

Enriching the address based data with UPRN: learning from Domestic EPC and Land Registry PPD

Technical Report

Bin Chi, Mark Livingston, Nick Bailey



1. Introduction

The opening up of Ordnance Survey's two geospatial identifiers, Unique Property Reference Numbers (UPRNs) and Unique Street Reference Number (USRN) under an Open Government Licence (OGL), has created a huge opportunity for those that work in areas with address-based data. It creates the ability to accurately bring together these data easily and accurately at property level.

The current landscape for property-based data is fragmented by the lack of standardised address identifiers and the inconsistencies in how address information is captured. This creates a barrier to the linkage of data related to the same address, making research and analysis difficult. Even important national property datasets like the Department for Levelling Up, Housing and Communities (DLUHC) Domestic Energy Performance Certificates (Domestic EPCs) or the Land Registry (LR) Priced Paid Data (PPD) store address information using different conventions or structures. Combining data on house sales or rents with energy performance, and perhaps with energy usage, age of the property and other locational characteristics could help fill some of the most pressing gaps in our understanding of housing, but it is very challenging without standard identifiers. The problem is by no means unique to the UK.¹

Much personal or individual data also contains address information which can be a valuable means to link related data. In Scotland, for example, address information within health records has been used to identify household relationships which are otherwise largely absent from these systems.² Similar work has been undertaken in Wales.³ Here too, successful linkage depends upon the quality of address information and how it is structured.

We are aware of several different attempts to tackle the address-to-UPRN matching problem. As well as the deterministic approaches used by work mentioned in the previous paragraph, others have applied fuzzy matching techniques or a combination of the two. In this work, we focus on our own deterministic algorithm.

The current UK's geospatial strategic vision is to build up "a coherent national location data framework" by 2025. To this end, and in response to the Open Standards Board's call, OS have made Unique Property Reference Numbers (UPRNs) and Unique Street Reference Number (USRN) available under an Open Government Licence. Nowadays, UPRN and USRN are "the public sector standard for referencing

¹ One US example of a commercial service to provide standardised identifiers - <https://www.placekey.io/blog/the-most-common-address-standardization-problems-and-what-you-can-do>

² Clark D, Dibben C. A guide to CHI-UPRN Residential Linkage (CURL) file. Scottish Centre for Administrative Data Research and Public Health Scotland; November 2020. Available from: <https://www.isdscotland.org/Products-and-Services/EDRIS/docs/CURL-Report-November-2020.pdf>

³ Harper, G., Boomla, K., Robson, J., Stables, D., Ahmed, Z., Fry, R., and Dezateux, C. (2020) Allocating Unique Property Reference Numbers to Patient Addresses Using A Deterministic Address-Matching Algorithm: Evaluation of Accuracy, Match Rate and Bias, *International Journal of Population Data Science* 5 (5).

and sharing property and street information”⁴. The change to both these geospatial identifier numbers has increased the research focus in this area, particularly in an academic context where UPRN and USRN linkage to historic data significantly extends the prospects for longitudinal build environment analyses related to, for example, energy performance improvements or the dynamics of housing affordability.

Ordnance Survey (OS) have a well-managed system which oversees the production of UPRNs for every addressable location in the UK, both residential and commercial. However, few systems which collect address based data use these UPRNs from the outset. Instead, address information is stored in a variety of more or less structured ways. Unfortunately, there is no consistent format for the recording of address data in the UK, which makes accurate attribution of a UPRN difficult. Flats are particularly problematic, with many ways of recording a flat’s position in a building. Different organisations have developed or are developing a variety of methods to identify and add UPRNs to data, including fuzzy logic address matching, machine learning and rules-based approaches^{5,6}. In Scotland, for example, the Improvement Service’s Datahub offers a service for local authorities and others to attach UPRNs to any address based information.⁷

This short project sets out to examine how address based data can be integrated more effectively by identifying limitations of current address capture and making recommendations for the future. We will achieve this by applying rules-based approaches developed by one of the authors to two existing datasets, the Domestic EPCs and the LR PPD. We explore how far it is possible to reliably identify the correct UPRN. With the EPC data, we compare our results with those available from the data owner’s efforts. The EPC data have recently had UPRNs attached retrospectively via an undisclosed algorithm containing a “combination of rules-based and machine-learning approaches”⁸. Since September 2021, EPC assessors have been asked to add UPRNs when creating a record.

1,1 Aims

- To apply rules-based approaches to attach UPRNs to Domestic EPC and LR PPD datasets in England and Wales.
- To compare matched UPRNs from DLUHC and our own methodologies, summarising and accounting for any differences.
- To understand what causes failed record linkage in Domestic EPC and LR PPD.
- To make suggestions for improvements in the recording of address based data to improve future success rates in the processing of current address data.

⁴ Identifying property and street information: <https://www.gov.uk/government/publications/open-standards-for-government/identifying-property-and-street-information>

⁵ Office for National Statistics. ONS working paper series no 17 - Using data science for the address matching service. Available from: <https://www.ons.gov.uk/methodology/methodologicalpublications/generalmethodology/onsworkingpaperseries/onsworkingpaperseriesno17usingdatasciencefortheaddressmatchingservice#authors>

⁶ Harper, G., Stables, D., Simon, P., Ahmed, Z., Smith, K., Robson, J., and Dezateux, C. (2021) Evaluation of the ASSIGN open-source deterministic address-matching algorithm for allocating Unique Property Reference Numbers to general practitioner-recorded patient addresses, *International Journal of Population Data Science* 6 (1).

⁷ <https://datahub.scot/home/>

⁸ <https://news.opendatacommunities.org/energy-performance-certificates-now-include-uprn/>

[Type here]

Background

As part of the second mission in UK's geospatial strategy, Geospatial Commission opened two Ordnance Survey's Geospatial identifiers, Unique Property Reference Numbers (UPRNs) and Unique Street Reference Number (USRN) under an Open Government Licence (OGL). Later in the same year, the Open Standards Board has mandated that UPRN and USRN are "the public sector standard for referencing and sharing property and street information". UPRN is designed to comply with the requirement of British Standard BS7666. Every UK address record has a unique UPRN. The relationship property's addresses and UPRN can be found in Ordnance Survey (OS)'s AddressBase products. OS first published three level AddressBase datasets (AddressBase, AddressBase Plus and AddressBase Premium) in 28/9/2011. OS then keeps updating these three products every six weeks and recently add another three AddressBase products to widely benefit its users. The top 2 detailed OS AddressBase products are OS AddressBase Premium and OS AddressBase Plus. The former contains the most detailed level UPRN information. It is the only AddressBase product containing historical and current UPRNs with the detailed property's lifecycle information. Meanwhile, the later contains all the current active UPRNs.

Populating UPRN in addressing data not only results in better location data, but also good for greater reliable data connectivity. First, UPRNs as machine readable codes offer greater confidence in the accuracy of property addresses. This overcomes the inherent challenges of using the traditional addressing format with postcode and address string. The address of the same property can be recorded slightly differently within the same datasets or even between datasets. For example, "5, PARKERS CLOSE, BS10 7PH" vs "5 PARKERS CLOSE, SOUTHMEAD, BRISTOL, BS10 7PH" vs "5 PARKER CLOSE, SOUTHMEAD, BS10 7PH". This issue creates an obstacle to exchanging information between datasets, or even extracting the duplicate address records within the same dataset.

Driven by the UK's geospatial strategy, the need for geocoding the address base data in the UK is ever increasing. Different data owners within UK may have data with different quality, especially for the address related. There is practical challenge to tag the UPRN for the currently un-UPRN-tagged address datasets. Some of the government has taken this challenge with different degree to success. Take Domestic Energy Performance Certificates (EPCs) data as a typical case. Both of the Scottish government and Department of Levelling up, Housing and Communities (DLUHC) separately start to publish Domestic EPCs with UPRN information in 2021. However, neither of them achieved a 100% success rate in tagging UPRN. Domestic EPCs in Scotland shows a 94% success rate, while Domestic EPCs in England and Wales shows a 92% success rate. Although a 100% success rate is probably not realistic, a higher success rate for these Domestic EPC datasets can potentially better unlock the power of the EPC data. To better investigate the possibility to geocoding the address data with UPRN in England and Wales, two housing related case studies were conducted. One is highly UPRN tagged data, Domestic EPC. The other is the un-UPRN-tagged dataset, Land Registry Price Paid Data (PPD).

2. Data and Methods

3.1 Data

In this research we tag two residential address based datasets with UPRNs. The first, DLUHC's Domestic EPCs, provides energy performance certificates for all properties surveyed in England and Wales. The second is the Land Registry PPD which contains data on all house sales in England and Wales. Since 2021, DLUHC have started to publish Domestic EPCs with UPRN. Currently at time of writing the Land Registry PPD does not contain UPRNs. Both datasets recorded property level information with the property's postcode and detailed addresses as address strings. Figure 1 lists all the address related field in the two datasets. It is obvious that address information recorded in these two address based datasets are different. The different record structures may influence the geo-tag UPRN linkage process.

Domestic EPCs		LR PPD	
Explanation	Fields	Fields	Explanation
The postcode of the property. e.g. NW6 2AG	POSTCODE	POSTCODE	The postcode of the property e.g. NW6 2AG
First line of the address. e.g. Flat 12 Beacon House	ADDRESS1	SAON	Secondary Addressable Object Name. Where a property has been divided into separate units (for example, flats), the PAON (above) will identify the building and a SAON will be specified that identifies the separate unit/flat. e.g. FLAT 12
Second line of the address. e.g. Hemstal Road	ADDRESS2	PAON	Primary Addressable Object Name. such as the house number or name. e.g. BEACON HOUSE
Third line of the address. e.g. NULL	ADDRESS3	Street	e.g. HEMSTAL ROAD
Field containing the concatenation of address1, address2 and address3. e.g. Flat 12 Beacon House, Hemstal Road	ADDRESS	Locality	e.g. LONDON
Describes the type of building that is being inspected. Based on planning use class. e.g. Flat	PROPERTY_TYPE	Town/City	e.g. LONDON
		District	e.g. CAMDEN
		County	e.g. GREATER LONDON
		Property type	Indicates the type of house. e.g. F(Flats/Maisonettes)

Figure 1. Address related fields in Domestic EPC and LR PPD

The Domestic EPCs dataset used is Version 9 (Figure A in Appendix A) published by DLUHC in November 2021. It contains a total of 21,857,699 Domestic EPC records for England and Wales from 1/10/2008 to 30/9/2021. This is the first published version with UPRN information, with a UPRN match rate of 92.51%. Figure 2 shows the UPRN source in this version. 86.14% in the 92.51% UPRN match rate are achieved by DLUHC's address matching algorithm. The rest 7.48% UPRN match rate is submitted by an energy assessor. This Domestic EPC are regularly updated by DLUHC. During the project research period, two newer versions were published with different UPRN match rates. These three Domestic EPC data sets have the same address format, which enables the reuse of any address matching algorithms and code.

[Type here]

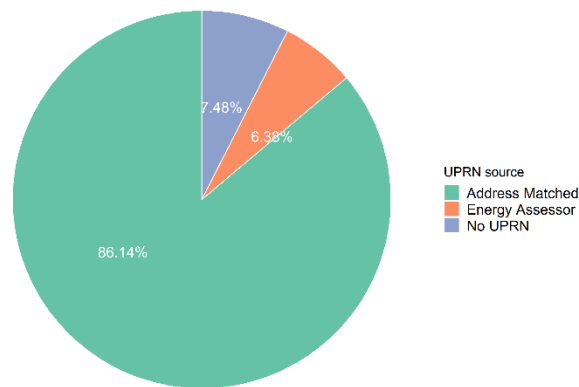


Figure 2 The UPRN sources in the ninth version of Domestic EPCs

The Land Registry PPD dataset used in this research was downloaded on March 2022. The dataset contains records of 26,883,169 transactions between 1/1/1995 and 28/1/2022 in England and Wales. Similar as Domestic EPCs. These administrative datasets are regularly updated by LR, the form of storing property address in the same dataset keeps same. This allows the data linkage method to be continuously usable.

In this research we use the OS AddressBase Plus which contains all the current or active UPRNs at the time of production (cut time). The version we have used for this research is Epoch Number 90⁹. The publication date for this version is January 2022 and the data cut date is December 2021. This OS AddressBase Plus has a total of 37,385,807 UPRNs/property addresses for the whole of Great Britain. UPRNs have a hierarchical structure with a ‘parent’ UPRN for the whole block or property address shell, and ‘child’ UPRNs for individual units or flats within it. Twenty percent of UPRNs recorded in AddressBase Plus are child UPRNs.

