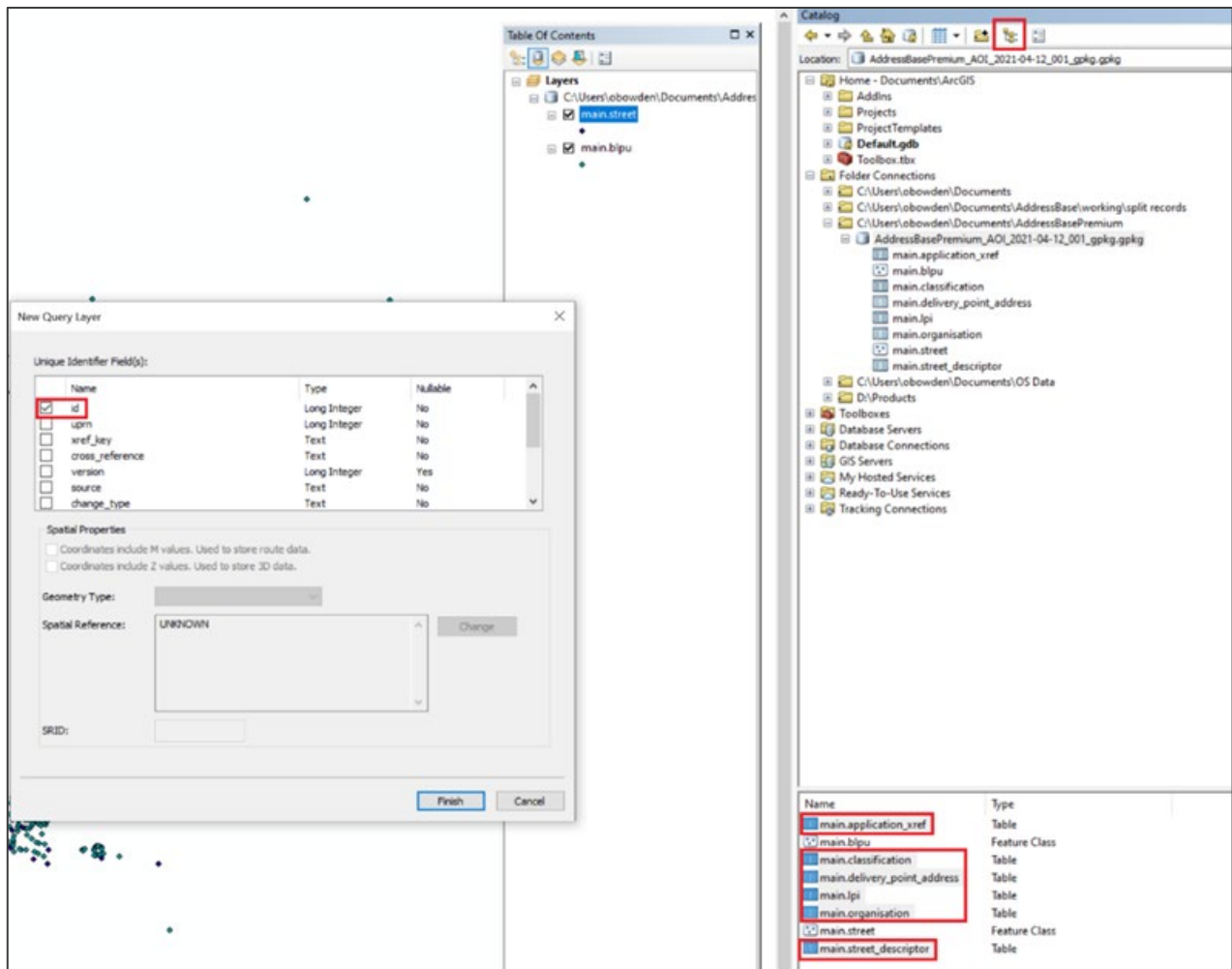


3. Navigate to the newly connected folder where the GeoPackage is stored. Open the GeoPackage to view its contents.
4. Drag the data with spatial elements (i.e. *main.street* and *main.blpu*, which are designated by the point symbol) into the Map View. From here, they can be visualised.



5. Add the standalone tables to the map. The tables have no spatial element so cannot be visualised, but they can be related to the data with a spatial element. Select the standalone tables and drag them into the Map View. When prompted, select *ID* as the *Unique Identifier Field(s)* before clicking *Finish*.

Note: The 'Toggle Contents Panel' can be used to open a window where you can select multiple tables at once.



6. You need to create the following joins / relates between the tables, as stated in the [AddressBase Premium Technical Specification](#):

BLPU (ID21_BLPV_Records):

- UPRN – Application Cross Reference (ID23_XREF_Records) UPRN
- UPRN – LPI (ID24_LPI_Records) UPRN
- UPRN – Delivery Point Address (ID28_DPA_Records) UPRN
- UPRN – Organisation (ID31_Org_Records) UPRN
- UPRN – Classification (ID32_Class_Records) UPRN

LPI (ID24_LPI_Records):

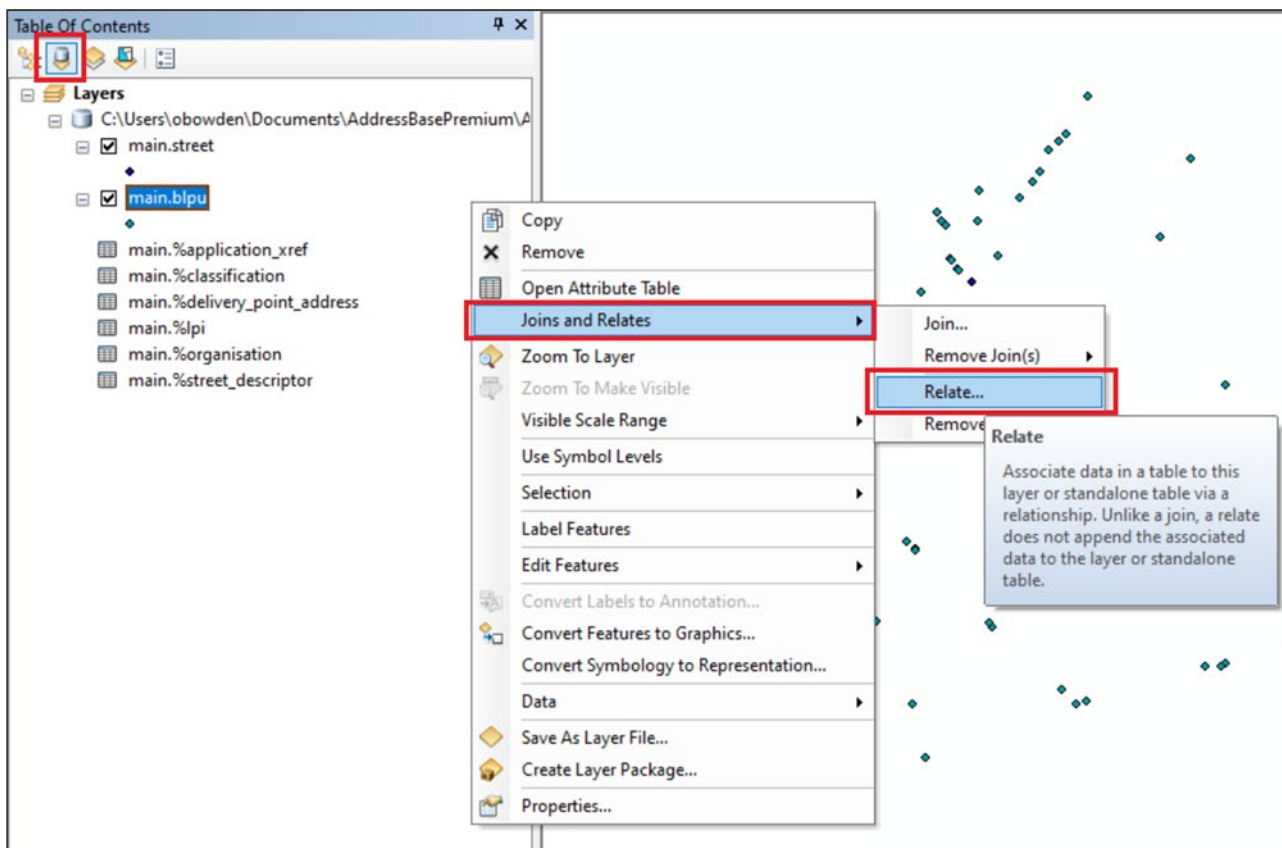
- USRN – Street (ID11_Street_Records) USRN

Street (ID11_Street_Records):

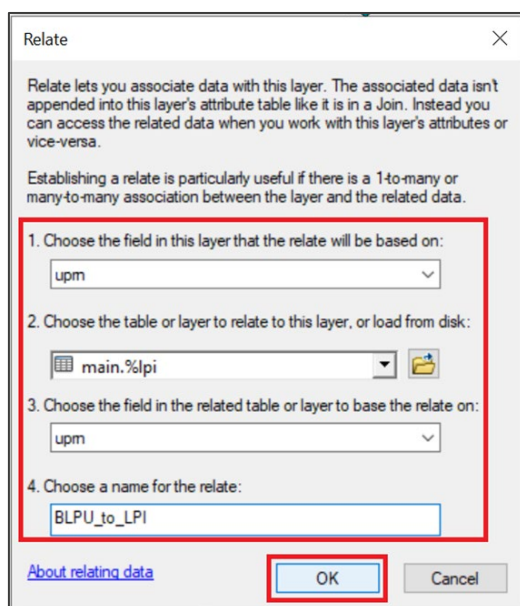
- USRN – Street Descriptor (ID15_StreetDesc_Records) USRN

7. To create the joins / relates between tables, select the *Source* button in the left-hand navigation window and right-click on the first table that you wish to relate to another. To create the relevant *relates*, use the following instructions:

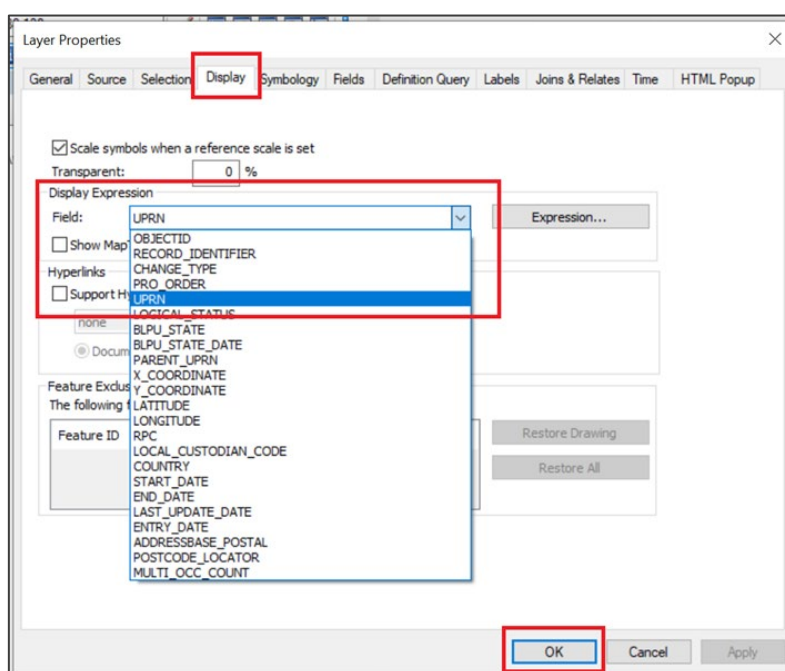
a. Click *Joins and Relates > Relate...*



- b. From the first dropdown menu, select the attribute from the first table that will be used to create the relate between the two tables. (Apply the relationships as listed in step 5.)
- c. From the second dropdown, select the table that is going to be related to. (Apply the relationships as listed in step 5.)
- d. From the third dropdown, select the attribute from the table that is being related to. (Apply the relationships as listed in step 5.)
- e. In the fourth box, input a relevant name for the relate (for example, *BLPU_to_LPI*).