3.2 Data linkage method

There are three commonly used address matching methods for UPRN tagging task. The first is rules-based address matching. It directly determines a matching based on a variety of matching rules. The second is fuzzy address matching, which determines a matching with varying degrees of certainty. Fuzzy matching is useful especially when there are partial differences in address information, such as “11, WALCOTT GREEN” vs “11, WALCOT GREEN”. Fuzzy address matching is good at handling the incorrect addresses but could cause incorrect matches. The combination of rules-based address matching and then fuzzy addressing matching can reduce the incorrect matches created through using fuzzy matching alone. The rule-based approach is arguably the

⁹ AddressBase products epoch dates

<https://www.ordnancesurvey.co.uk/business-government/tools-support/addressbase-epoch-dates>

most accurate method, because it is a defined process, rather than a best guess. It also can solve the incorrect addresses by manually correcting the errors. This is time consuming as it requires the creation of multiple rules and the need to manually check some records. To best control the linkage quality and learned lesson, this research we use rules-based address matching.

OS AddressBase products record the relationship between UPRN and the property's addresses. Each record in AddressBase Plus includes a UPRN and the address broken into several address fields. We have treated each address related field in OS AddressBase product as the gold standard for address strings. In contrast, addresses in both the EPC and PPD are in different formats. The UPRN linkage method used by us is a rules-based matching process. In detail, it matches the address string for each property in the target dataset with the available "standard" address string for the same property address in the OS AddressBase products. As OS AddressBase datasets also provide the relationship between UPRN and property's "standard" addresses, the successfully linked result will automatically be assigned with the associated UPRN.

The rules-based linkage method is created by exploiting the address-matching experience of the authors. As part of the linkage, this method uses different controls to correct known address errors but also to maximise the likelihood of identifying the correct address. So for instance an incorrect address string like "CASPAR HOUSE" might be changed to "CASPER HOUSE". We also carry out the linkage at different postcode levels (postcode first and then postcode sector) to capture addresses which have been recorded with different postcodes. It is common for flats to have errors of this nature. For example, the property address for "1, BOUNDARY PLACE" in one dataset is recorded in GL19 3RF, while the other dataset recorded it in GL19 3RG. Also, it can be necessary to ignore part of the address component and relink to achieve a successful link. This is particularly useful for address-matching in Wales, when street level name is written in different languages or different strings. This is also usefully for potential real case that recorder use different street name when a building is located at the cross of two streets. This is an iterative process which requires running initial and subsequent linkages and checking to identify failed links. We do this by randomly checking records where the linkage has failed, concentrating our checks in the postcodes which have the most failed links, this enables the final linkage process to cover all the potential linkage methods. Finally, from past experience we know that we can not concentrate our checking in one sub-area but need to sample across the whole of England and Wales. This is because address conventions differ across the UK and focusing in one area will ignore different conventions for recording addresses. To best understand how easy it is to attribute UPRNs to address information, the case studies here consider the whole dataset rather than focusing on a subset of the data.

We focus only on the UPRNs recorded in AddressBase Plus, which only have active UPRNs, rather than AddressBase Premium which also contains retired (or historical) UPRNs. This reduces the complexity of the UPRN address matching. Thus, we always tag a child UPRN where possible. We remove all parent UPRNs where a property also has a child UPRN. As a result, 1,275,370 parent UPRNs are removed from the data (3.41%).

3.2.1 Attaching UPRN in Domestic EPCs

Figure 3 illustrates the process of address matching OS AddressBase Plus addresses to Domestic EPC addresses. As the address structures are different, we build a two-step linkage process. We

[Type here]

first capitalise four address fields (i.e. ADDRESS1, ADDRESS2, ADDRESS3 and ADDRESS) in the EPC dataset. In the second stage, we create a single address string field to represent the property's address by considering all the possible combination of the address fields in both datasets (OS AddressBase Plus and Domestic EPCs). These strings are then compared across the two data sets.

We conduct matching at a postcode level and where these fail we then carry out matching at postcode sector. This process creates 446 detailed matching rules. Of the 446 match rules, 336 are conducted at the postcode level, the remaining 109 match rules are conducted at postcode sector level or even higher. Details of the match rules are listed in Table B1 in the Appendix B.

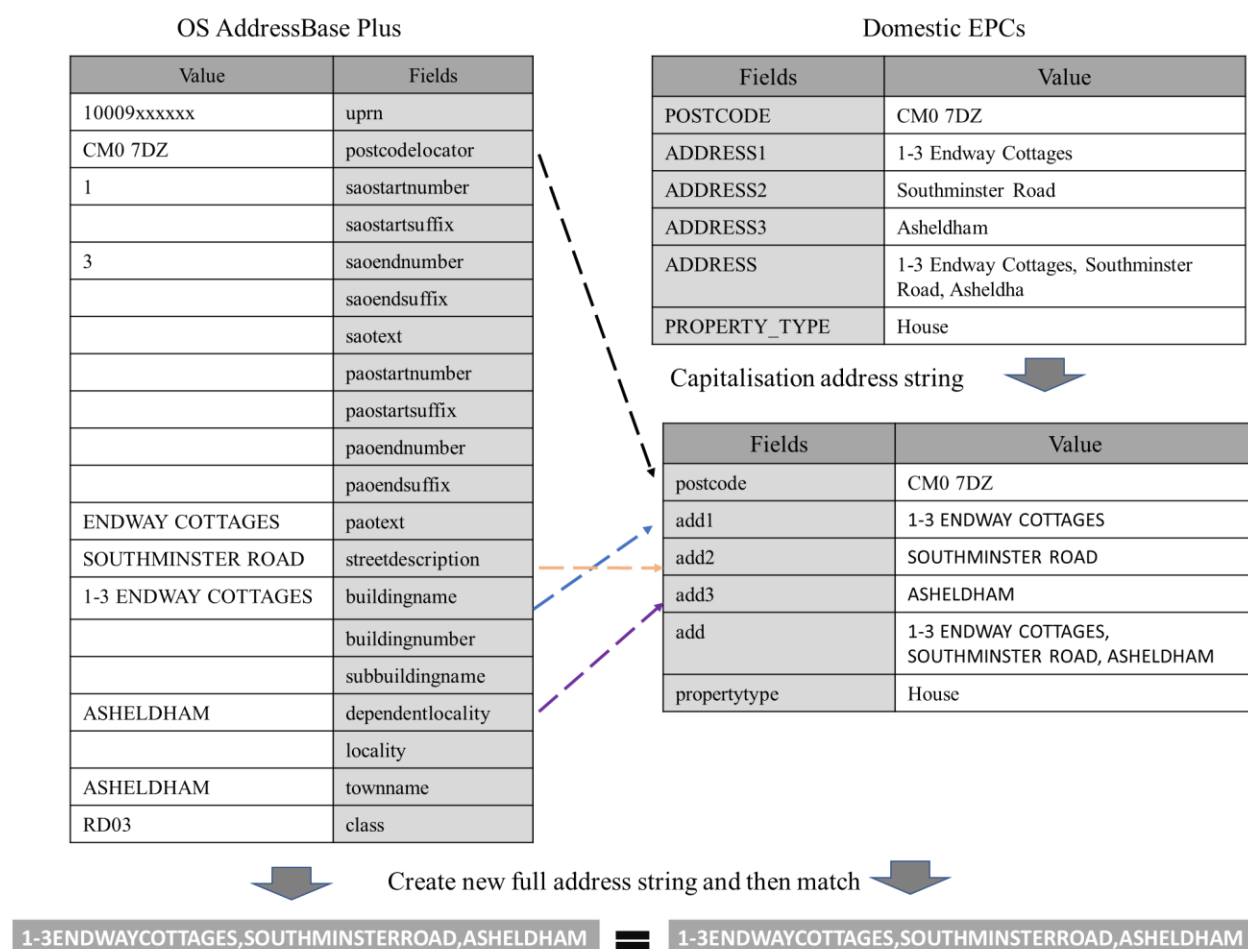


Figure 3 An example of address matching process for Domestic EPCs.

The whole linkage workflow can be seen in Figure B1 (Appendix B). With each matching rule we can match one address with multiple UPRNs. These one-to-many linkage results from each rule based matching process are then cleaned in a customized data cleaning process. There are three approaches in this data cleaning process for EPCs which link to multiple UPRNs. Firstly, when only one of the UPRNs in the multiple linked UPRNs is the residential UPRN, the customized cleaning method keeps this unique residential UPRN. Secondly, additional matching rules are used to get one UPRN result from the multi-linked UPRNs. Thirdly, a combination of the above two approaches.

[Type here]

Only some of the match rules are able to identify a unique residential UPRN when there are one to many possible linkages. Details of the cleaning method is summarised in table B2(Appendix B). The whole linkage process, with 446 match rules, follow a workflow described in Figure B1 to conduct the address matching in order.

3.2.2 Attaching UPRN in LR Price Paid Data (PPD)

Figure 4 shows the address fields in LR PPD and AddressBase Plus data. It is clear that the structure of address fields in LR PPD is totally different than the structure of address fields in EPCs, but more close to the structure in OS AddressBase Plus. This leads to another new rules based matching method is created for address matching.

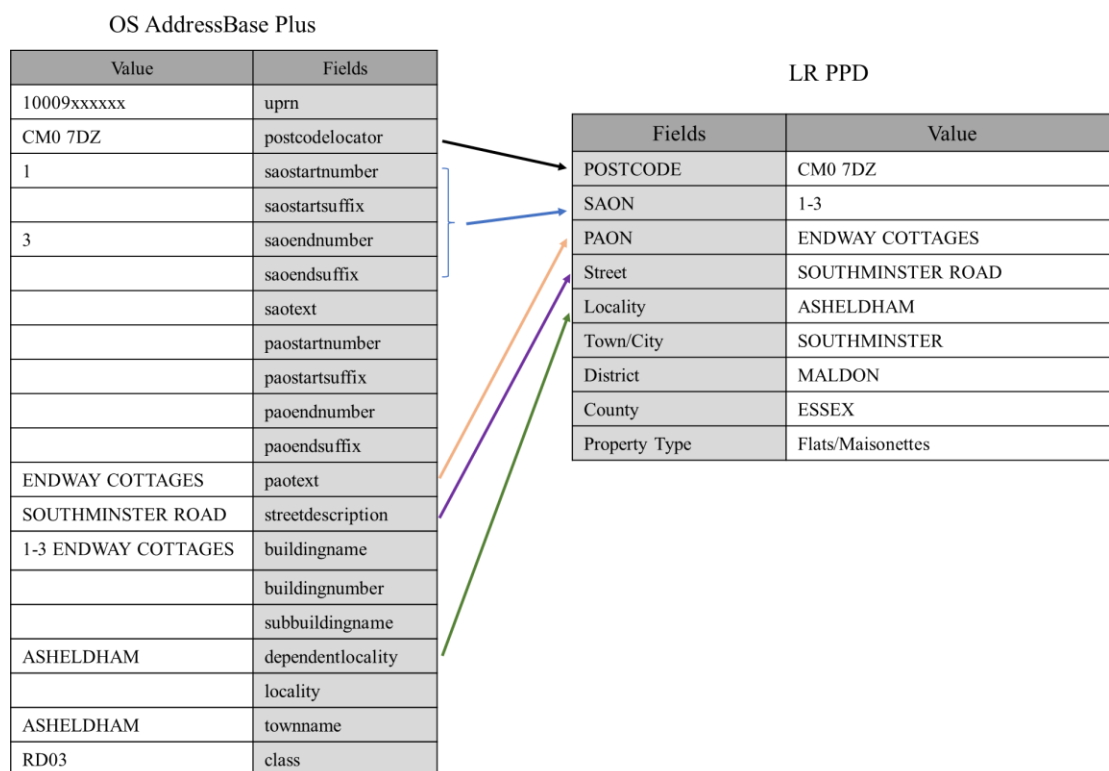


Figure 4 Address fields in LR PPD and AddressBase Plus data

Basic data cleaning and standardization were implemented before the data linkage. These can be viewed in the Table C1(Appendix C), 61 new address fields were created in either LR PPD or AddressBase Plus data. Twenty of them were created in the Land Registry PPD and the rest of were created in the AddressBase Plus data.