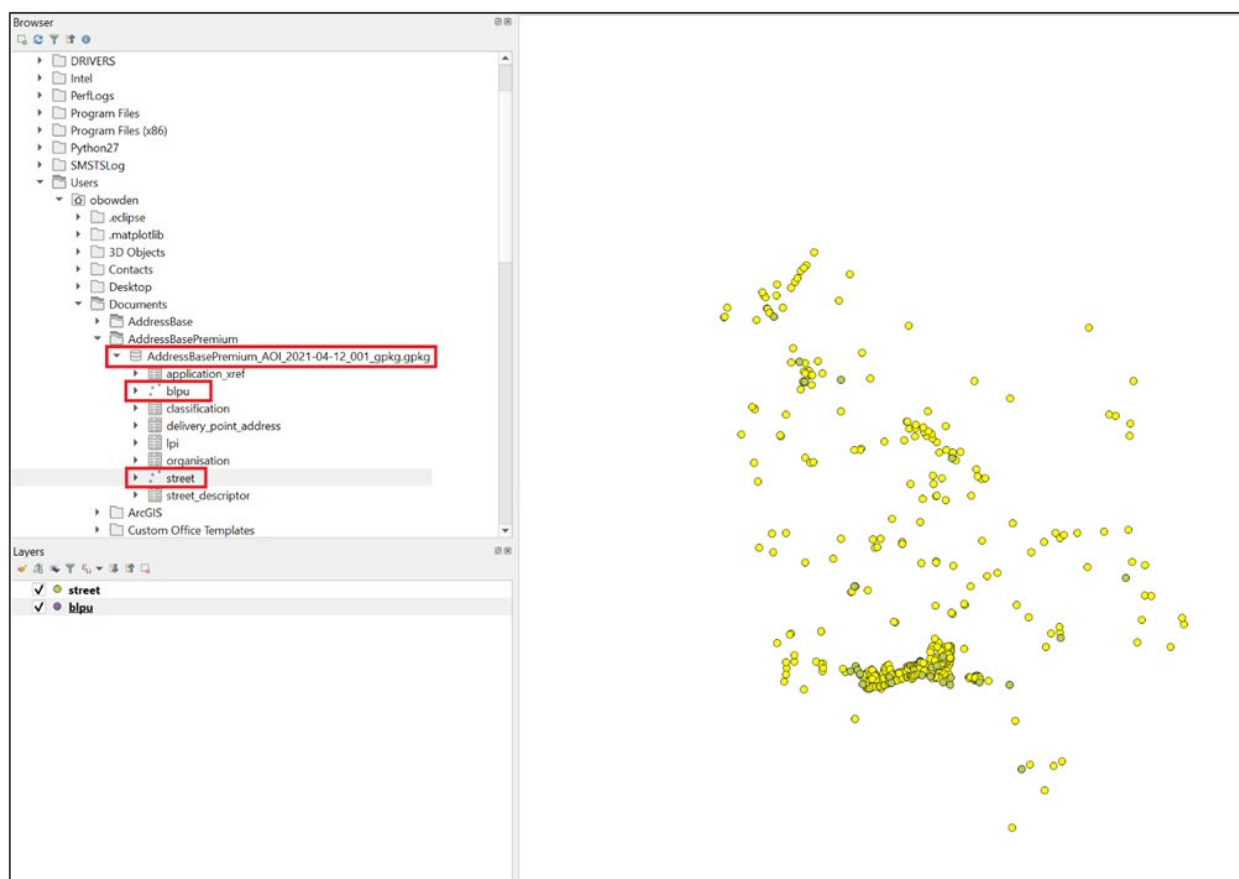
- f. Click OK.
 - g. Repeat this process for all of the *joins / relates* listed in step 5.
8. Once the data has been loaded into ArcMap, you may wish to display more relevant information in the *Info* tool than the Esri-defined *Object ID*. To change this, use the following instructions:
- a. Double-click on the spatial dataset that you wish to change the *Display Expression* of.
 - b. Select the *Display* tab.
 - c. Change the *Field* to the desired field (for example, *UPRN*).
 - d. Click OK.



7.3 Accessing GeoPackage data via QGIS

These instructions were created using QGIS version 3.14. Other versions of QGIS can be used, from version 2.10.1 onward.

1. Launch QGIS.
2. Navigate to the GeoPackage location in the QGIS Browser panel. If the GeoPackage is not listed when this container is expanded, then right-click on the GeoPackage location and select *New Connection*.
3. In the File Explorer window that launches, navigate to the location where you saved the AddressBase Premium GeoPackage and select it.
4. In the QGIS Browser panel, click on the drop-down arrow next to your AddressBase Premium GeoPackage to view the contents of the GeoPackage file. Drag the data with spatial elements (i.e. *main.street* and *main.blpu*, which are designated by the point symbol) into the map. From here, they can be visualised.



5. Next, add the standalone tables to the map. These tables have no spatial element so cannot be visualised, but they can be related to the data with a spatial element.

6. You need to create the following joins / relates between the tables, as stated in the [AddressBase Premium Technical Specification](#):

BLPU (ID21_BLPUs_Records):

- UPRN – Application Cross Reference (ID23_XREF_Records) UPRN
- UPRN – LPI (ID24_LPI_Records) UPRN
- UPRN – Delivery Point Address (ID28_DPA_Records) UPRN
- UPRN – Organisation (ID31_Org_Records) UPRN
- UPRN – Classification (ID32_Class_Records) UPRN

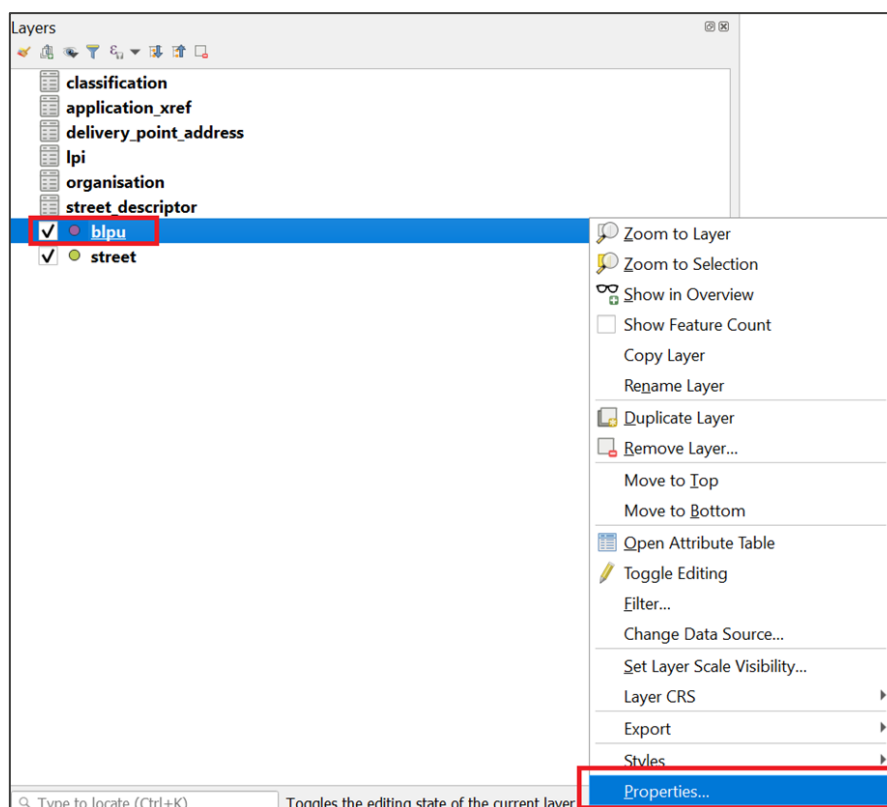
LPI (ID24_LPI_Records):

- USRN – Street (ID11_Street_Records) USRN

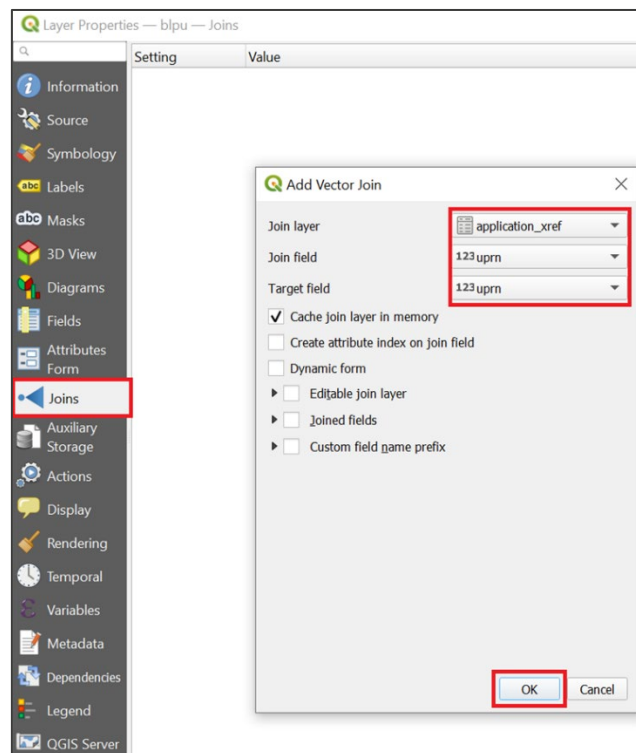
Street (ID11_Street_Records):

- USRN – Street Descriptor (ID15_StreetDesc_Records) USRN

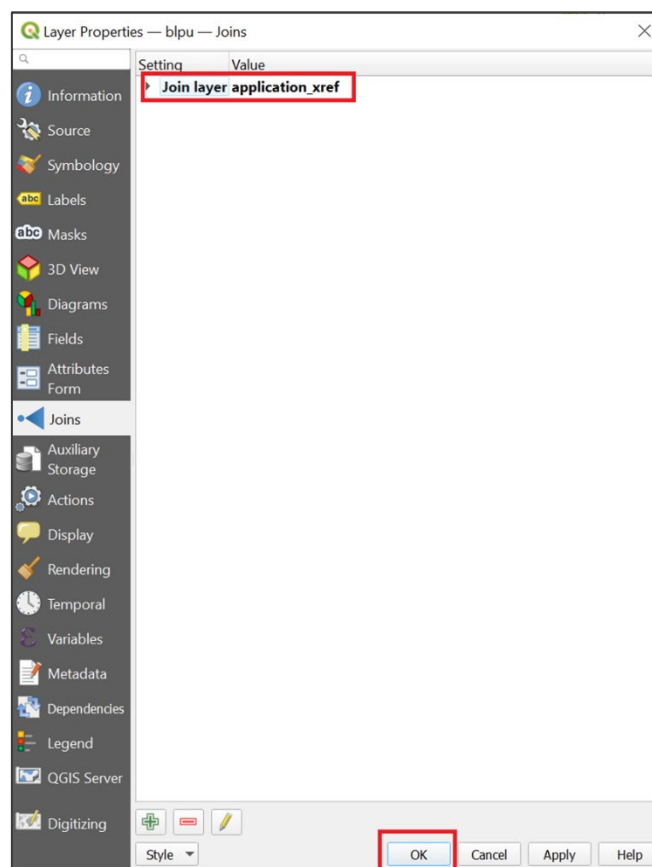
7. To create the joins / relates between the tables, right-click on your *BLPU* layer and select *Properties...*



8. Go to the *Joins* tab found on the left-hand side.
9. Click the *green plus button* in the bottom left-hand corner.
10. Select your Join layer. For this example, it is *Application Cross reference (application_xref)*.
11. Select the Join field. For all BLPU links, this will be the *UPRN* (see the joins listed in step 4 above).
12. The Target field will also be the *UPRN* for this example.



13. Click OK. You should now have a *join* listed in your *Joins* window.



14. Click OK to return to your main QGIS mapping screen.

15. If you now select one of your BLPU records in the main mapping window, you will see the BLPU attributes and the relevant Application Cross References for that record.