Unlike the linkage approach in EPCs, here we conducted the matching in 12 different stages by considering in different conditional situation on PAON (Primary Addressable Object Name) or

[Type here]

SAON(Secondary Addressable Object Name)¹⁰ or both. Within each stage, a series of match rules were conduct in order. As a result, a twelve-stage process with 142 match rules (Figure C1 in Appendix C) was created to attach active UPRN in OS ABP to all PPDs. For each matching result, we only kept the one-to-one linkage result in the linked data. The detail of match rule in each stage are listed in Table C2.

¹⁰ Details of the explanation of both PAON and SAON showed in Figure 1

3. Data linkage

4.1 Data linkage validation

With the two separate rules-based address matching processes, 96.31% of Domestic EPCs and 96.53% of LR PPD transactions have been geo-tagged with one unique UPRN. However, this match rate varies at local scales. To better understand how the match rate differs at the local level, this section separately evaluates the UPRN match rate at local authority for the two datasets.

The UPRN match rate at local authority level is based on the original published Domestic EPCs ranging from 55% to 98% (Figure 5). There are only 19 Local authorities with a lower 85% match rate, which have been labelled on the map. The majority of these are located in the inner London with six local authorities located outside of London. These six outside London areas are Blackpool, Brighton and Hove, Isles of Scilly, Hastings, Scarborough and Ceredigion, and are all coastal towns. Moreover, the majority of North Wales and the South West coast shows a slower 90% UPRN match rate. With our linkage approach, all these low UPRN match rate areas showed an increased UPRN match rate (Figure 6). Unlike the UPRN match rate in original Domestic EPCs (version 9), the overall matching rates by local authority are more equally distributed. The majority of the local authorities are in the highest match rate group (dark green colour). In detail, 282 local authorities in the total 331 local authorities in England and Wales show a higher 95% UPRN match rate. Our linkage work still has 6 local authorities showing a lower 85% UPRN match rate. Except for the Isles of Scilly, the rest of five local authorities are inner London areas. Hammersmith and Fulham in London showed the lowest UPRN match rate (78.61%.) Isles of Scilly, Kensington and Chelsea, Westminster, Camden and Haringey have a UPRN match rate ranging from 82.64 % to 84.48%. Figure 7 summarises the comparison of the UPRN match rate between UBDC rules-based linkage method and DLUHC's method. Our linkage approach is not only able to improve the DLUHC's low match rate, but also maintain the higher 95% match rate.

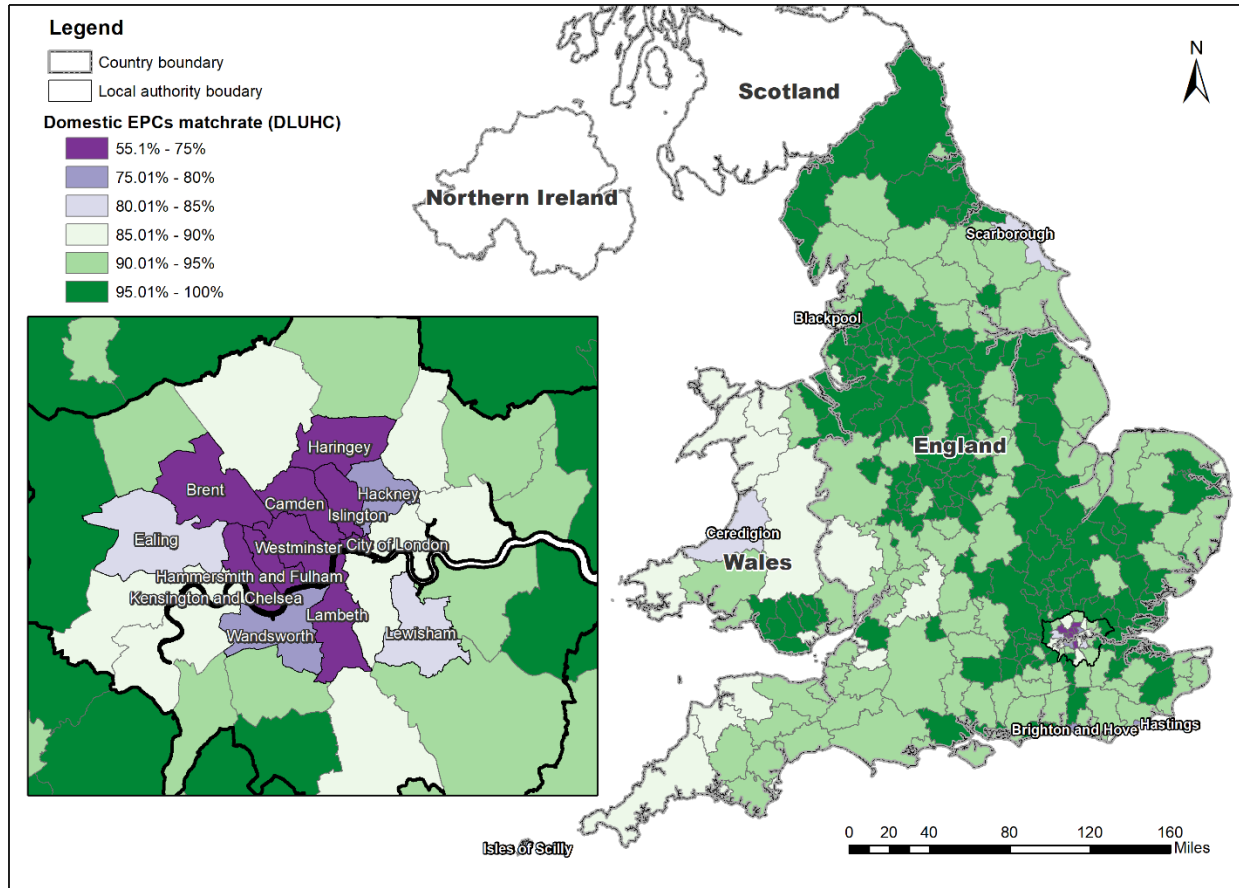


Figure 5 Match rate at LA level by DLUHC's method in England and Wales

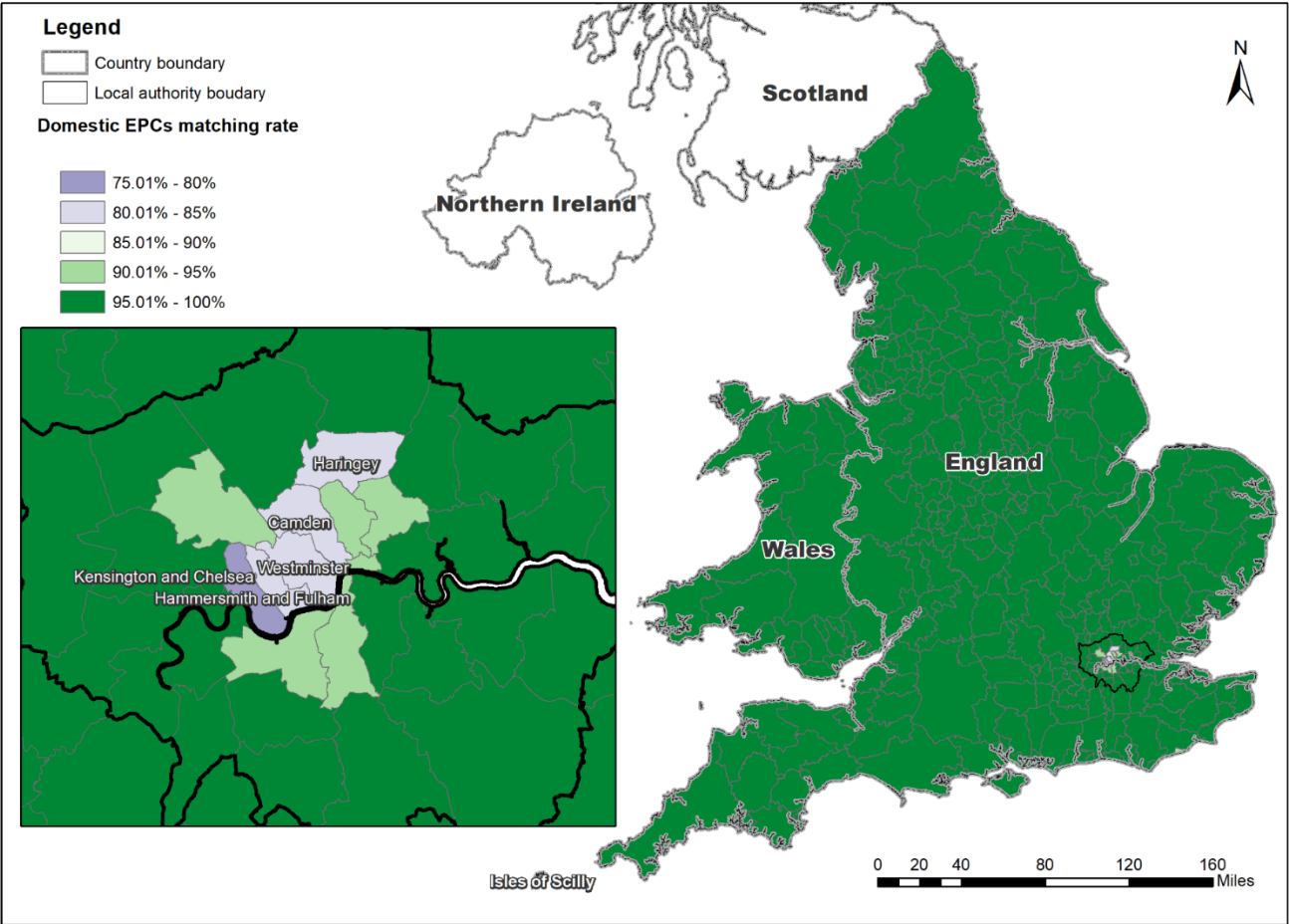


Figure 6 The UPRN match rate at local authority by our method

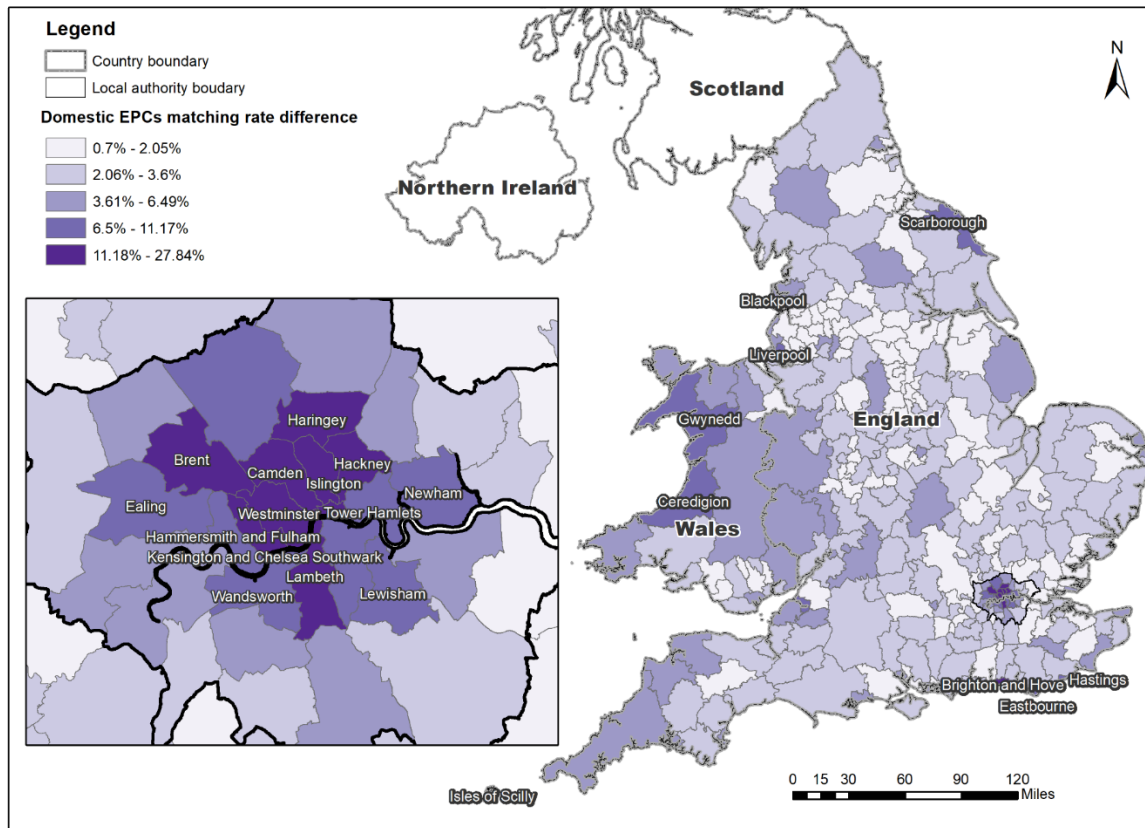


Figure 7 The UPRN match rate different between our method and DLUHC's method

The UPRN match rate for the LR PPD is not equally distributed (Figure 8), but it shows a different pattern than that seen in the equivalent Domestic EPC figure (Figure 6). Of the total 331 local authorities, 282 have an over 95% match rate. Most of the remaining areas are mainly with a match rate between 90% and 95%. The areas with a lower 90% match rate are clustered in inner London, the north part of Wales, and Cornwall and Devon and Lake District. Local authorities with a lower than an 85% UPRN match rate are also labelled in Figure 8. Hammersmith and Fulham, Haringey, Camden, Isles of Scilly, Kensington and Chelsea and Lambeth have a lower than 85% match rate. Among these areas, only Hammersmith and Fulham and Haringey have a lower 80% match rate. All the low match areas are in inner London with the exception of the Isles of Scilly, which is a small group of islands off the UK south west coast.

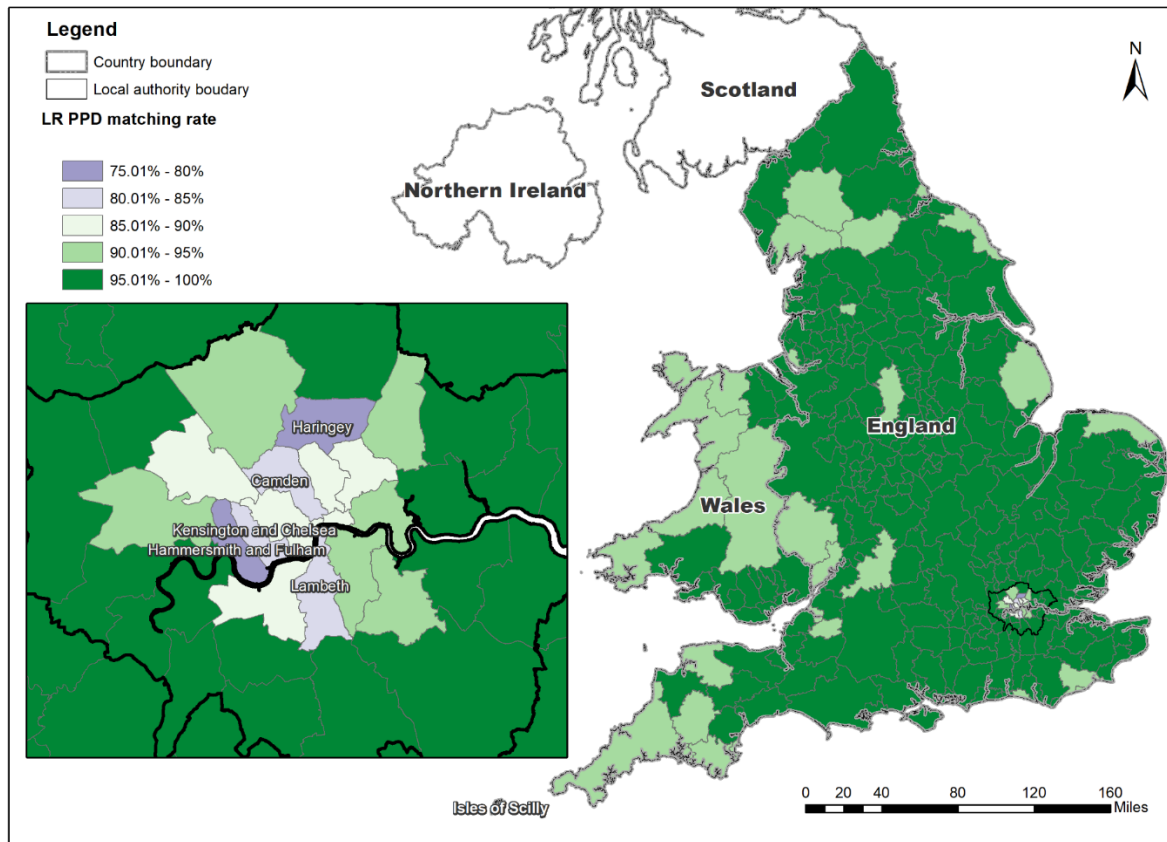


Figure 8 The UPRN match rate for LR PPD at local authority level

4.2 learning lessons from the data linkage

No linkage method provides a perfect match on the first run. Having a good understanding of the linkage quality along knowledge of where and how the linkage failed is useful to support the development of linkage algorithm. We created 446 matching rules for the Domestic EPCs Linkage, while LR PPD only uses 142 matching rules to achieve a similar level of matching rate. EPC linkage was the most time consuming part of this research, compared to that needed for the PPD. Matching rules for each linkage process are ranked by match rate. Figure 9 displays the contribution of the top 20 matching rules to the total matching rate. The address matching linkage for the LR PPD is easier than for the Domestic EPCs, with a higher level of matching achieved for any given number of rules. In detail, using the top 1 linkage rule, 41.45% of the Domestic EPC are linked with a unique UPRN, while this increases to 84.33% for the LR PPD when using its top 1 linkage rule. What is more, LR PPD reaches a 90% match rate using only the first five linkage rules. Domestic EPCs require 28 linkage rules to achieve this milestone. The detail reasons are complex. However, one of the most important reasons for the more efficient linkage with the PPD is that the structure used for recording address information in this data set is more closely aligned to the structure used in OS AddressBase Plus. The second reason is that the address components (e.g. flat name, street name) in the Domestic EPC data set are more frequently recorded differently than the same property in OS AddressBase.

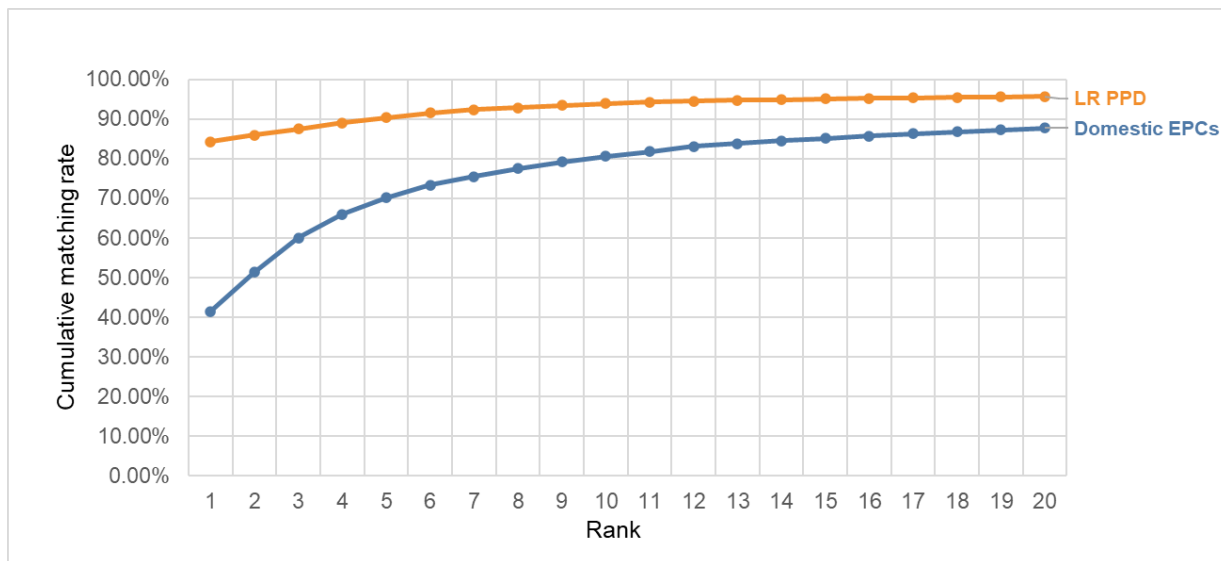


Figure 9 Cumulative match rates by number of matching rules for PPD and Domestic EPC datasets

Hundreds of matching rules are needed to achieve a matching rate greater than 95% for both Domestic EPCs and LR PPD data sets. The majority of these rules deal with three commonly observed differences in the way addresses are stored. These are responsible for many of the issues of matching between address base and EPC/PPD. Firstly, the same address records the same data differently. For example, "101, KNOLE LANE " vs "101 KNOLE LANE, BRENTRY, BRISTOL". This slight difference in recording the same property level requires the creation of more than two match rules to allow the different combinations of the address fields in OS Addressbase. Secondly, the use of different punctuation in the address string compared to that used in the AddressBase products also require a more complex matching process. This is the most significant obstacle for matching these data, which requires a significant number of linkage rules to solve these issues. The difficulty in linking LR PPD and Domestic EPCs with UPRN is largely decided by the use and number of different punctuation marks within different address fields and in the full address field. This is not just the presence of different punctuation marks but also how and where they are used. Compared with Domestic EPC and LR PPD two datasets, OS AddressBase products use fewer punctuation marks and they are rarely used in its address fields. We observed 6 punctuation marks in OS Addressbase Plus: full stop; hyphen; apostrophe; ampersand; forward slash and parentheses. These punctuation marks are often used in some address fields (e.g. buildingname, paotext, streetdescription), but they are used for different reasons. Table 1 contains 10 typical examples in the buildingname field to show how the punctuation marks have been used. Scenarios 1 to 8 are the most common in the building name field, while scenarios 9 and 10 are quite rare. However, directly removing all punctuation marks before the linkage process could lead to incorrect linkage. For example, removing the forward slash as in scenarios 1 and 8 linked with the incorrect address. So when we created the match rules for address matching, we first conduct the matching with punctuation present and then carefully remove the punctuation marks with a series of conditional statements.

The linkage work for EPC dataset is more problematic as the EPCs as it sometimes uses unnecessary punctuation marks. For example, the hyphen and two commas in the address field ["FLAT A- 127 EDEN RD,, CHADDESSEN"]. Different punctuation formats in recording religious

street names also contribute to the difficulty in matching. Such as linking any pair of the following address strings among "ST. MARTINS LANE ", "ST. MARTIN'S LANE " and "SAINT MARTINS LANE". Thirdly, parts of the address strings in the full addresses are different from those in OS AddressBase Plus. This could be the building name, such as "CASPER HOUSE" in EPC vs "CASPAR HOUSE" in OS AddressBase, or street name, such as "78, HARTINGTON ROAD, L8 0SH" in EPC vs "78, FERN GROVE, L8 0SH" in OS AddressBase Plus. There could also be mismatched postcode information, such as "1, BOUNDARY PLACE" are recorded in the EPC data set as GL19 3RE but GL19 3RG in OS AddressBase.

Table 1 Examples of punctuation marks used in OS AddressBase buildingname field

Scenario ID	Field name	Example	Punctuation marks
1	buildingname	1/1	Forward slash
2	buildingname	2-4	Hyphen
3	buildingname	ST. HELENA'S COURT	Full stop and apostrophe
4	buildingname	UNIT 4/12-4/14	Forward slash and hyphen
5	buildingname	UNIT E1/1.74	Forward slash and full stop
6	buildingname	51.02 APARTMENTS	Full stop
7	buildingname	10/6 ST. CATHERINES MANOR	Forward slash and full stop
8	buildingname	3/3 BAILIE FYFE'S CLOSE	Forward slash and apostrophe
9	buildingname	HIDE & SEEK	Ampersand
10	buildingname	(CASTLE)	Parentheses

With the two rule-based linkage methods, 21,051,379 Domestic EPCs and 25,950,983 LR PPD are able to be linked with a unique UPRN. Both of the datasets include property type information which allows us to explore the matching rate by property type. Figure 10 shows the matching rate by the five property types in LR PPD. With houses (Terraced, Semi-detached and Detached), UPRNs are successfully linked in at least 97% of cases but with Flat/Maisonettes, the matching rate falls to 93%. Although Domestic EPCs use different a property type classification system, it also shows a similar rate to that found in the PPD. EPC records another five types of property, which are House, Bungalow, Flat, Maisonette and Park home. The match rate by these five properties are shown in Figure 11. Among the four property types, Flat and Maisonette have a 92% matching rate, while the houses (House and Bungalow) have at a least 97% matching rate.