Identify Results	
Feature	Value
▼ blpu	
▼ fid	139
▶ (Derived)	
▶ (Actions)	
fid	139
uprn	10025441019
logical_status	1
blpu_state	NULL
blpu_state_date	NULL
parent_uprn	NULL
x_coordinate	349453.1
y_coordinate	1003870.8
latitude	58.919117
longitude	-2.8794747
rpc	1
local_custodian_code	7655
country	S
addressbase_postal	N
postcode_locator	KW17 2SA
multi_occ_count	0
change_type	I
start_date	2011-09-01
end_date	NULL
last_update_date	2018-12-04
entry_date	2011-09-01
application_xref_id	518
application_xref_xref_key	7655X900986852
application_xref_cross_reference	osgb5000000001273889
application_xref_version	12
application_xref_source	7666MA
application_xref_change_type	I
application_xref_start_date	2018-12-04
application_xref_end_date	NULL
application_xref_last_update_...	2018-12-04
application_xref_entry_date	2018-10-26

16. You can now repeat steps 7 to 14 for all the additional joins (listed in step 6) you wish to make.

8. Loading GeoPackage into a database

The following sub-section provides step-by-step instructions on how to load the AddressBase Premium GeoPackage into PostgreSQL using GDAL / Command Prompt.

8.1 Loading a GeoPackage into PostgreSQL using GDAL

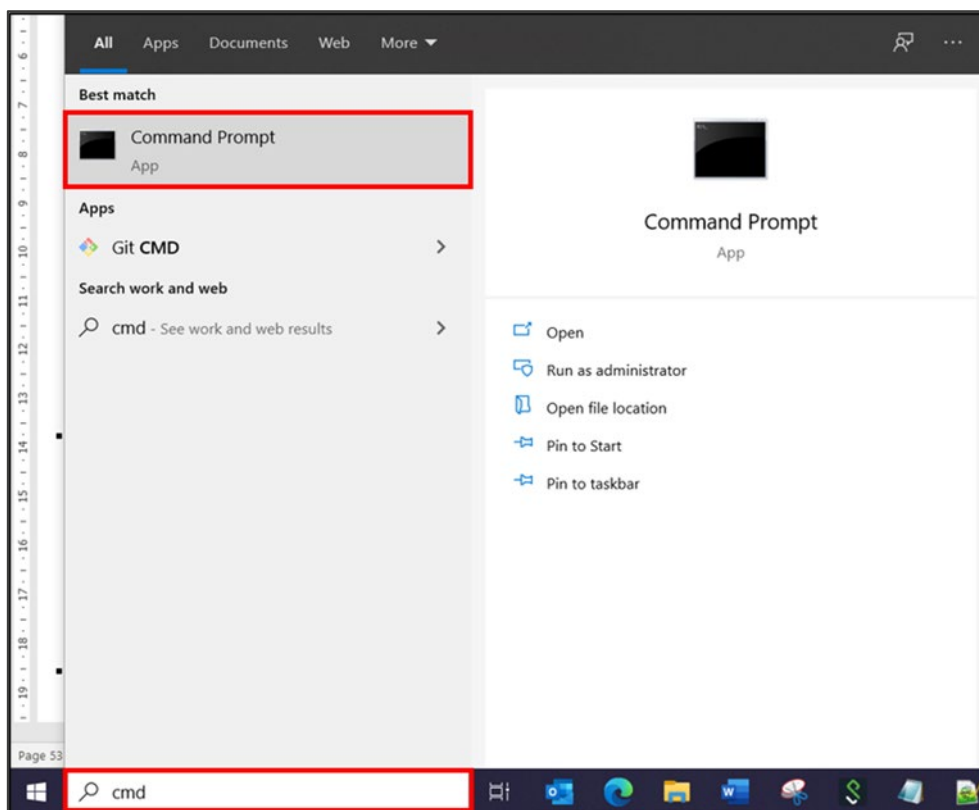
The following step-by-step instructions are based on PostgreSQL version 12.3 and assume that you have set-up your database with the PostGIS spatial extension. It is recommended that you have a basic understanding of database terminology before following these instructions.

Requirements:

- A development platform for PostgreSQL (for example, pgAdmin or dBeaver)
- A PostgreSQL database
- PostGIS extension
- GDAL extension
- Access to a Command Prompt or similar

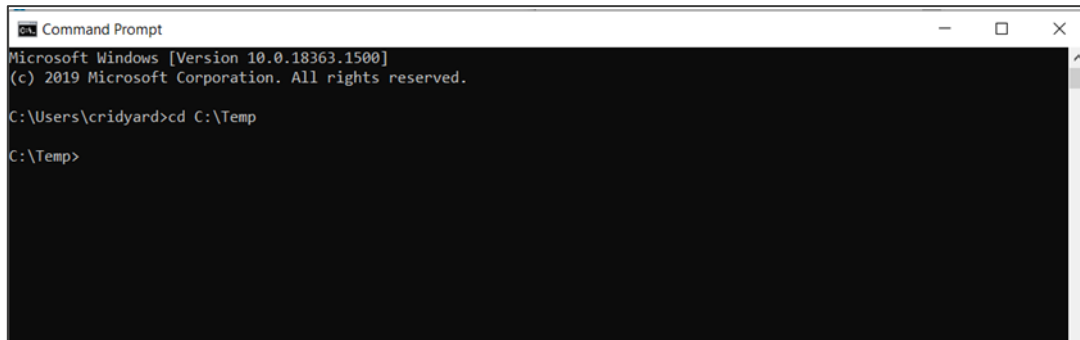
Note: Make sure the AddressBase Premium GeoPackage is stored in a suitable location and the file is unzipped.

1. Open the Command Prompt by clicking the Windows *Start* button in the bottom left-hand side of the screen and typing *cmd* into the search bar.



2. The Command Prompt will appear.

3. Change the directory of the Command Prompt App if necessary. The directory needs to point to the folder where the GeoPackage is stored. For this example, the GeoPackage to be loaded into PostgreSQL is stored in a C:\Temp folder.
4. In the Command Prompt, type `cd` followed by the directory of the location of the GeoPackage. Press *Enter* on the keyboard. This will change the directory.



5. Enter the following command:

```
ogr2ogr -progress -gt 65000 -f PostgreSQL "PG:user=<username> password=<password>  
dbname=<database> host=<host>" <data_name>.gpkg
```

Where:

username, **password**, **database** and **host** can be found within the subsequent database.

data_name is the AddressBase Premium GeoPackage file, for example, AddressBasePremium_AOI_2021-04-12_001_gpkg.gpkg.

Example:

```
ogr2ogr -progress -gt 65000 -f PostgreSQL "PG:user=postgres password=PG123 dbname=osdata  
host=localhost" AddressBasePremium_AOI_2021-04-12_001_gpkg.gpkg
```

6. Open your chosen development platform (for example, dBeaver) by going to the Windows *Start* button and selecting your chosen software.
7. In your Database Navigator, move to the database you loaded data into and expand the schema. Using the code above, the GeoPackage should be in your default schema (shown in bold).
8. The GeoPackage will now appear as new tables / a new table in the schema nominated as default.
9. Once loaded, you may want to add Primary and Foreign Keys to the data. 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. For the BLPU table, the UPRN provides a unique value.

9. Loading GML

This section provides step-by-step instructions on how to load the Geography Markup Language (GML) format of AddressBase Premium products into FME and a database.

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.

9.1 Loading GML via a translator

Several organisations provide a loader which will translate AddressBase Premium from GML and insert the data into a database or a GIS. Due to the relational nature of AddressBase Premium, GML will not load straight into most GIS, meaning an external translator would be helpful to most users. If you would like to load AddressBase Premium in GML format into a GIS, please contact your vendor for more details.

For more information on Ordnance Survey Partners who provide GML loaders, please visit the [Ordnance Survey Partner website](https://www.ordnancesurvey.co.uk/business-government/partner-member) (<https://www.ordnancesurvey.co.uk/business-government/partner-member>).

9.2 Loading GML into FME

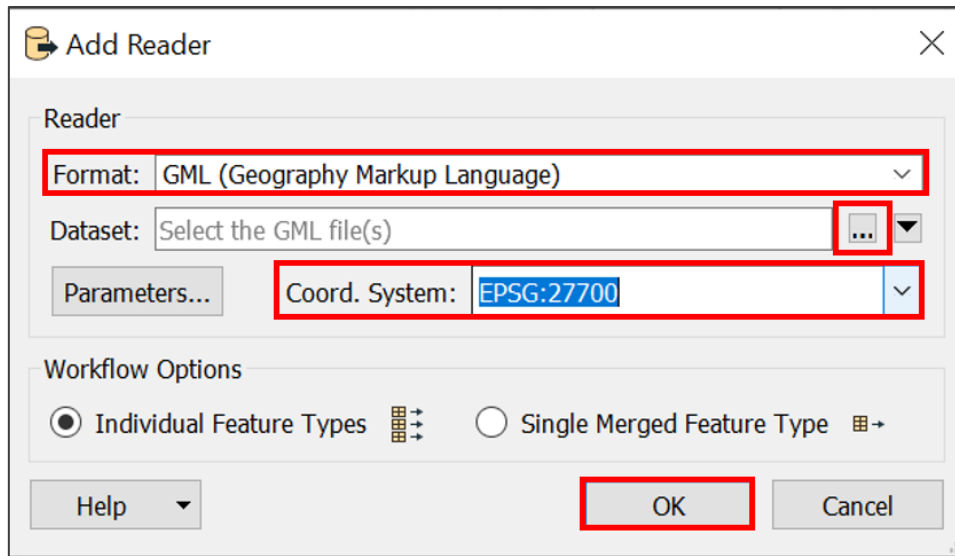
GML data can be viewed and loaded into Safe FME. When AddressBase Premium is read into FME, the non-spatial tables within the product are automatically related to the spatial tables.

Note: The following instructions are written using FME version 2020.0, but other versions are available.

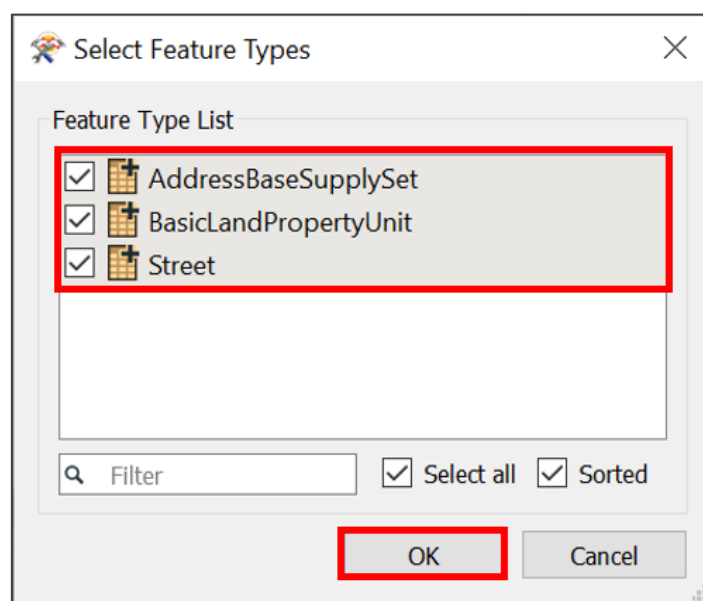
1. Open FME Workbench.
2. Click the *Reader* button in the ribbon.



3. An Add Reader dialog box will appear.



4. Select *GML (Geography Markup Language)* as the Format.
5. Select a coordinate reference system from the Coord. System dropdown list. It's recommended that *British National Grid (EPSG:27700)* is selected as the coordinate reference system; however, you can select a different coordinate reference system if required.
6. Click the *three dots* button next to the Dataset box. This will bring up another dialog box where you can select the GML dataset to read into FME.
7. Click OK and OK again.
8. A Select Feature Types dialog box will appear in which you can select what feature types should be read into FME. The BasicLandPropertyUnit and Street feature types also include the non-spatial table information, such as LPI and organisation.



9. Click OK.

10. The three feature types should appear as readers.

[More information on GML readers in FME](https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_ReadersWriters/gml/gml.htm) is available on the Safe Software website (https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_ReadersWriters/gml/gml.htm).

9.3 Using GDAL to load GML into a database

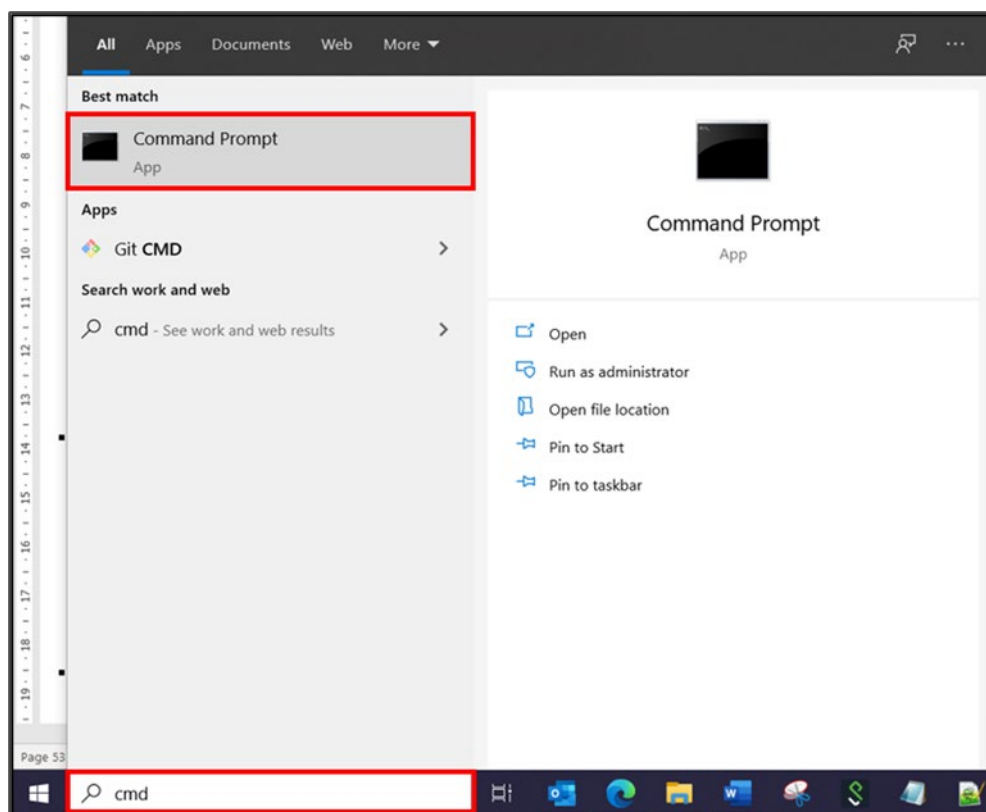
The following step-by-step instructions are based on PostgreSQL version 12.3 and assume that you have set-up your database with the PostGIS spatial extension in a Windows environment. It is recommended that you have a basic understanding of database terminology before following these instructions.

Requirements:

- A development platform for PostgreSQL (for example, pgAdmin or dBeaver)
- A PostgreSQL database
- PostGIS extension
- GDAL extension
- Access to a Command Prompt or similar

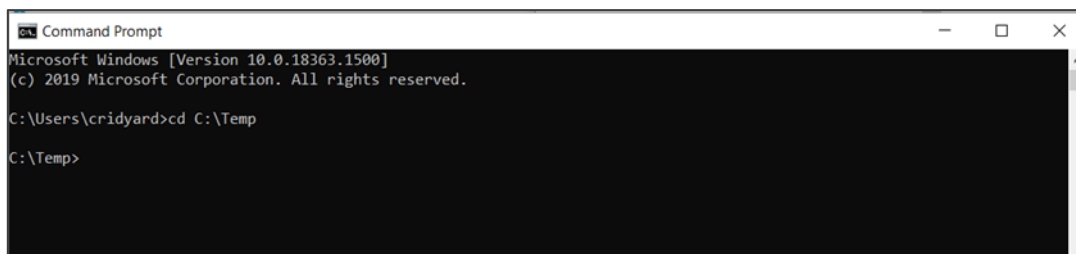
Note: Before you start, make sure the AddressBase Premium GML is stored in a suitable location and the file is unzipped.

1. Open the Command Prompt by clicking the Windows *Start* button in the bottom left-hand side of the screen and typing *cmd* into the search bar.



2. The Command Prompt will appear.

3. Change the directory of the Command Prompt App if necessary. The directory needs to point to the folder where the GML is stored. For this example, the GML to be loaded into PostgreSQL is stored in a C:\Temp folder.
4. In the Command Prompt, type `cd` followed by the directory of the location of the GML. Press *Enter* on the keyboard. This will change the directory.



5. Enter the following command:

```
ogr2ogr -f "PostgreSQL" PG:"host=<host> port=<port> dbname=<database> user=<username>  
password=<password>" <data_name.gml> -overwrite -progress -t_srs EPSG:<CRS>
```

Where:

host, **port**, **database**, **username** and **password** can be found within the subsequent database.

data_name is the AddressBase Premium GML file, for example, AddressBasePremium_AOI_2021-04-12_001_gml.gml.

CRS is the coordinate reference system. British National Grid (27700) is recommended but you can use alternative coordinate reference systems if required.

Example:

```
ogr2ogr -f "PostgreSQL" PG:"host=localhost port=5432 dbname=osdata user=postgres  
password=pgdata1234" AddressBasePremium_AOI_2021-04-12_001_gml.gml -overwrite -  
progress -t_srs EPSG:27700
```

6. Open your chosen development platform by going to the Windows *Start* button and selecting your chosen software.
7. In your Database Navigator, move to the database you loaded data into and expand the schema.
8. From using the code above, the GML files should be in your default schema (shown in bold). The data should load into two tables: basiclandpropertyunit and street. Data in non-spatial tables will be automatically matched to the spatial tables based on a unique identifier.
9. Once loaded, you may want to add Primary and Foreign Keys to the data. 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. For the BLPU table, the UPRN provides a unique value.

Note: You can also manage your database using the PostGRES SQL Command Prompt.

10. COU implementation guidance

10.1 Introduction to COU

AddressBase Premium and AddressBase Premium Islands are complex relational datasets that are used by a variety of customers who use a variety of methods and software to manage the data. Some of the software solutions take a considerable length of time to load and manage the data. A change-only update (COU) is a simple and effective way to keep data holdings up to date without spending considerable time loading and managing a full supply every time the data is refreshed.

A COU means you will only be supplied with the features which have changed since your last supply. The following sub-sections provide guidance on how to manage a COU supply of AddressBase Premium data.

Note: If you receive a tile supply, you will receive Change Chunks. This means if a record within your tile has changed, then all of the records in that tile will be provided to you as inserts, and no updates or deletes will be issued. This is not applicable for AddressBase Premium Islands as a tile supply is not available for that product.

10.2 Types of change within a COU

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

1. Deletes (CHANGE_TYPE 'D') are objects that have ceased to exist in your area of interest (AOI) since the last product refresh.
2. Inserts (CHANGE_TYPE 'I') are objects that have been newly inserted into your AOI since the last product refresh.
3. Updates (CHANGE_TYPE 'U') are objects that have been updated in your AOI since the last product refresh.

10.3 High-level COU implementation model

Figure 2 outlines how to implement the AddressBase Premium COU within a database. It also shows the necessary primary keys needed to implement the COU for each relational table.

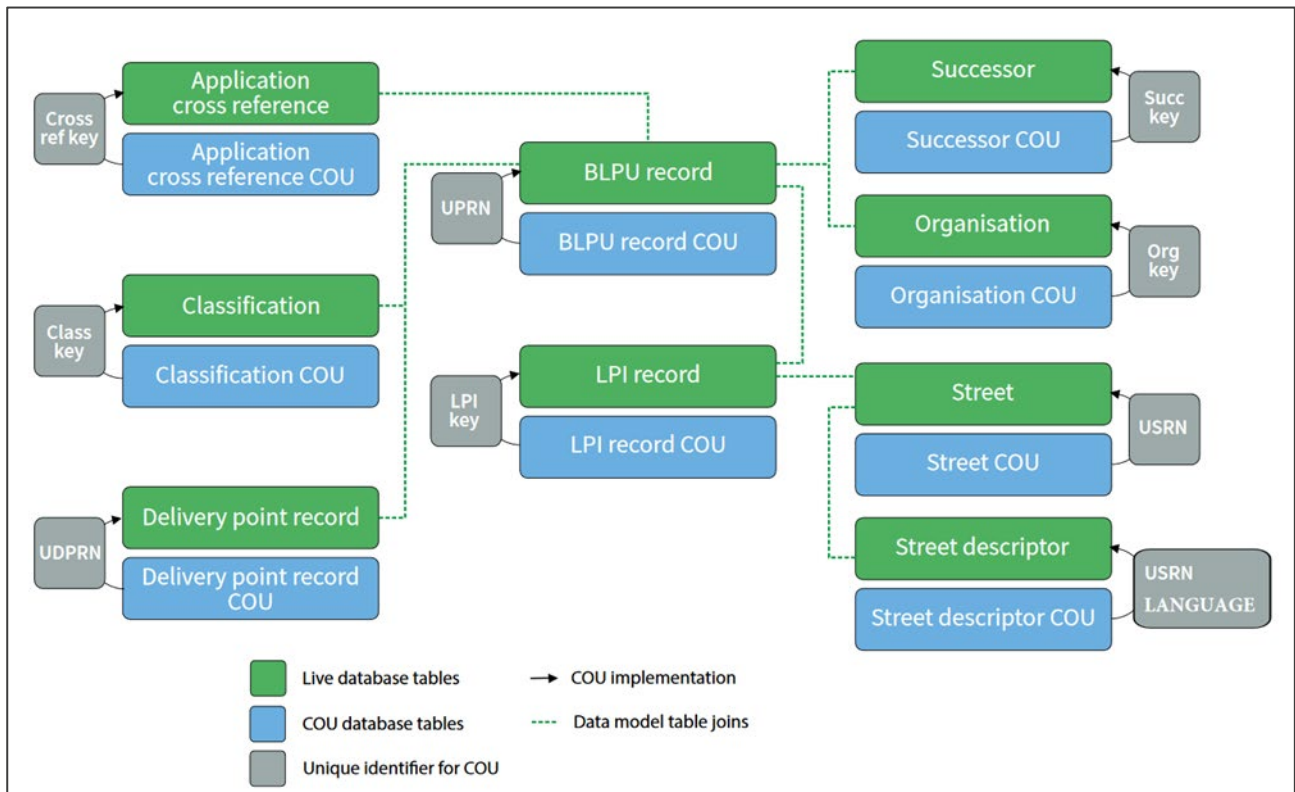


Figure 2: High-level COU implementation model.

10.4 High-level COU implementation model – with archiving

Before a COU is applied, there may be a business requirement to archive existing address records. Figure 3 outlines how to implement the AddressBase Premium COU within a database, shows the necessary primary keys needed to implement the COU for each relational table, and how to archive existing records.