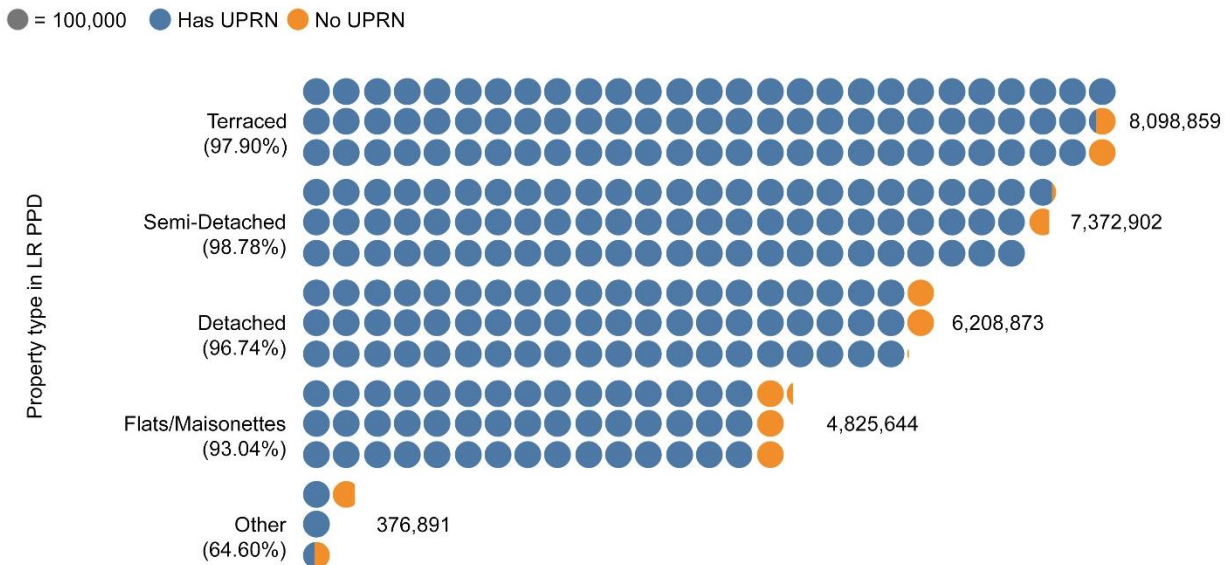


Figure 10 Matching rates by property type in LR PPD

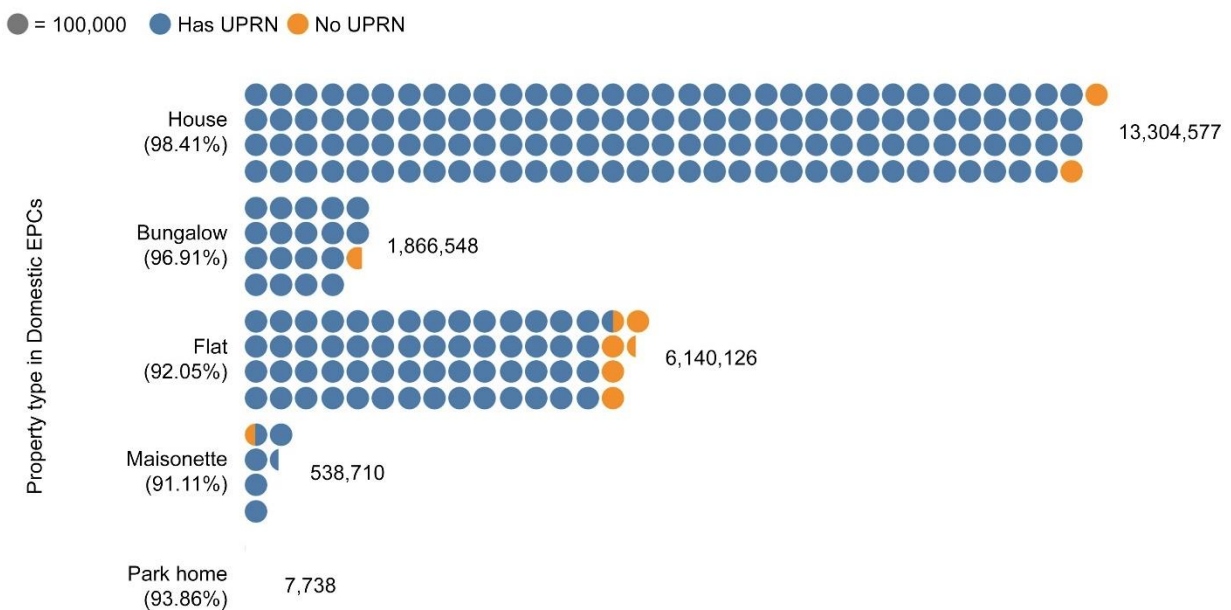


Figure 11 Matching rates by property type in Domestic EPCs

Both data linkages show a lower matching rate for Flat and Maisonette property type. This is mainly caused by the various way in which flat's name or number are record in the three address based datasets. Figure 11 displays 21 types of flat name/number we observed. However, there may be more which we have failed to observe. These different formats in recording flat name or number also leads to a need to create more linkage rules to get a matching. Some of the pair different

[Type here]

formats can get a match in the linkage process. For instance, when two datasets separately use Type 1 and 2 flat name formats for the same address (Figure 12). We can match these by changing “FLAT” string to “APARTMENT” sting in one dataset before conducting the linkage. However, it is impossible to have uniform mathematical rules to match any pair of formats (Figure 12). For example, “FLAT 1” in type 1 is impossible to link when matching the flat name as type 18. This is the one of mains reason that Flat and Maisonette property type showing a lower matching rate in both EPC and PPD.

Type 1	Type 2	Type 3	Type 4	Type 5	Type 6	Type 7	Type 8	Type 9	Type 10
FLAT 1	APARTMENT 1	FLAT 01	STUDIO 1	ROOM 1	UNIT 1	STUDIO FLAT 1	PLOT 1	APT 1	FT 1
FLAT 2	APARTMENT 2	FLAT 02	STUDIO 2	ROOM 2	UNIT 2	STUDIO FLAT 2	PLOT 2	APT 2	FT 2
FLAT 3	APARTMENT 3	FLAT 03	STUDIO 3	ROOM 3	UNIT 3	STUDIO FLAT 3	PLOT 3	APT 3	FT 3
FLAT 4	APARTMENT 4	FLAT 04	STUDIO 4	ROOM 4	UNIT 4	STUDIO FLAT 4	PLOT 4	APT 4	FT 4
Type 11	Type 12	Type 13	Type 4	Type 15	Type 16	Type 17	Type 18		
FLAT A	CLUSTER 1	CLUSTER FLAT 1	CHALET 1	ROOM F 1	LOFT 1	F01	BASEMENT FLAT		
FLAT B	CLUSTER 2	CLUSTER FLAT 2	CHALET 2	ROOM F 2	LOFT 2	F02	GROUND FLOOR FLAT		
FLAT C	CLUSTER 3	CLUSTER FLAT 3	CHALET 3	ROOM F 3	LOFT 3	F03	FIRST FLOOR FLAT		
FLAT D	CLUSTER 4	CLUSTER FLAT 4	CHALET 4	ROOM F 4	LOFT 4	F04	SECOND FLOOR FLAT		
Type 19			Type 20				Type 21		
BASEMENT AND GROUND FLOOR FLAT			MAISONETTE BASEMENT AND GROUND FLOOR				GROUND FLOOR FLAT 1		
FIRST FLOOR FLAT			FLAT FIRST FLOOR				FIRST FLOOR FLAT 2		
SECOND AND THIRD FLOOR			FLAT SECOND FLOOR				TOP FLOOR FLAT		

Figure 12 Types of recording for flat name in the address based data

The reason for the low matching rate for the ‘Other’ category in the PPD is complex, but it allows us to uncover more issues in address matching. This is in part due to the heterogenous nature of the accommodation, which includes things like lodges, barns, garages, parking spaces etc. Five percentage of this category has no postcode information, which requires specific address matching rules. Sometimes, there is no conventions for the way these types of property are referred to. There may also not be an addressable record in OS AddressBase data. One of the main reasons that these types of records are not matched is the level of detail available in the different data sets see Figure 13.

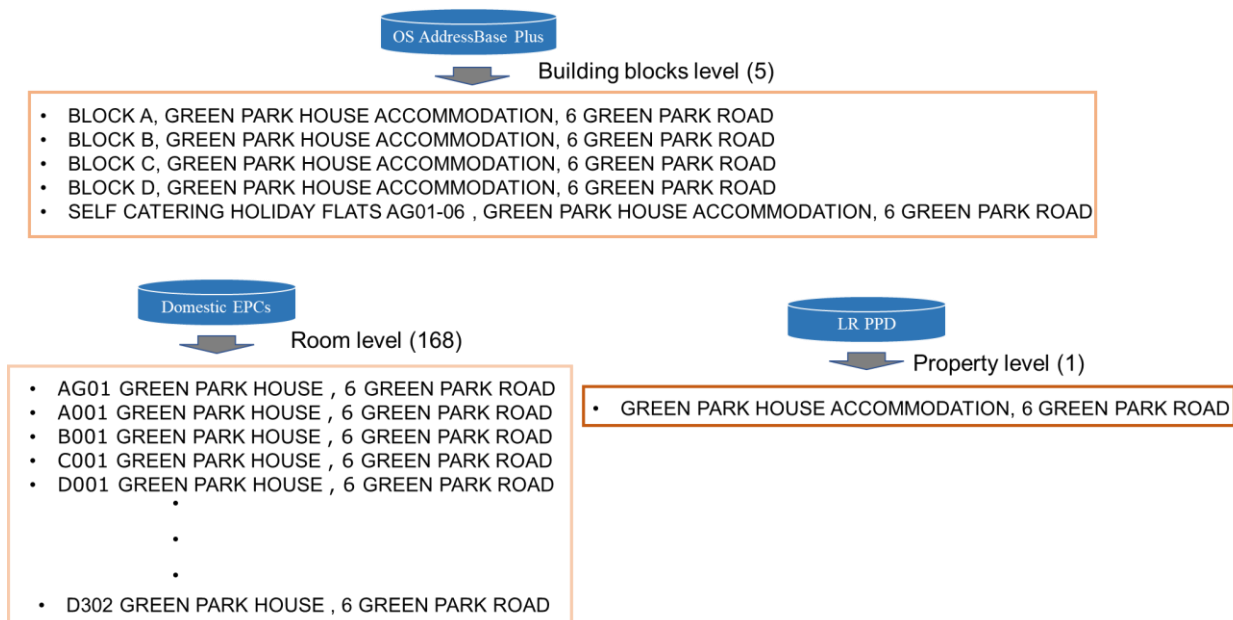


Figure 13 Examples of different recording of the same location in OS AddressBase, Domestic EPCs and LR PPD.

It is common for one address to link to multiple UPRNs during the address matching. This can be due to the incomplete address. For example, “APARTMENT 101, 1 VIADUCT GARDENS” this address is unable to link with one unique UPRN. There are “APARTMENT A101”, “APARTMENT D101” and “APARTMENT F101” all in the same building which can link to the address, but it is impossible to know which one is the correct match. Another example is where the flat’s address has no flat number, such as “FLAT, 355 PORTOBELLO ROAD”.

Another reason for failure can be the inconsistency between address fields in OS AddressBase Data. Figure 14 displays two UPRN examples in the OS AddressBase Plus. The right one has an inconsistency issue between the buildingname and paostartnumber fields. In this case, using buildingname field could cause the “347, THORPE ROAD” record linked with both UPRNs at the same time. What is more, when one of the fields (dependentlocality or locality) has a null value, this could lead to a fake one-to-one linkage result as it could be the one-to-many linkage.

UPRN 1		UPRN 2	
Fields	Value	Value	Fields
postcodelocator	PE3 6LU	PE3 6LU	postcodelocator
paostartnumber	347	349	paostartnumber
paostartsuffix			paostartsuffix
paotext			paotext
streetdescription	THORPE ROAD	THORPE ROAD	streetdescription
buildingname	347	347	buildingname
buildingnumber			buildingnumber
subbuildingname			subbuildingname
dependentlocality			dependentlocality
locality			locality
townname	PETERBOROUGH	PETERBOROUGH	townname
class	RD02	CI03	class

Figure 14 One pair example to show the inconsistency between buidlingname and paostartnumber fields

4.3 learnt lessons from UPRN difference in Domestic EPCs

DLUHC published the ninth version of the Domestic EPC dataset reporting a 92.52% UPRN matching rate. While our method has attained a 96.31% matching rate. Our method has not been successful in matching all the same EPC records which were successfully matched by DLUHCs. The Sankey diagram (Figure 15) below summarises the differences in the UPRN matching between the original DLUHC linkage and our rules-based matching.