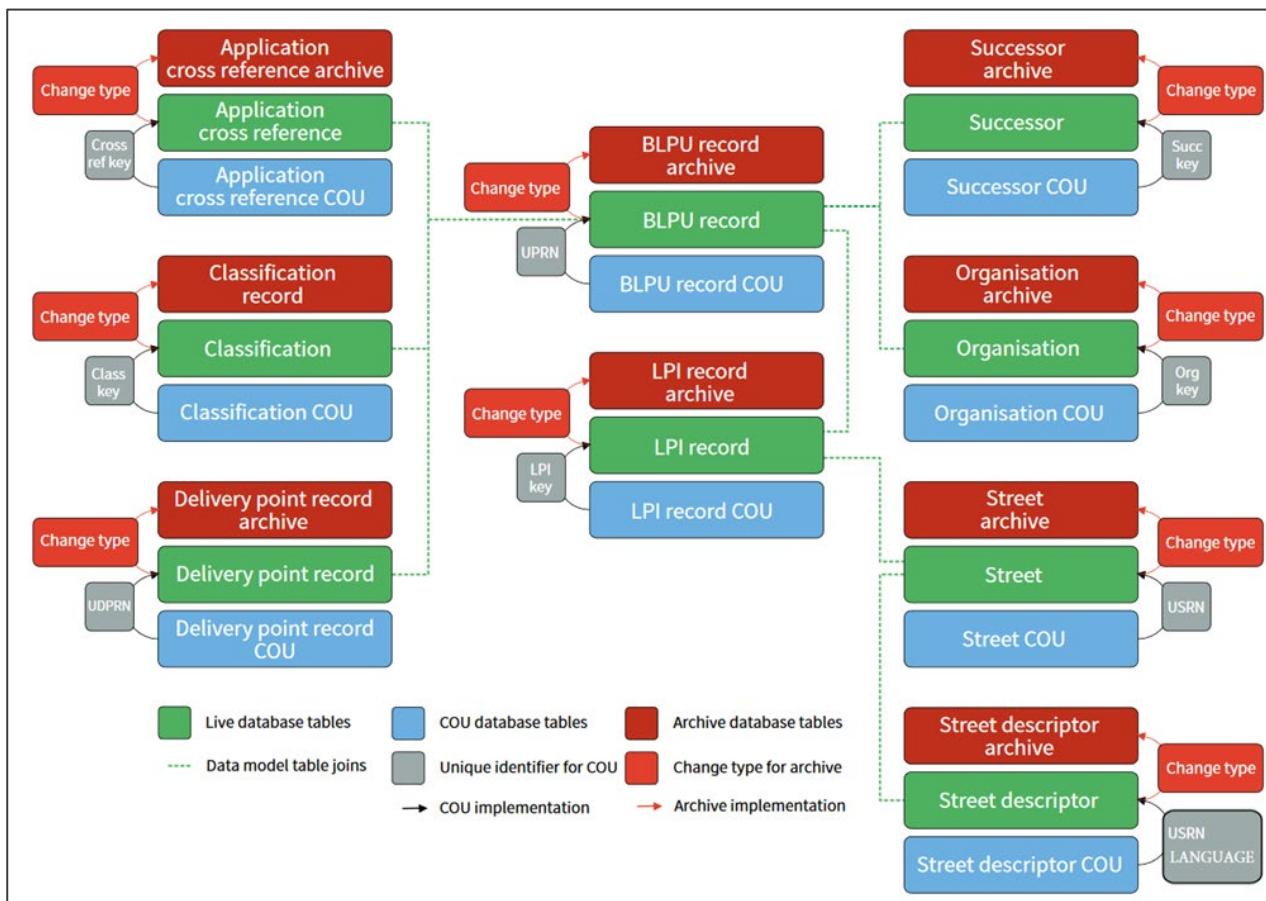


Figure 3: High-level COU implementation model on how to create archive tables and apply a COU.

10.5 Applying COU to tables

10.5.1 Changes to the BLPU table

Within the Basic Land and Property Unit (BLPU) table, there will not be any 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 abp_blpu
GROUP BY uprn
HAVING ( COUNT(uprn) > 1 );
```

This query should return 0 rows, and this confirms there are no duplicates. As there are no duplicate records, the UPRN can be used 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 abp_blpu WHERE uprn IN (SELECT distinct uprn FROM abp_blpu_cou WHERE change_type!= 'I');
```

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

```
INSERT INTO abp_blpu SELECT * FROM abp_blpu_cou WHERE change_type != 'D';
```

Some of the COU records that are change type 'U' (updates) may change the Logical Status Code from 'I' to '8', meaning that this address has become 'Historical'. This means that the BLPU table intrinsically archives historical records.

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 and populate it with records that are being Updated and Deleted from the live BLPU table.

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

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

```
CREATE TABLE abp_blpu_archive AS SELECT * FROM abp_blpu WHERE uprn IN (SELECT distinct uprn FROM abp_blpu_cou WHERE change_type != 'I');
```

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

```
DELETE FROM abp_blpu WHERE uprn IN (SELECT distinct uprn FROM abp_blpu_cou WHERE change_type!= 'I');
```

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

```
INSERT INTO abp_blpu SELECT * FROM abp_blpu_cou WHERE change_type != 'D';
```

10.5.2 Changes to the Classification table

Because there is a one-to-many relationship between the BLPU table and the Classification table, there can be records in the Classification table that share a UPRN. To apply COU to the correct record, users should use the Class_Key to ensure that the correct classification record is updated.

Table 3: Example of using the Class_Key to apply a COU to one of two classification records that share a UPRN in a Classification table.

Classification	Record 1	Record 2	Classification	Updated record	Record 2	Classification	COU record
Record identifier	32	32	Record identifier	32	32	Record identifier	32
Change type	I	I	Change type	U	I	Change type	U
Pro order	922371	922372	Pro order	922500	922372	Pro order	922500
UPRN	100062645004	100062645004	UPRN	100062645004	100062645004	UPRN	100062645004
Class key	1715C000002050	1715C802457028	Class key	1715C000002881	1715C802457028	Class key	1715C000002881
Classification code	U	CS	Classification code	CR08	CS	Classification code	CR08
Class scheme	AddressBase Premium classification scheme	VOA Primary Description	Class scheme	AddressBase Premium classification scheme	VOA Primary Description	Class scheme	AddressBase Premium classification scheme
Scheme version	1.0.0	1.0.0	Scheme version	1.0.0	1.0.0	Scheme version	1.0.0
Start date	2011-12-01	2010-03-16	Start date	2011-12-01	2010-01-16	Start date	2011-12-01
End date	N/A	N/A	End date	N/A	N/A	End date	N/A
Last update	2011-12-01	2010-08-12	Last update date	2013-05-04	2010-08-12	Last update	2013-05-04
Entry date	2011-12-01	2010-03-16	Entry date	2011-12-01	2010-03-16	Entry date	2011-12-01

The example in Table 3 illustrates a scenario when a user would need to choose between two classification records that have the same UPRN. In this case, the Class_Key has been used to apply the COU to Record 2.

To achieve this outcome (without archiving the 'old' Record 2), we can use the following SQL commands to apply the COU:

#Initially delete the existing records that are being updated and deleted:

```
DELETE FROM abp_classification WHERE class_key IN (SELECT distinct class_key FROM abp_classification_cou WHERE change_type != 'I');
```

#Insert the new update records and the new insert records:

```
INSERT INTO abp_classification SELECT * FROM abp_classification_cou WHERE change_type != 'D';
```

One thing you may want to consider is keeping an archive of the updated and deleted classification records. For example, this might be useful to understand when an address has changed use from residential to commercial.

To achieve this outcome for change types 'U' (updates) or 'D' (deletes) (with archiving), we can use the following SQL commands to apply the COU:

#The following command creates an archive table of the records that are being updated and deleted from the existing Classification table. If this table already exists, you can simply use INSERT INTO rather than CREATE TABLE:

```
CREATE TABLE abp_classification_archive AS SELECT * FROM abp_classification WHERE class_key IN (SELECT distinct class_key FROM abp_classification_cou WHERE change_type != 'I');
```

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

```
DELETE FROM abp_classification WHERE class_key IN (SELECT distinct class_key FROM aabp_classification_cou WHERE change_type != 'I');
```

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

```
INSERT INTO abp_classification SELECT * FROM abp_classification_cou WHERE change_type != 'D';
```

Table 4: Classification and archive record.

Classification	Archive record
Record identifier	32
Change type	I
Pro order	706838
UPRN	116000665
Class key	9055C000081107
Classification code	CL10RE
Class scheme	AddressBase Premium classification scheme
Scheme version	1.0.0
Start date	2011-12-01
End date	N/A
Last update date	2011-12-01
Entry date	2011-12-01

Note: When the updated or deleted records are moved into an archive table, the end date may not always be populated, as seen in Table 4. If this is the case, users may wish to consider adding an end_date (which could be based on the epoch date that it was archived) as shown in Table 5. Adding an end date to an updated or deleted record will enable querying for a particular timeframe.

Table 5: Classification and archive record with an end date.

Classification	Archive record
Record identifier	32
Change type	I
Pro order	706838
UPRN	116000665
Class key	9055C000081107
Classification code	CL10RE
Class scheme	AddressBase Premium classification scheme
Scheme version	1.0.0
Start date	2011-12-01
End date	2013-05-04
Last update date	2011-12-01
Entry date	2011-12-01

10.5.3 Changes to the Organisation table

The numerous one-to-many relationships between the BLPU table and the Organisation table mean there can be records in the Organisation table that share a UPRN. To apply COU to the correct record, we should use the Org_Key to ensure that the correct classification record is updated.

To apply the COU to the Organisation table (without archiving), the following code can be used:

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

```
DELETE FROM abp_organisation WHERE org_key IN (SELECT distinct org_key FROM  
abp_organisation_cou WHERE change_type != 'I');
```

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

```
INSERT INTO abp_organisation SELECT * FROM abp_organisation_cou WHERE change_type !=  
'D';
```

As with the Classification table, the changes in Organisation name may be useful to keep as archives as doing so will allow a business to find previous organisations and understand when name changes were made.

To apply the COU to the Organisation table (with archiving), the following code can be used:

#The following command creates an archive table of the records that are being updated and deleted from the existing Organisation table. If this table already exists, you can simply use INSERT INTO rather than CREATE TABLE:

```
CREATE TABLE abp_organisation_archive AS SELECT * FROM abp_organisation WHERE org_key  
IN (SELECT distinct org_key FROM abp_organisation_cou WHERE change_type != 'I');
```

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

```
DELETE FROM abp_organisation WHERE org_key IN (SELECT distinct org_key FROM  
aabp_organisation_cou WHERE change_type != 'I');
```

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

```
INSERT INTO abp_organisation SELECT * FROM abp_organisation_cou WHERE change_type !=  
'D';
```

10.5.4 Changes to the Delivery Point Address table

Within the Delivery Point Address table, there will not be any records with the same Unique Delivery Point Reference Number (UDPRN). This can be tested by checking the number of records that have the same UDPRN. The following SQL code would notify you of any duplicates:

```
SELECT udprn, COUNT(udprn) AS NumOccurrences FROM abp_delivery_point  
GROUP BY udprn  
HAVING (COUNT(udprn) > 1 );
```

This query should return 0 rows, and this confirms that there are no duplicates. As there are no duplicate records, we can therefore use the UDPRN to apply the COU.

To apply the COU to the Delivery Point Address table (without archiving), the following code can be used:

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

```
DELETE FROM abp_delivery_point WHERE udprn IN (SELECT distinct udprn FROM  
abp_delivery_point_cou WHERE change_type != 'I');
```

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

```
INSERT INTO abp_delivery_point SELECT * FROM abp_delivery_point_cou  
WHERE change_type != 'D';
```

The Delivery Point Address table does not have the ability to hold historical records as it is the current view of the Royal Mail Delivery Point Address File (PAF). Therefore, in order to capture the historical records, you will need to create an archive table that is populated when records are either deleted or updated. The following code will create the archive records:

To apply the COU to the Delivery Point Address table (with archiving), the following code can be used:

#The following command creates an archive table of the records that are being updated and deleted from the existing Delivery Point Address table. If this table already exists, you can simply use INSERT INTO rather than CREATE TABLE:

```
CREATE TABLE abp_delivery_point_archive AS SELECT * FROM abp_delivery_point  
WHERE udprn IN (SELECT distinct udprn FROM abp_delivery_point_cou  
WHERE change_type != 'I');
```

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

```
DELETE FROM abp_delivery_point WHERE udprn IN (SELECT distinct udprn  
FROM abp_delivery_point_cou WHERE change_type != 'I');
```

#The following command then inserts the new insert records and the new updated records into the Delivery Point Address table:

```
INSERT INTO abp_delivery_point SELECT * FROM abp_delivery_point_cou  
WHERE change_type != 'D';
```

10.5.5 Changes to the Land and Property Identifier table

The numerous one-to-many relationships between the BLPU table and the Land and Property Identifier (LPI) table mean there can be records in the LPI table that share a UPRN. To apply the COU to the correct record, we should use the LPI_Key to ensure that the correct classification record is updated.

To apply the COU to the LPI table (without archiving), the following code can be used:

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

```
DELETE FROM abp_lpi WHERE lpi_key IN (SELECT distinct lpi_key FROM abp_lpi_cou  
WHERE change_type != 'I');
```

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

```
INSERT INTO abp_lpi SELECT * FROM abp_lpi_cou WHERE change_type != 'D';
```

As with the BLPU table, some of the COU records that are change type 'U' (updates) may change the Logical Status Code from 'I' to '8', meaning that this address has become 'historical'. This means that the LPI table intrinsically archives the historical record.

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 and populate it with records that are being updated and deleted from the live LPI table.

To apply the COU to the LPI table (with archiving), the following code can be used:

#The following command creates an archive table of the records that are being updated and deleted from the existing LPI table. If this table already exists, you can simply use INSERT INTO rather than CREATE TABLE:

```
CREATE TABLE abp_lpi_archive AS SELECT * FROM abp_lpi WHERE lpi_key IN (SELECT distinct  
lpi_key FROM abp_lpi_cou WHERE change_type != 'I');
```

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

```
DELETE FROM abp_lpi WHERE lpi_key IN (SELECT distinct lpi_key FROM abp_lpi_cou  
WHERE change_type != 'I');
```

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

```
INSERT INTO abp_lpi SELECT * FROM abp_lpi_cou WHERE change_type != 'D';
```


Table 6: An original LPI record next to a COU record. In this example, the record is being made historical (logical status code: 8) and therefore has a populated end date attribute.

LPI	Record	COU Record
Record identifier	24	24
Change type	I	U
Pro order	478857	478857
UPRN	100000527208	100000527208
LPI key	4520L000005174	4520L000005174
Language	ENG	ENG
Logical status	I	8
Start date	'2001-03-23'	'2001-03-23'
End date		'2013-04-24'
Last update date	'2010-05-21'	'2013-04-24'
Entry date	'2001-03-23'	'2001-03-23'
SAO start number		
SAO start suffix		
SAO end number		
SAO end suffix		
SAO text		
PAO start number		
PAO start suffix		
PAO end number		
PAO end suffix		
PAO text	'SITE OF FORMER MISER NETHAULERS'	'FORMER SITE OF MISER NETHAULERS'
USRN	36815950	36815950
USRN match indicator	I	I
Area name		
Level		
Official flag		

10.5.6 Changes to the Street table

Within the Street table, there will not be any records with the same Unique Street Reference Number (USRN). This can be tested by checking the number of records that have the same USRN. The following SQL code would notify you of any duplicates:

```
SELECT usrn, COUNT(usrn) AS NumOccurrences
FROM abp_street
GROUP BY usrn
HAVING ( COUNT(usrn) > 1 );
```

This query should return 0 rows, and this confirms there are no duplicates. As there are no duplicate records, we can use the USRN to apply the COU.

To apply the COU to the Street table (without archiving), the following code can be used:

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

```
DELETE FROM abp_street WHERE usrn IN (SELECT distinct usrn FROM abp_street_cou WHERE
change_type != 'I');
```

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

```
INSERT INTO abp_street SELECT * FROM abp_street_cou WHERE change_type != 'D';
```

The Street table does not have the ability to hold historical records as it does not have a Logical Status Code attribute. Therefore, to capture the historical records, you will need to create an archive table that is populated when records are either deleted or updated.

To apply the COU to the Street table (with archiving), the following code can be used:

#The following command creates an archive table of the records that are being updated and deleted from the existing Street table. If this table already exists, you can simply use INSERT INTO rather than CREATE TABLE:

```
CREATE TABLE abp_street_archive AS SELECT * FROM abp_street WHERE usrn IN (SELECT
distinct usrn FROM abp_street_cou WHERE change_type != 'I');
```

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

```
DELETE FROM abp_street WHERE usrn IN (SELECT distinct usrn FROM abp_street_cou WHERE
change_type != 'I');
```

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

```
INSERT INTO abp_street SELECT * FROM abp_street_cou WHERE change_type != 'D';
```

10.5.7 Changes to the Street Descriptor table

Within the Street Descriptor table, there will not be any records with the same USRN and the same language. This is called a compound key, rather than having a single column as a Primary Key. This can be tested by checking the number of records that have the same USRN. The following SQL code would notify you of any duplicates:

```
SELECT usrn, language, COUNT(usrn) AS NumOccurrences
FROM abp_street_descriptor
GROUP BY usrn, language
HAVING ( COUNT(usrn) > 1 );
```

This query should return 0 rows, and this confirms there are no duplicates using the compound key. As there are no duplicate records, we can therefore use the USRN and LANGUAGE to apply the COU.

To apply the COU to the LPI table (without archiving), the following code can be used:

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

```
DELETE FROM abp_street_descriptor WHERE EXISTS (SELECT 1 FROM
abp_street_descriptor_cou WHERE abp_street_descriptor_cou.usrn = abp_street_descriptor.usrn
AND abp_street_descriptor_cou.language = abp_street_descriptor.language
AND abp_street_descriptor_cou.change_type != 'I' )
```

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

```
INSERT INTO abp_street_descriptor SELECT * FROM abp_street_descriptor_cou
WHERE change_type != 'D';
```

The Street Descriptor table does not have the ability to hold historical records as it does not have a Logical Status Code attribute. Therefore, to capture the historical records, you will need to create an archive table that is populated when records are either deleted or updated.

To apply the COU to the Street Descriptor table (with archiving), the following code can be used:

#The following command creates an archive table of the records that are being updated and deleted from the existing Street table. If this table already exists, you can simply use INSERT INTO rather than CREATE TABLE:

```
CREATE TABLE abp_street_descriptor_archive AS SELECT * FROM abp_street_descriptor
WHERE EXISTS (SELECT 1 FROM abp_street_descriptor_cou
WHERE abp_street_descriptor_cou.usrn = abp_street_descriptor.usrn
AND abp_street_descriptor_cou.language = abp_street_descriptor.language
AND abp_street_descriptor_cou.change_type != 'I' )
```

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

```
DELETE FROM abp_street_descriptor WHERE usrn IN (SELECT distinct usrn
FROM abp_street_descriptor_cou WHERE change_type != 'I');
```

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

```
INSERT INTO abp_street_descriptor SELECT * FROM abp_street_descriptor_cou  
WHERE change_type != 'D';
```

10.5.8 Changes to the Cross Reference table

Within the Cross Reference table, there will not be any records with the same XREF_KEY. This can be tested by checking the number of records that have the same XREF_KEY. The following SQL code would notify you of any duplicates:

```
SELECT XREF_KEY, COUNT(XREF_KEY) AS NumOccurrences  
FROM ABP_XREF  
GROUP BY XREF_KEY  
HAVING (COUNT(XREF_KEY) > 1);
```

The query above should return 0 rows and therefore confirms that there are no duplicates. As there are no duplicates, we can therefore use the XREF_KEY to apply the COU.

To apply the COU to the Cross Reference Table (without archiving), the following code can be used:

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

```
DELETE FROM ABP_XREF WHERE XREF_KEY IN (SELECT distinct XREF_KEY FROM  
ABP_XREF_COU WHERE CHANGE_TYPE != 'I');
```

Insert the new records and the updated records:

```
INSERT INTO ABP_XREF SELECT * FROM ABP_XREF_COU WHERE CHANGE_TYPE != 'D';
```

The Cross Reference table does not have the ability to hold historical records as it does not have a logical status code attribute. Therefore, to capture the historical records, you will need to create an archive table which is populated when records are either deleted or updated.

To apply the COU to the Cross Reference table (with archiving), the following code can be used:

The following command creates an archive table of records which are being updated and deleted from the existing Cross Reference table. If this table already exists, you can simply use INSERT INTO rather than CREATE TABLE:

```
CREATE TABLE ABP_XREF_ARCHIVE AS SELECT * FROM ABP_XREF WHERE XREF_KEY IN  
(SELECT DISTINCT XREF_KEY FROM ABP_XREF_COU WHERE CHANGE_TYPE != 'I');
```

The following command then deletes the records from the existing table that are either updates or deletes:

```
DELETE FROM ABP_XREF WHERE XREF_KEY IN (SELECT DISTINCT XREF_KEY FROM  
ABP_XREF_COU WHERE CHANGE_TYPE != 'I');
```

The following command then inserts the new insert records and the updated records:

```
INSERT INTO ABP_XREF SELECT * FROM ABP_XREF_COU WHERE CHANGE_TYPE != 'D';
```

II. Creating a single-line or multi-line address using AddressBase Premium products

II.1 Delivery Point Address vs. Geographic Address

The AddressBase Premium products contain a variety of data fields which allow a user to construct, for a given addressable object, different forms of an address dependent on how the address is to be used.

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

- Delivery Point Address
- Geographic Address

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

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

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 aid, 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.

High-level data model

The AddressBase Premium data model accommodates both the Delivery Point Address and the Geographic Address by linking them using the unique property reference number (UPRN) as the key.

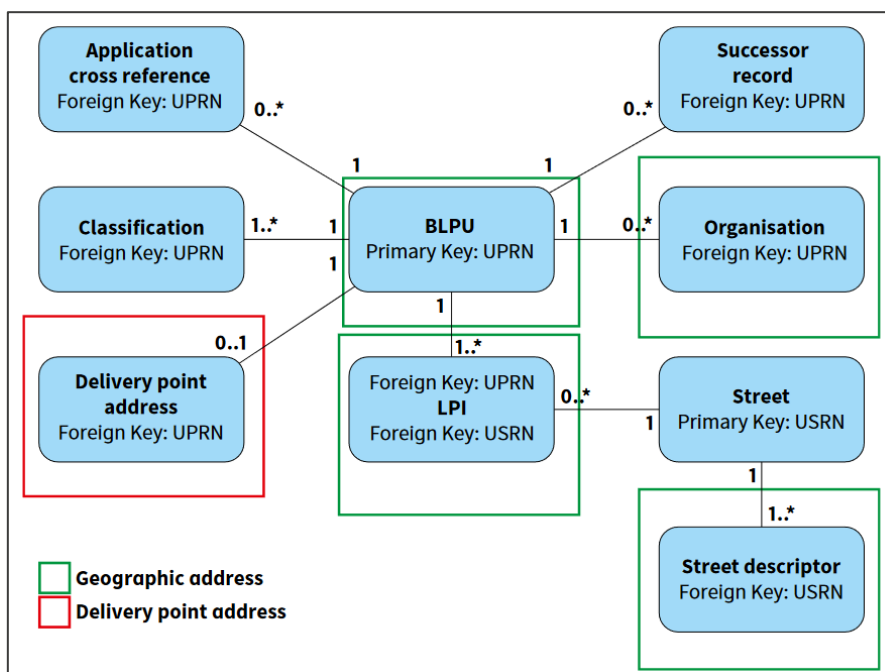


Figure 4: AddressBase Premium high-level data model with address 'types' highlighted.

It is important to note the cardinality differences that the Geographic and Delivery Point Address components have with the Basic Land and Property Unit (BLPU):

- The relationship between the Delivery Point Address and the BLPU is 0..1 – 1.
- This means that the Delivery Point Address is an optional component, so a Delivery Point Address will only be created when it has been matched to the Geographic Address. Moreover, only one Delivery Point Address can be matched to a BLPU.
- The relationship between the Land and Property Identifier (LPI) and the BLPU is 1..* – 1
- This means that the LPI component is mandatory; therefore, at least one LPI must exist for each BLPU. Moreover, there can be more than one LPI linked to a single BLPU.