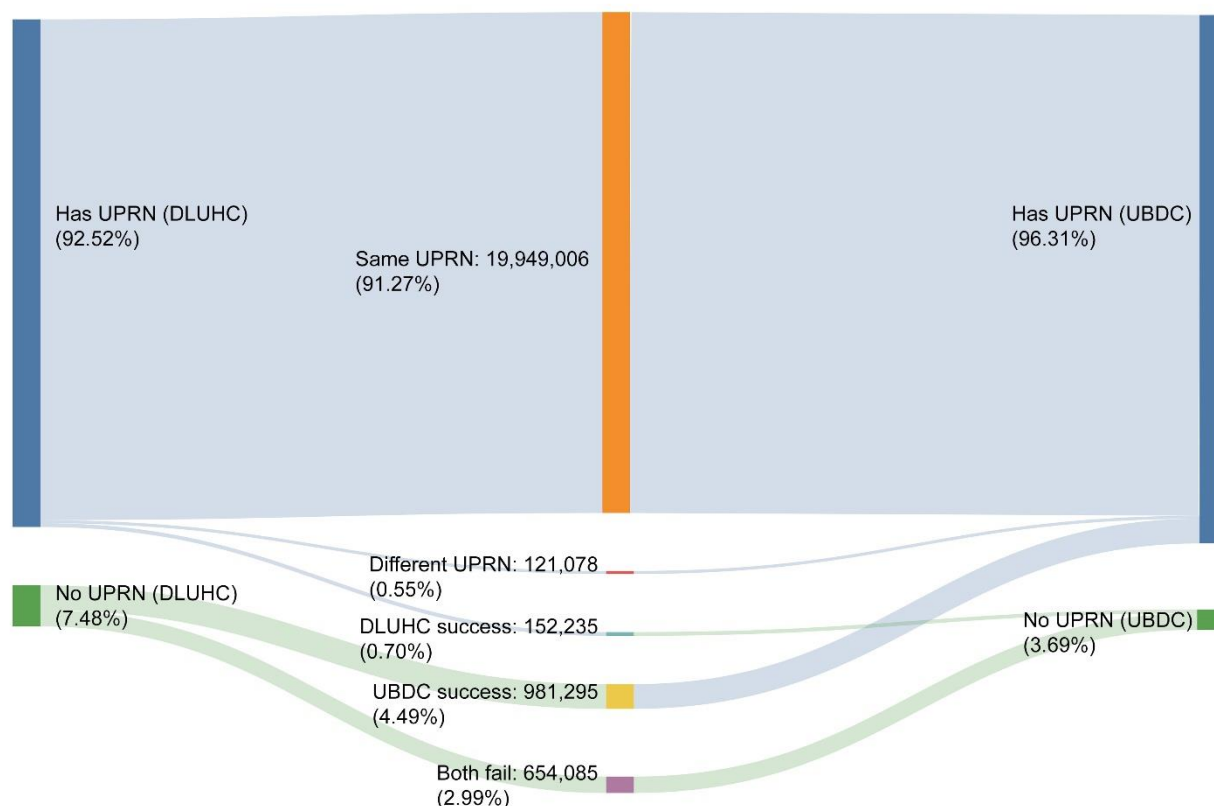


Figure 15 DLUHC and UBDC UPRN linkage comparison

The UPRN assigned by both DLUHC's method and our method (UBDC) shows a great deal of consistency with the same 91% of records being matched by both DLUHC and UBDC (19,949,006 of 21,857,699). Only 0.55% of records linked by DLUHC and UBDC have a different UPRN number. DLUHC have identified 0.7 % of EPC records which UBDC failed to identify, while the UBDC method has attached UPRNs in 4.49% of EPC records for which DLUHC was unable to find a match. This leaves 2.99% of EPCs which both DLUHC and UBDC have failed to match.

Of the 121,078 records (0.55%) which were allocated different UPRNs by the two teams, 91% had a parent-child relationship, with DLUHC assigning the parent UPRN and UBDC assigning the child UPRN. However, in OS AddressBase Plus, sometimes a parent UPRN may only have one child UPRN. The mismatched UPRNs for 0.45% (99,449) of the EPC records are due to DLUHCs choice of the parent UPRN and our choosing the one child UPRN. For example, for "60, QUEEN STREET" address, DLUHC tag the parent level UPRN (100110133800) while our method tags the child UPRN (10023176043). However, it is not always clear which method gives the correct UPRN, especial for an incomplete address like "FLAT, 109 OLDHAM STREET". Table 2 listed four examples to show details on this issue. The difference between how each of the two methods assigns UPRNs is complex and it is difficult to ascertain which linkage, if any, is the correct one. Taking the first three addresses in table 2 as a group example, UBDCs method maybe correct for the address in ID 1, and DLUHC maybe correct for ID 2, and again our method must be right for the ID 3 if considering the property information for this address record.

[Type here]

Table 2 Examples of differently tagged UPRNs with parent and children UPRN relationships

ID	Example address	The UPRN address by DLUHC's method	The UPRN address by our method	Notes
1	FLAT, 154 LEAGRAVE ROAD	154 LEAGRAVE ROAD	FIRST FLOOR, 154 LEAGRAVE ROAD	The property type of this address shows as flat in Domestic EPCs
2	FLAT, 69 GOLBORNE ROAD	69 GOLBORNE ROAD	FIRST AND SECOND FLOOR FLAT, 69 GOLBORNE ROAD	The property type of this address shows as house in Domestic EPCs
3	FLAT, 26 WALCOT BUILDINGS	26 WALCOT BUILDINGS	FLAT, 26 WALCOT BUILDINGS	The property type of this address shows as Maisonette in Domestic EPCs
4	1, ST. ALPHEGE ROAD	1, ST ALPHEGE ROAD	FIRST FLOOR FLAT, 1, ST ALPHEGE ROAD	The other children uprn is "1A, ST ALPHEGE ROAD" or "GROUND FLOOR FLAT, 1, ST ALPHEGE ROAD "

The records where DLUHC have been able to assign a UPRN but our method was unsuccessful represents 0.70% of the whole Domestic EPCs, which is 152,235 records. This group can be further divided into three sub-groups:

- Over half of the 152,235 records (54%) are observed to linked with multiple UPRNs using our rules base method. Our method can allocate multi UPRNs, while DLUHC method on the other hand chooses to link to one of the UPRNs. Most of these records have the building level addresses, but these buildings now are made up of more than one property. Table 3 shows two typical examples of this issue. Tagging the parent UPRN could be one solution for this part, but we might argue that this is not always the right choice.
- Twelve percent of these DLUHC matched records part are tagged with a UPRN only available from OS AddressBase Premium. Almost all these UPRNs have been retired before December 2021. It is not possible to get a successful match with OS AddressBase Plus. There are 217 active UPRNs in OS AddressBase Premium, which are not recorded in AddressBase Plus.
- The remaining 34% is DLUHC successfully linked records. In exploring this part in more depth, we found some interesting results. Here we found both DLUHC's Address Matched

[Type here]

and Energy Assessor could result in a small number of incorrect UPRNs. Tabel 4 listed a couple of this incorrect linked address examples. Moreover,there are 201 mysterious UPRNs achieved by DLUHC that we are unable to find in both OS AddressBase Premium and open UPRN data. They could be the a fake UPRN (i.e. 63074215).

Table 3 Examples of addresses with tagged UPRNs using DLUHC’s method and our method

Case ID	Example address	The parent UPRN address by DLUHC’s method	The multi child UPRNs’ addresses by our method	Note
1	50B ST MICHAELS LANE	50B ST MICHAELS LANE	FLAT 1, 50B ST MICHAELS LANE FLAT 2, 50B ST MICHAELS LANE	The property type for this address is Flat
2	24 CARTERHATCH ROAD	24 CARTERHATCH ROAD	FLAT 1, 24 CARTERHATCH ROAD FLAT 2, 24 CARTERHATCH ROAD FLAT 3, 24 CARTERHATCH ROAD FLAT 4, 24 CARTERHATCH ROAD FLAT 5, 24 CARTERHATCH ROAD FLAT 6, 24 CARTERHATCH ROAD	The property type for this address is House

Table 4 Examples of potentially incorrect linked result carried out by Address Matched and Energy Assessor two approaches

CASE ID	Property Addresses	UPRN’s address	UPRN sources
1	11A JOYDENS WOOD ROAD, BEXLEY	13 JOYDENS WOOD ROAD , BEXLEY	Energy Assessor
2	FLAT 11 BLOCK 2 TURNPIKE WALK, BESWICK	FLAT 1,2A TURNPIKE WALK, MANCHESTER	Energy Assessor
3	ANNEXE, POUND BRIDGE COTTAGE, 15 LITTLE STREET	11, LITTLE STREET	Energy Assessor

[Type here]

4	GROUND FLAT, 2 TEMPLE ROAD	2 TEMPLE ROAD ¹¹	Address Matched
5	UPPER MAISONETTE, 8 WESTCROFT SQUARE, LONDON	8 WESTCROFT SQUARE, LONDON	Address Matched
6	2C, PERTH ROAD	9 , BALMORAL CLOSE ,GOSPORT	Address Matched

The remaining records (654,085 EPC records) failed to be linked by both methods (DULHC and UBDC) and represents 2.99% of the EPC records. One of reasons for the lack of linkages is the addresses are not present in AddressBase Plus. Some of these records are student accommodation. For example, “Flat D15, Europa Student Accommodation, 190 Erskine Street, Liverpool, L6 1AH”.

While exploring the URPNs which have been added to the EPC record by the energy assessor, we found some of these successful cases are not possible to link by current addressing matching approaches. The main reason for this is the task addresses and matched address are quite different. Figure 16 displays some of these cases.

Task address		Linked address
STUDIO TWO, FOREST FRONT, HYTHE	→	BRAMBLES, FOREST FRONT, HYTHE
THE COTTAGE AT, THE BARN, DRY LANE	→	THE BYRE ,THE BARN, DRY LANE
FLAT FIRST FLOOR, 20 BLOOM PARK ROAD	→	TOP FLOOR FLAT, 20 BLOOM PARK ROAD
SUNSET, HIGH STREET, EARLS COLNE	→	CLOTH FAIR, HIGH STREET, EARLS COLNE
SHONA, SUTON STREET, SUTON	→	SUTON CORNER HOUSE , SUTON STREET, SUTON
2 THE GARDENS, FORE STREET	→	APPLE TREE COTTAGE, FORE STREET
2 MALKILN COTTAGE, MALKILN LANE, BISPHAM	→	WHEELWRIGHTS BARN , MALKILN LANE, BISPHAM
LIZARDS FARM, KITSWELL ROAD, LANCHESTER	→	MEADOW VIEW HOUSE, KITSWELL ROAD, LANCHESTER

Figure 16 Matched addresses from energy assessor approach

4. Solutions

Based this research we are suggesting some changes that would lead to improvements in future linkage. We break these down to changes that focus on improving address curation, which will make matching much more straight forward. We also suggest some changes to the approaches to matching in the future based on our observations we have made as a result of the two linkages carried out here.

¹¹ The right address should be “GROUND FLOOR FLAT, 2 TEMPLE ROAD”

[Type here]

5.1. Improving address capture

The low quality of the address information in address based data is the main obstacle to achieving higher rates of linkage more accurate UPRN tagging. It is worthwhile to control the quality of address strings in address based data. Here, three suggestions are offered on how this might be achieved:

- We suggest controlling and standardising the quality of address information in AddressBase datasets. We suggest that agreement is needed on a national standard methodology for collecting and storing address data. Both datasets we examined have examples of address components at the same level (like street name) which are recorded in different fields. To link data that is inconsistent in this way requires more matching rules to be created to overcome the problem.
- It is now possible to use software that provides pull down menus which fill in addresses and these are used on many commercial web sites to ensure the accuracy of delivery information. This type of system avoids users making errors when they are typing in addresses. It would seem perfectly possible to have a similar system which is based on OS' AddressBase products, only requiring manual entry when there is no correct address. This would mean that UPRNs can largely be attached at the point of creating the records. There will inevitably need to be arrangements for manual entry, not least to deal with lags between the creation of UPRNs and data capture systems being updated.
- Data validation and cleaning should be applied regularly to address based data as part of standard curation, ensuring matching is more accurate and complete. This would help deal with punctuation and abbreviations issues. At the very least address validation could avoid or at least flag up unclear property addresses.
- OS should continue to work to improve the AddressBase product. While the quality of addresses stored in the AddressBase products are very high and are the industry standard, there are still some remaining errors or inconsistencies in the data. The main issues are to do with inconsistencies in where some information, such as buildingname or locality names, are stored. There are also questions about missing addresses or properties.