Together, these differences mean that there are more Geographic Addresses in the product than there are Delivery Point Addresses, because:

- a. Not every BLPU has a Delivery Point (postal) Address, only those that have been matched to the Royal Mail PAF database.
- b. A single BLPU can have only one Delivery Point Address.
- c. A single BLPU can have more than one Geographic Address (because alternative and historical addresses are available in AddressBase Premium).

11.2 Background to single-line and multi-line address labels

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: single line and multi-line. The simplest label 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), for example:

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

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

ROSE COTTAGE
5 MAIN STREET
ADDRESSVILLE
LONDON
SE99 9EX

The following sub-sections outline a methodology for structuring and layering a single address label using AddressBase Premium. The rules outlined 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.

11.3 Delivery Point 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 listed in Table 7 can be used to build a Delivery Point Address label.

Table 7: Delivery Point Address components.

Delivery Point Address component	Type
DEPARTMENT_NAME	Character
ORGANISATION_NAME	Character
SUB_BUILDING_NAME	Character
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 (listed in Table 7). The following examples will use the English version of these attributes.

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 illustrates how these fields should be ordered in a single address label:

Delivery Point Address component	Example
DEPARTMENT_NAME	CUSTOMER SERVICE DEPARTMENT
ORGANISATION_NAME	JW SIMPSON LTD.
SUB_BUILDING_NAME	UNIT 3
BUILDING_NAME	THE OLD FORGE
BUILDING_NUMBER	7
PO_BOX_NUMBER	
DEPENDENT_THOROUGHFARE	RICHMOND TERRACE
THOROUGHFARE	MAIN STREET
DOUBLE_DEPENDENT_LOCALITY	HOOK
DEPENDENT_LOCALITY	WARSASH
POST_TOWN	SOUTHAMPTON
POSTCODE	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, as shown in the following example:

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

11.3.1 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 7, 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.

Examples of SQL logic to create a single-line Delivery Point Address are on our [GitHub repository \(https://github.com/OrdnanceSurvey/AddressBase/tree/master/SQL\)](https://github.com/OrdnanceSurvey/AddressBase/tree/master/SQL), which incorporates the following elements:

- 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.

11.3.2 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:

- Generally, if there is a building number, it should appear on the same line as the thoroughfare (or dependent thoroughfare). If there is no thoroughfare information, the building number should appear on the same line as the first locality line.
- 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 (or on the first locality line if no thoroughfare is present).
- In most other cases, the building name, if present, should appear on a separate line above the thoroughfare name or dependent thoroughfare or locality line if no thoroughfare is present.
- 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 to 42 of the [PAF Programmers' Guide](https://www.royalmail.com/marketing-services/address-management-unit/address-data-products/programmers-guide) (<https://www.royalmail.com/marketing-services/address-management-unit/address-data-products/programmers-guide>).

11.4 Geographic Address (Local Authority Address)

The structure of a Geographic Address is based on the British Standard BS7666 and is split into a number of 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 products, the core property-level address information is stored within the Primary Addressable Object (PAO) and Secondary Addressable Object (SAO) fields of the LPI table. The additional attribution required to build a full address label is maintained in the BLPU (postcode_locator), ORGANISATION (organisation) and STREET_DESCRIPTOR (street_description, locality_name, town_name, administrative_area) tables.

For a full description of PAOs and SAOs, and the complete set of AddressBase fields, please refer to the relevant [AddressBase Premium Technical Specification](#).

11.4.1 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 listed in Table 8 should be used to build a Geographic Address label.

Table 8: Geographic Address components.

Table	Geographic Address Component
Organisation	ORGANISATION
LPI	SAO_TEXT
LPI	SAO_START_NUMBER
LPI	SAO_START_SUFFIX
LPI	SAO_END_NUMBER
LPI	SAO_END_SUFFIX
LPI	PAO_TEXT
LPI	PAO_START_NUMBER
LPI	PAO_START_SUFFIX
LPI	PAO_END_NUMBER
LPI	PAO_END_SUFFIX
Street Descriptor	STREET_DESCRIPTION
Street Descriptor	LOCALITY
Street Descriptor	TOWN_NAME
Street Descriptor	ADMINISTRATIVE_AREA*
BLPU	POSTCODE_LOCATOR

*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 build the address using the alternate language for SAO_TEXT, PAO_TEXT and Street Descriptor entries. This can be achieved by filtering on the language field of the LPI and Street Descriptor tables. The same order as above would be applicable.

11.4.2 Linking address components

The LPI table includes the PAO and SAO fields. However, in order to obtain the rest of the address, it is necessary to join the LPI table to the Street Descriptor table to pick up the street name, locality and town information (using the USRN as the key), and also to the Organisation and BLPU tables (using the UPRN as the key) to pick up the organisation names and postcodes, respectively.

Figure 5 shows the links that need to be made in order to build a full Geographic Address from the different BS7666 components in AddressBase Premium.

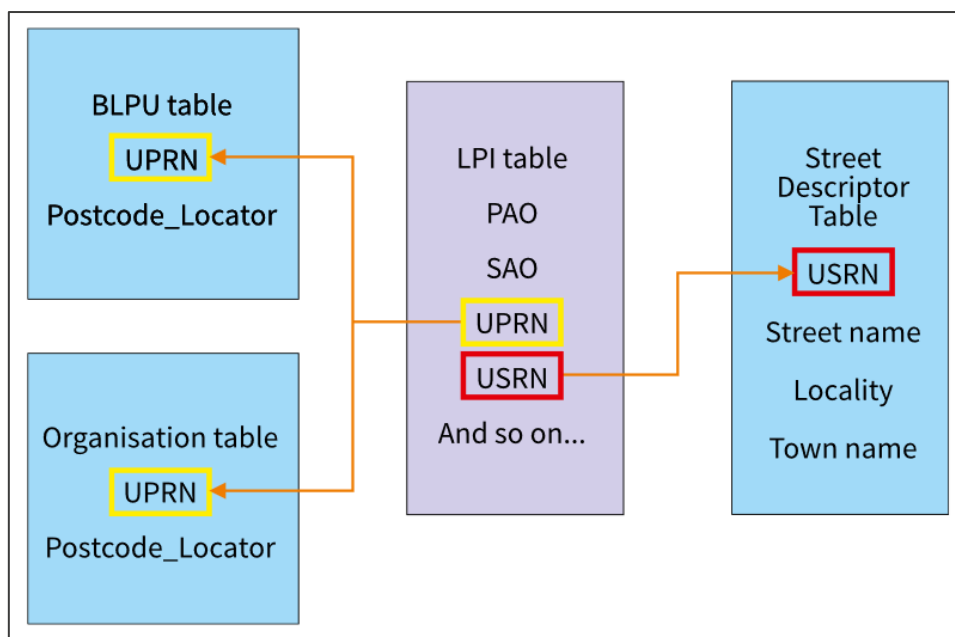


Figure 5: Table join logic for Geographic Address label.

Using the LPI table as a starting point, the remaining address components can be picked up using table joins to the other tables on UPRNs and USRNs. Note that there can be more than one LPI for each UPRN, so if only one address is required per BLPU, the LPI with logical_status = 1 (approved) should be selected (there can be only one approved LPI per BLPU).

11.4.3 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	1	1	1	1
PAO_START_SUFFIX		A		A
PAO_END_NUMBER			5	5
PAO_END_SUFFIX				C
Rendered PAO range	1	1A	1-5	1A-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.

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, as shown in the following table:

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 primary addressable objects (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.

11.4.4 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 is often the case. The following table shows an example where the administrative area name (in this case, BURY) has been included and excluded from a single-line address:

Administrative area not included	34, CROW LANE, RAMSBOTTOM, BL0 9BR
Administrative area included (BURY)	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 illustrates how they should be ordered in a single address label.

Table 9: Geographic Address formatting.

Geographic Address Component	Example
ORGANISATION	JW SIMPSON LTD
SAO_TEXT	THE ANNEXE
SAO (number / range string)*	IA
PAO_TEXT	THE OLD MILL
PAO (number / range string)*	7-9

Geographic Address Component	Example
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
ADMINISTRATIVE_AREA	'SOUTHAMPTON'	
POSTCODE_LOCATOR	'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)*	'I'	I
STREET_DESCRIPTION	'HORSHAM LANE'	HORSHAM LANE
LOCALITY_NAME	null	
TOWN_NAME	'HORSHAM'	HORSHAM
ADMINISTRATIVE_AREA	'HORSHAM'	* Duplicate name omitted
POSTCODE_LOCATOR	'RH12 1EQ'	'RH12 1EQ'

11.4.5 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 pre-format the SAO and PAO number / range strings and join tables together. 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 8, 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 \(https://github.com/OrdnanceSurvey/AddressBase/tree/master/SQL\)](https://github.com/OrdnanceSurvey/AddressBase/tree/master/SQL), which incorporates the following elements:

- 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 want no duplicate UPRNs to be returned an additional DISTINCT line needs to read DISTINCT(I.UPRN)

11.4.6 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:

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

Example

11A MAIN STREET

- 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, 11A MAIN STREET

- 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 line 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, 11A MAIN STREET

- 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, 11A MAIN STREET	1A ROSE COURT, 11A MAIN STREET

- 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, 11A MAIN STREET

- 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, 11A MAIN STREET

- 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

- 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

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

11.5 Advice on creating mailing lists using AddressBase Premium products

It's possible to create mailing lists using the AddressBase Premium and AddressBase Premium Islands products (you can also create them using AddressBase Plus and AddressBase Plus Islands). Given that the AddressBase Premium products contain two different types of address, a decision needs to be made on whether to use the Geographic or Delivery Point Addresses, or a mixture.

The following two options should be considered:

- Use Delivery Point Addresses whenever they are available, and when they are not, use a Geographic Address.
- 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, 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.

11.5.1 Address status

When building your query to extract a mailing list, it is important that you consider filtering your results based on the address status and type. The status of an address is often something that needs to be considered when working with address data. Questions need to be answered before AddressBase Premium can be used effectively, such as "Is the addressable object in planning, being constructed, current, demolished or accurately positioned?".

The table below offers guidance on what status filters should be considered. Please see the [AddressBase Premium Technical Specification](#) on our website for more information about each of these attributes.

Table 10: Guidance on what status filters should be considered when building a query to extract a mailing list using AddressBase Premium.

Status attributes	Table	Use	Values
LOGICAL_STATUS	BLPU	Describes where a land or property unit is in its lifecycle.	1 = Approved 6 = Provisional 8 = Historical
LOGICAL_STATUS	LPI	Describes where an address is in its lifecycle.	1 = Approved 3 = Alternative 6 = Provisional 8 = Historical
BLPU_STATE_CODE (optional)	BLPU	Informs the user what physical state the land or property is in (for example, 'under construction', 'in use', 'demolished').	1 = Under construction 2 = In use 2 = Unoccupied 4 = No longer existing 6 = Planning permission granted

Status attributes	Table	Use	Values
			Null = Unknown or N/A
RPC_CODE	BLPU	To ascertain how accurate the coordinate is. Use in conjunction with the postcode_locator field to understand the accuracy of the address' position.	1 = Visual centre 2 = General internal point 3 = SW corner of 100m grid ref 4 = Start of referenced street 5 = Postcode unit point 9 = Centre of Local Authority area
ADDRESSBASE_POSTAL	BLPU	This field can be used to limit your records based on whether they are capable of receiving mail or not.	D = A record which is linked to PAF C = A record which is postal and has a parent linked to PAF L = A record identified as postal via Local Authority information N = Not a postal address
LANGUAGE	LPI STREET_DESCRIPTOR	This information can be used to limit your records based on the language.	ENG = English CYM = Welsh GAE = Gaelic

11.5.2 Other filters available in AddressBase Premium products for use in addressing labels

AddressBase Premium is a rich addressing dataset that contains a wealth of other attributes that could be used in conjunction with address labels. For example:

- Classification can be used to target certain types of property.
- OS MasterMap Topography TOID cross references can be used to link address labels to Topographic objects and viewed in a GIS for Great Britain (AddressBase Premium Islands does not have OS MasterMap Topography TOID cross references).