5.2 Improving address matching work

Although it is not possible to have 100% match as address data is dynamic, with properties being demolished, built and converted to flats or properties being merged, there is much that can be learnt from comparing address matching approaches which have been developed within different organisations. A confidence measure for each linkage, an indication of the strength of certainty of the linkage, would also be a useful to inform users of potential limitations in the data.

[Type here]

Different address based data stores addresses in different formats, we suggested creating a linkage method for a given address based dataset. It is impossible to have uniform efficient tools for both linkage work.

Here, we make some suggestions for possible improvements based on our exploration of address matching for the EPC and PPD.

- We believe it would be helpful to have technical guidance from OS or GeoPlace on when to use either the parent or child UPRN during matching, also how to identify retired UPRNs in OS AddressBase Premium products. DLUHC and ourselves have chosen to use different methods for allocating UPRNs when there is a one-to-many UPRN relationship. We have chosen to remove the parent UPRNs when child UPRN are identified and focus only on the active UPRNs. However, extending this work to consider both active and retired UPRN, and parent UPRNs and child UPRNs is more complex, and clear technical guidance on how to handle these two particular issues would help to standardise linking methods and improve accuracy.
- It is worthwhile considering the property type information during the linkage. For example, changing “FLAT” to “Apartment” in Flat/Maisonettes records before conducting a linkage may avoid uncertain linkage results.
- Some addresses have different postcodes in different address datasets, one reason for this is that the addresses on a postcode boundary can have different postcodes in different address databases, so conducting the linkage at postcode unit level and at a higher post level can solve this issue as these addresses can be identified.
- Finally, UPRNs as has been already stated are dynamic and changing and higher match rates can be achieved by ensuring regular updates which include new UPRNs or redundant UPRNs.

Future work

- Improve the data linkage by considering in UPRN’s lifecycle and retired UPRNs in OS AddressBase Premium
- Knowledge exchange workshop with data owners for better UPRN tagging work
- Extend the data linkage work to Scotland

Appendix A

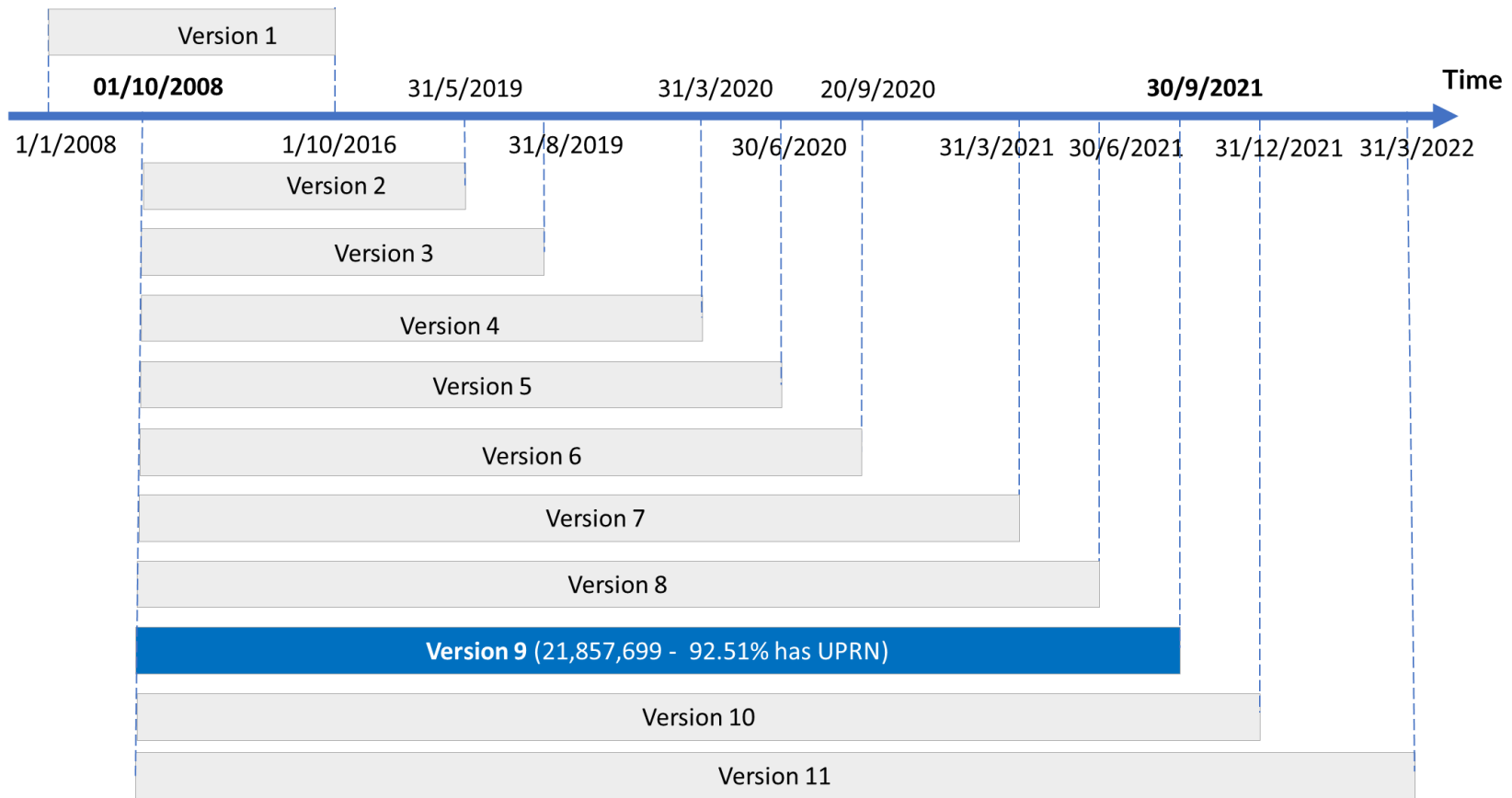


Figure A Historical Domestic EPC versions in England and Wales

[Type here]

Appendix B

Table B1 A summary of the 446 data linkage method for the Domestic EPCs

Method	Spatial Level	Match rule
method1	postcode	buildingnumber,streetdescription[]=add[] ¹²
method2	postcode	buildingnumber,streetdescription[]=add[]
method3	postcode	buildingnumber,streetdescription,townname[]=add[]
method4	postcode	buildingnumber,streetdescription,townname[]=add[]
method5	postcode	buildingnumber,streetdescription[]=add[]
method6	postcode	pp,streetdescription[]=add[]
method7	postcode	buildingnumber,streetdescription,dependentlocality[]=add[]
method8	postcode	buildingname,streetdescription,dependentlocality[]=add[]
method9	postcode	subbuildingname_buildingname,streetdescription[]=add[]
method10	postcode	subbuildingname_buildingname,buildingnumber,streetdescription[]=add[]
method11	postcode	buildingname,streetdescription['.]=add['.]
method12	postcode	buildingname,dependentlocality[]=add[]
method13	postcode	subbuildingname,buildingnumber,streetdescription[]=add[]
method14	postcode	subbuildingname,buildingname,buildingnumber,streetdescription,dependentlocality[]=add[]
method15	postcode	buildingnumber,streetdescription['.]=add['.]
method16	postcode	saotext,paotext,streetdescription[]=add[]
method17	postcode	saotext,buildingname,streetdescription,dependentlocality[]=add[]
method18	postcode	saotext,paostartnumber,streetdescription,dependentlocality[]=add[]
method19	postcode	paotext,streetdescription[]=add[]

¹² buildingnumber,streetdescription[]=add[] is a brief equation to represent the linkage method. The left side of the “=” is the method to create a new address string in OS AddressBase, while the right side of the “=” refers to the address field used in Domestic EPCs. Everything mention in [] is the removed punctuation of the newly created address string. The method 1 here can be interpreted as a three steps linkage method. Firstly, combining buildingnumber and streetdescription two fields with a comma and then removing the blank space as a new address string in OS AddressBase. Secondly, removing the blank space in new address string. Thirdly, conducting the linkage based on the newly created address string at the postcode level.

[Type here]

method20	postcode	paotext,locality,townname[]=add[]
method21	postcode	subbuildingname,buildingnumber_streetdescription,dependentlocality[]=add[]
method22	postcode	subbuildingname,buildingnumber,streetdescription,dependentlocality[]=add[]
method23	postcode	paostartnumber-paoendnumber,streetdescription[]=add[]
method24	postcode	subbuildingname_buildingname,buildingnumber,streetdescription,dependentlocality[]=add[]
method25	postcode	subbuildingname,buildingname,streetdescription[]=add[]
method26	postcode	subbuildingname,buildingname_streetdescription[]=add[]
method27	postcode	subbuildingname_buildingnumber_streetdescription=add
method28	postcode	saotext_buildingnumber_streetdescription=add
method29	postcode	saotext_pp_streetdescription=add
method30	postcode	buildingname=add
method31	postcode	ss_paotext=add
method32	postcode	pp_streetdescription=add
method33	postcode	subbuildingname_buildingname[]=add[]
method34	postcode	buildingnumber,paotext=add
method35	postcode	ss,paotext[]=add[]
method36	postcode	buildingname =add[,]
method37	postcode	buildingname [,-./']=add [,-./']
method38	postcode	ss_paotext [,-./']=add [,-./']
method39	postcode	saotext_poatext,paostartnumber-paoendnumber,streetdescription[]=add[]
method40	postcode	buildingnumber,paotext[.] =add['.]
method41	postcode	ss_paotext=add[,]
method42	postcode	subbuildingname_paotext[]=add[]
method43	postcode	saotext_paotext[]=add[]
method44	postcode	buildingname(first word)_streetdescription[]=add[]

[Type here]

method45	postcode	pp_streetdescription[]=add[]
method46	postcode	ss_streetdescription[]=add[]
method47	postcode	buildingnumber_streetdescription[]=add[]
method48	postcode	subbuildingname,buildingname_buildingnumber_streetdescription[]=add[]
method49	postcode	saotext,buildingname_buildingnumber_streetdescription[]=add[]
method50	postcode	subbuildingname,buildingname_pp_streetdescription[]=add[]
method51	postcode	buildingnumber[]=add(string before the first comma)[] only for SY16 1Q
method52	postcode	subbuildingname,paotext_buildingnumber_streetdescription[']=add[']
method53	postcode	saotext,paotext_buildingnumber_streetdescription[']=add[']
method54	postcode	saotext,paotext_pp_streetdescription['.]=add['.]
method55	postcode	subbuildingname,paotext_buildingnumber,_streetdescription['.]=add['.]
method56	postcode	subbuildingname,paotext_pp_streetdescription['.]=add['.]
method57	postcode	buildingnumber,paotext['.]=add['.]
method58	postcode	saotext,pp_buildingnumber_streetdescription[']=add[']
method59	postcode	subbuildingname_buildingname=add
method60	postcode	paotext_buildingname=add
method61	postcode	saotext_buildingname=add
method62	postcode	subbuildingname,buildingname_buildingnumber,streetdescription[]=add[]
method63	postcode	saotext,buildingname_buildingnumber,streetdescription[]=add[]
method64	postcode	subbuildingname,paotext_buildingnumber,streetdescription[]=add[]
method65	postcode	subbuildingname,paotext_pp,streetdescription[]=add[]
method66	postcode	saotext,paotext_pp,streetdescription[]=add[]
method67	postcode	buildingnumber_paotext[]=add[]
method68	postcode	buildingname_paotext[]=add[]
method69	postcode	ss,streetdescription[]=add[]