12. 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 section will step through two such approaches that may be used when working with AddressBase Premium products: free text search and structured component search.

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

12.1 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:

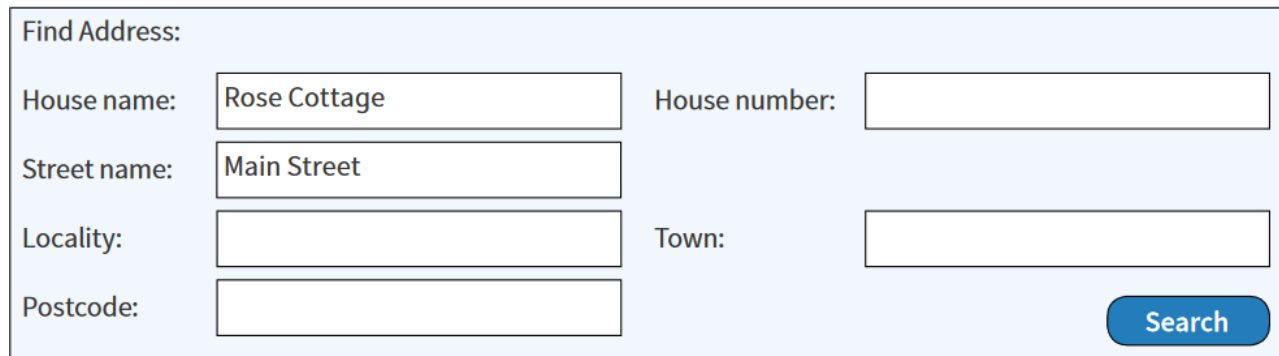
Find address	Results
CLOVER AVENUE, SW99 9ZZ	1, Clover Avenue, Fieldtown, Addressville, SW99 9ZZ 2, Clover Avenue, Fieldtown, Addressville, SW99 9ZZ 3, Clover Avenue, Fieldtown, Addressville, SW99 9ZZ 4, Clover Avenue, Fieldtown, Addressville, SW99 9ZZ 5, Clover Avenue, Fieldtown, Addressville, SW99 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 they may have been 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.

12.2 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"/>	

Figure 6: An example of a structured address component 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 also 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?

The following sub-sections suggest how to implement the two search methods described above. Both methods should be used alongside the instructions on formatting single address labels given in [Section 11](#).

12.3 Understanding the different addresses available

As described in [Section 11](#), at a high level, the AddressBase Premium products provide two different types of address: the Delivery Point Address and the Geographic Address. However, for some Geographic Addresses, an alternative, provisional or historical variant of the approved record may also be provided as well as the approved address (all sharing the same UPRN). Table 11 outlines what these addresses are and how to access them in the products.

Table 11: A breakdown of the location and definition of Delivery Point addresses and the four categories of Geographic Addresses available in AddressBase Premium products.

Address type	What is it?	Where is it?
Delivery Point Address	The postal address as assigned to the property by Royal Mail (and widely used by the public).	Delivery Point Address table.
Approved Geographic Address	The legal / approved address as assigned by the local naming and numbering authority.	LPI table with Logical Status = 1, joined to Street Descriptor, Organisation and BLPU tables.
Provisional Geographic Address	Provisional addresses may exist for a property from the moment that an address has been granted planning permission to be built to the time when construction has been completed.	LPI table with Logical Status = 6, joined to Street Descriptor, Organisation and BLPU tables.
Alternative Geographic Address	Any alternative addresses that may exist for this property (for example, alternative names). There may be more than one alternative address per property.	LPI table with Logical Status = 3, joined to Street Descriptor, Organisation and BLPU tables.
Historical Geographic Address	Any historical addresses (recorded since data collection began) that may have existed in the past for this property (for example, previous house names or business names, and so on). There may be more than one historical address per property.	LPI table with Logical Status = 8, joined to Street Descriptor, Organisation and BLPU tables.

12.4 The search operation

An address search operation typically requires two stages of interaction from a user, and several processing steps from the underlying IT system. These steps are summarised in Figure 7:

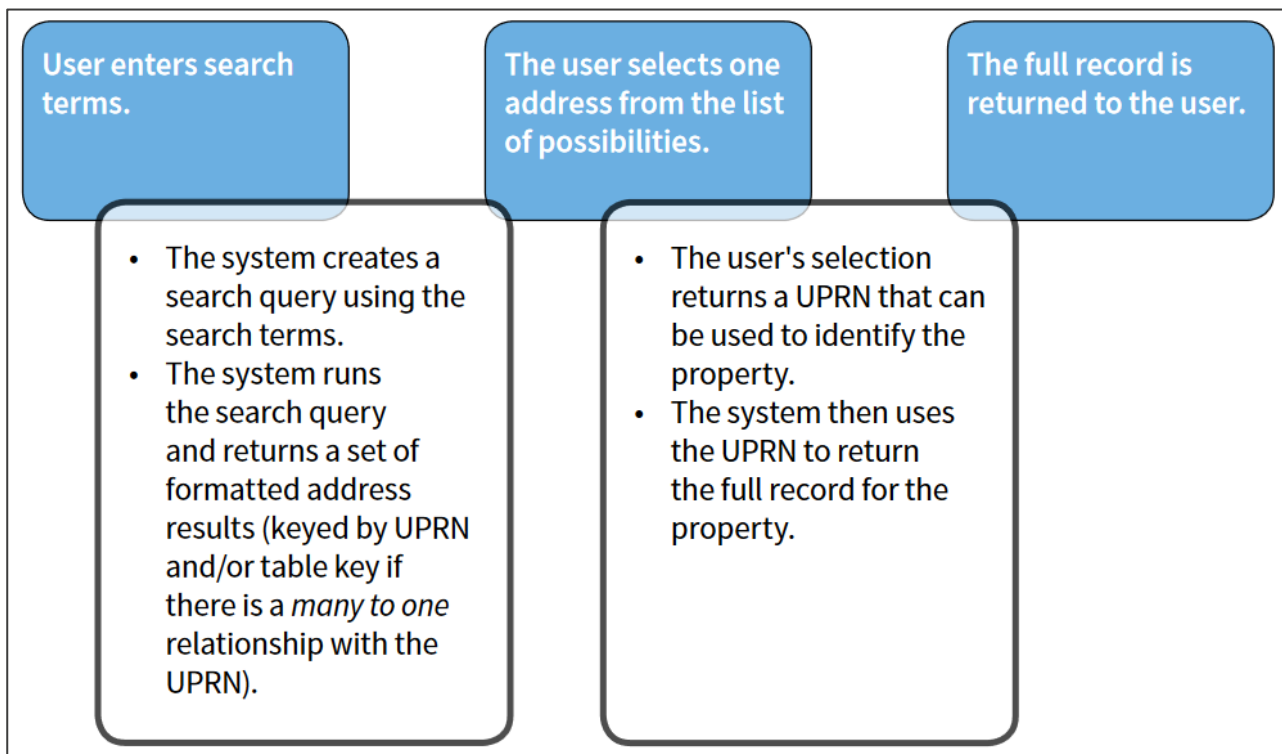


Figure 7: Summary of the steps needed to successfully implement an address search operation.

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.

12.5 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 the AddressBase Premium products. 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 (Delivery Point Address and Geographic Address) 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 Premium and AddressBase Premium Islands are as follows:

Table 12: The relationships between some common search fields and the fields found in AddressBase Premium and AddressBase Premium Islands.

Search Box	Mapped Delivery Point fields	Mapped Geographic fields
Business Name	Organisation_Name	Organisation PAO_Text SAO_Text
Flat / Subdivision Name	Sub_Building_Name Department_Name	SAO_Text
Flat / Subdivision Number	Sub_Building_Name	SAO_StartNumber SAO_StartSuffix SAO_EndNumber SAO_EndSuffix
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 break down 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 in the table above), 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.

12.6 Structuring the query for a structured address search

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

- Ignore any search boxes that are not filled in with values.
- 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) where a street, locality and town name have been entered by the user:

```
SELECT dp.UPRN, GetFormattedAddress(dp.*) FROM abp_delivery_point 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 11](#) 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 dp.UPRN, GetFormattedAddress(dp.*) FROM abp_delivery_point 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)
UNION
SELECT b.uprn, GetFormattedAddress(b.*, l.*, s.*, o.*) FROM abp.blpu b INNER JOIN abp.lpi l ON l.uprn = b.uprn
INNER JOIN abp.street_descriptor s ON s.usrn = b.usrn LEFT JOIN abp.organisation o ON o.uprn = b.uprn WHERE
(s.street_name = streetsearchtext OR l.pao_text = streetsearchtext) AND
(s.locality = localitysearchtext OR s.town = localitysearchtext OR s.street_name = localitysearchtext)
AND (s.town = townsearchtext OR s.locality = townsearchtext)
```

The geographic query requires four joins between the BLPU, LPI, Street Descriptor and Organisation tables in order to access all the fields required to build an address.

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: formatted addresses along with their UPRN. Exact duplicates will be omitted, but all ‘variations’ of the same address will be outputted (one row for each variation, with the same UPRN repeated more than once potentially). It may be wise to also return the ‘logical status’ and / or ‘postal address flag’ values against each to enable further filtering (that is, to include or exclude historical addresses for example, or to restrict the results to postal addresses only).

12.7 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. 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:

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

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):

SQL code	Description
<code>UPPER(dp.post_town) = UPPER(townsearchtext)</code>	Case insensitive equality

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

SQL code
<code>UPPER(dp.post_town) LIKE ('%' UPPER(townsearchtext) '%')</code>

12.8 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.

12.9 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 'logical status' values, which would allow the results to be filtered on 'approved', 'provisional' and 'historical' statuses, for example).

Table 13 shows a possible address index table structure:

Table 13: A possible address index table structure.

UPRN	Address Text	Statuses (multiple fields)
123456789012	4 THE MEADOWS HIGH STREET WALTHAMSDALE BURRIDGE BU27 9UB	Approved
123456789012	FLAT 4 THE MEADOWS HIGH STREET WALTHAMSDALE BURRIDGE BU27 9UB	Alternative + PAF
123456789012	4 HIGH STREET WALTHAMSDALE CLOSE BURRIDGE BU27 9UB	Historical
947364758903	ROSE COTTAGE MAIN STREET HAVERSHAM SUDBURY SU45 9TY	Approved + PAF
947364758903	ROSE FARMHOUSE MAIN STREET HAVERSHAM SUDBURY	Historical

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, current and historical, approved and alternative) 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.

12.10 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 the user input 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. Table 14 shows how the index table would be used in the above example to return relevant results:

Table 14: Index table for the 4 HIGH STREET WESTVILLE WV17 example.

UPRN	Address Text	Statuses (multiple fields)
894756389092	4 HIGH STREET WESTVILLE SUNNYTOWN WV17 7HL	Approved + PAF
894756389092	ROSE COTTAGE 4 HIGH STREET WESTVILLE SUNNYTOWN WV17 7HL	Alternative
894756389092	ROSE COTTAGE HIGH STREET WESTVILLE SUNNYTOWN WV17 7HL	Alternative

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

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 section.

12.11 Summary of Section 12: Searching for addresses

This section introduces implementing address search functionality using AddressBase Premium products. The main points are summarised below:

- 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.
- 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.
- With a structured search interface, the addresses can be queried directly by mapping the various address fields to the text boxes supplied.
- 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.
- It is possible to filter results by status, for example, ‘approved’, ‘alternative’ and ‘historical’, as well as ‘postal’ or ‘non- postal’, etc.
- Any search function should search all forms of an address (both Geographic and Delivery Point Addresses).
- Careful consideration should be given to the use of ‘fuzzy’ search algorithms (such as using wildcard or sound-alike searches).

Annex I: Support links

Product	Link to support documentation on the OS website
AddressBase Premium technical support documents	https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-premium-support
AddressBase Premium Islands technical support documents	https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-islands-support