[Type here]

method70	postcode	subbuildingname,buildingname,buildingnumber,streetdescription[']=add[']
method71	postcode	saotext_paotext,buildingname,streetdescription[']=add[]
method72	postcode	subbuildingname_paotext,buildingname,streetdescription[']=add[]
method73	postcode	buildingname,paotext,streetdescription[]=add[]
method74	postcode	ss,paotext,streetdescription[']=add[']
method75	postcode	buildingnumber_streetdescription,dependentlocality[']=add[']
method76	postcode	buildingnumber,paotext,streetdescription[']=add[']
method77	postcode	saotext,paotext,streetdescription[']=add [.']
method78	postcode	buildingnumber_streetdescription,townname[]=add[]
method79	postcode	pp_streetdescription,townname[]=add[]
method80	postcode	saotext,pp_buildingnumber_streetdescription [.']=add[, ']
method81	postcode	buildingnumber,streetdescription,dependentlocality[, ']=add[, ']
method82	postcode	pp,streetdescription,dependentlocality[, ']=add[, ']
method83	postcode	saotext,paostartnumber,streetdescription[]=add[]
method84	postcode	paostartnumber,streetdescription[']=add[']
method85	postcode	subbuildingname_buildingname,streetdescription,dependentlocality[']=add[']
method86	postcode	saotext_buildingname,streetdescription,dependentlocality[']=add[']
method87	postcode	subbuildingname_paotext,streetdescription,dependentlocality[']=add[.']
method88	postcode	subbuildingname_buildingname[']=add[']
method89	postcode	saotext,pp,streetdescription['/'].]=add['./]
method90	postcode	saotext,pp_streetdescription['/'].]=add['./]
method91	postcode	saotext,buildingname,buildingnumber,streetdescription[']=add[']
method92	postcode	subbuildingname,buildingname,pp,streetdescription[']=add[']
method93	postcode	subbuildingname,buildingname,buildingnumber_streetdescription[]=add[-]
method94	postcode	subbuildingname,buildingname,pp_streetdescription[']=add['.-]

[Type here]

method95	postcode	saotext,buildingname,pp_streetdescription['.']=add['.-']
method96	postcode	pp,townname[]=add[.]
method97	postcode	pp,dependentlocality['.']=add['.']
method98	postcode	pp,locality['.']=add['.']
method99	postcode	saotext,buildingnumber,streetdescription['.']=add['.']
method100	postcode	saotext,pp,streetdescription['.']=add['.']
method101	postcode	subbuildingname,pp,streetdescription['.']=add['.']
method102	postcode	subbuildingname,buildingname,streetdescription['.']=add['.']
method103	postcode	buildingnumber,paotext,locality['.']=add['.']
method104	postcode	buildingname_streetdescription,locality['.']=add['.']
method105	postcode	subbuildingname,buildingnumber,streetdescription['.']=add['./']
method106	postcode	buildingnumber,paotext_locality['.']=add['./']
method107	postcode	pp,streetdescription['.']=add[.]
method108	postcode	pp_streetdescription,townname['.']=add['./']
method109	postcode	buildingnumber,paotext,dependentlocality['.']=add['.']
method110	postcode	ss,paotext,dependentlocality['.']=add['.']
method111	postcode	subbuildingname,buildingname_paotext,buildingnumberstreetdescription['./']=add['./']
method112	postcode	ss_paotext,dependentlocality['.']=add['.']
method113	postcode	buildingnumber_paotext,dependentlocality['.']=add['.']
method114	postcode	buildingnumber_buildingname,dependentlocality['.']=add['.']
method115	postcode	ss_paotext,streetdescription['.']=add['.']
method116	postcode	ss_buildingname,streetdescription['.']=add['.']
method117	postcode	saotext,paostartnumber-paoendnumber_streetdescription[]=add['.']
method118	postcode	subbuildingname,paostartnumber-paoendnumber_streetdescription[]=add['.']
method119	postcode	buildingname,streetdescription['.']=add['.']

[Type here]

method120	postcode	subbuildingname,buildingnumber_streetdescription['.']=add['./']
method121	postcode	subbuildingname,pp_streetdescription['.']=add['./']
method122	postcode	saotext_paotext,paostartnumber-paoendnumber_streetdescription['.']=add['.']
method123	postcode	subbuildingname_paotext,paostartnumber-paoendnumber,streetdescription['.']=add['.']
method124	postcode	ss_paotext,paostartnumber_streetdescription['.']=add['./']
method125	postcode	saotext_paotext,streetdescription['.']=add['./']
method126	postcode	subbuildingname_paotext,streetdescription['.']=add['./']
method127	postcode	subbuildingname_buildingname,streetdescription['.']=add['./']
method128	postcode	saotext_buildingname,streetdescription['.']=add['./']
method129	postcode	saotext,ss_paotext,streetdescription['.']=add['./']
method130	postcode	saotext,ss_buildingnumber,streetdescription['.']=add['./']
method131	postcode	saotext_ss,paotext,streetdescription['.']=add['.']
method132	postcode	paotext_ss,streetdescription[]=add[]
method133	postcode	buildingnumber_paotext['.']=add['.']
method134	postcode	subbuildingname,buildingname_streetdescription['./']=add['./']
method135	postcode	saotext,paotext,pp_streetdescription['.']=add1,add2,add3['.']
method136	postcode	subbuildingname,buildingname_streetdescription['./']=add1,add2['./']
method137	postcode	saotext,buildingname,streetdescription['./']=add1,add2['./']
method138	postcode	saotext,pp,streetdescription['.']=add1,add2['./']
method139	postcode	subbuildingname,buildingname,ss_paotext,streetdescription['./']=add['./']
method140	postcode	subbuildingname,buildingname,streetdescription['./']=add1,add3['./']
method141	postcode	pp,streetdescription['.']=add['.']
method142	postcode	buildingnumber,streetdescription['.']=add1['.']
method143	postcode	buildingname['.']=add1['.']
method144	postcode	buildingname[-]=add1[-]
method145	postcode	buildingname_buildingnumber_streetdescription['./']=add1_add2['./']

[Type here]

method146	postcode	buildingname_pp_streetdescription[./]=add1_add2[./]
method147	postcode	saotext,buildingnumber_pp_streetdescription[./]=add1,add2[./]
method148	postcode	saotext,buildingname_streetdescription[./]=add1,add2[./]
method149	postcode	subbuildingname,pp_streetdescription[./]=add1,add2[./]
method150	postcode	saotext,pp_streetdescription[./]=add1,add2[./]
method151	postcode	subbuildingname,buildingname_paotext[./]=add1,add2[./]
method152	postcode	saotext,paostartnumber-paoendnumber,paotext[./]=add1,add2[./]
method153	postcode	subbuildingname,buildingnumber_streetdescription[./]=add1,add3[./]
method154	postcode	saotext,paotext,paostartnumber-paoendnumber_streetdescription[./]=add1,add2[./]
method155	postcode	saotext,paotext,ss_streetdescription[./]=add[./]
method156	postcode	saotext,paotext,ss_streetdescription[./]=add[./]
method157	postcode	saotext,paotext,paostartnumber-paoendnumber_streetdescription[.] =add[.]
method158	postcode	saotext,paotext,paostartnumber-paoendnumber_streetdescription[./] =add[./]
method159	postcode	saotext,paotext,pp_streetdescription[./]=add[./]
method160	postcode	subbuildingname,paotext,pp_streetdescription[./]=add[./]
method161	postcode	saotext,paotext[./]=add1,add2[./]
method162	postcode	saotext,paotext[./]=add1,add2[./]
method163	postcode	subbuildingname,buildingnumber_paotext[./]=add1,add2[./]
method164	postcode	saotext,paotext,streetdescription[./]=add1,add2[./]
method165	postcode	saotext_paotext,paostartnumber-paoendnumber,streetdescription[./]=add1,add2[./]
method166	postcode	subbuildingname_paotext,paostartnumber-paoendnumber,streetdescription[./]=add1,add2[./]
method167	postcode	subbuildingname_buildingname,streetdescription[,]=add1,add2[,]
method168	postcode	subbuildingname_buildingname,streetdescription[./,]=add1,add2[./,]
method169	postcode	paotext[./]=add1[./,]

[Type here]

method170	postcode	subbuildingname,buildingname,streetdescription[./]=add1[./]
method171	postcode	saotext,pp[]=add1,add2(before -)[.]
method172	postcode	saotext,pp[./]=add1,add2(before -)[./]
method173	postcode	buildingnumber_streetdescription[./]=add1[./]
method174	postcode	buildingnumber,streetdescription,locality=add1(before first blank space)_add2,add3[.]
method175	postcode	ss,streetdescription,locality[.]=add1(before first blank space)_add2,add3[.]
method176	postcode	ss_paotext[./]=add1[./]
method177	postcode	subbuildingname,buildingname[./]=add1,add2(string before comma)[./]
method178	postcode	buildingnumber_streetdescription[.-]=add[.-]
method179	postcode	subbuildingname,pp_paotext[./]=add1,add2[./]
method180	postcode	saotext_paotext,streetdescription[./]=add1,add3[./]
method181	postcode	subbuildingname,buildingnumber,streetdescription([-])[./]=add[./]
method182	postcode	subbuildingname_buildingname,buildingnumber_streetdescription[./]=add1_add2,add3[./]
method183	postcode	subbuildingname,buildingnumber,paotext[./]=add1,add2[./]
method184	postcode	saotext,buildingnumber,paotext[./]=add1,add2[./]
method185	postcode	subbuildingname,ss,paotext[./]=add1,add2[./]
method186	postcode	saotext,ss,paotext[./]=add1,add2[./]
method187	postcode	saotext,ss_paotext,pp_streetdescription[./]=add[./]
method188	postcode	saotext,pp_paotext[./]=add1,add2[./]
method189	postcode	saotext_pp_paotext[./]=add1_add2[./]
method190	postcode	buildingnumber,dependentlocality[.] =add[.]
method191	postcode	subbuildingname,buildingnumber_streetdescription[./]=add1,add2[./]]

[Type here]

method192	postcode	saotext,saostartnumber-saoendnumber_paotext,streetdescription[./]=add[./]
method193	postcode	saotext,saostartnumber-saoendnumber_paotext,streetdescription[./]=add1,add2[./]
method194	postcode	saotext,saostartnumber-saoendnumber_paotext[./]=add1,add2[./]
method195	postcode	saotext,buildingname_paotext[./]=add1,add2[./]
method196	postcode	subbuildingname,buildingnumber_streetdescription[]="FLAT "add1(before first blank space),add2[.]
method197	postcode	subbuildingname,buildingnumber_streetdescription[]="FLAT "add1(before first blank space),add2[.]
method198	postcode	"FLAT "_saostartnumber_saostartsuffix,paostartnumber-paoendnumber_streetdescription[]=add1,add2[]
method199	postcode	"FLAT "_saostartnumber_saostartsuffix,paostartnumber-paoendnumber_streetdescription['.']=add1,add2['.]
method200	postcode	"FLAT " to "APARTMENT" in subbuildingname_buildingname,streetdescription['.']=add1_add2,add3['.]
method201	postcode	"APARTMENT"_saotext,streetdescription['.']=add1_add2,add3['.]
method202	postcode	"APARTMENT"_saotext(string before first blank space),streetdescription['.']=add1,add3['.]
method203	postcode	subbuildingname,paostartnumber_streetdescription[./]=add1(apartment to flat),add3[./]
method204	postcode	subbuildingname_buildingname['.']=add1_add2(string before first comma)[/]
method205	postcode	paostartnumber_"FLAT "_saotext_streetdescription['.']=add['.]
method206	postcode	saotext,_pp_streetdescription="ROOM "_add1,_add3
method207	postcode	subbuildingname_buildingname,streetdescription[./]= "APARTMENT "add[/.']
method208	postcode	subbuildingname_buildingname,streetdescription[.]="APARTMENT "add1,add2[.]
method209	postcode	saotext,paotext,paostartnumber_paostartsuffix_streetdescription['.']= "APARTMENT "add['.]

[Type here]

method210	postcode	subbuildingname,buildingname,streetdescription['.']="FLAT "change to "APARTMENT ")add['.]
method211	postcode	saotext_paotext,streetdescription['.']="FLAT "add['.]
method212	postcode	saotext,pp,streetdescription[]=("FLAT" to "APARTMENT") add[]
method213	postcode	saotext,pp_streetdescription[]=ROOM" add[]
method214	postcode	saotext,pp_streetdescription[/.'.']="APARTMENT "add1(string before first blank space),add2['./]
method215	postcode	saotext_paotext,streetdescription['.']="APARTMENT "add1,add3['.]
method216	postcode	subbuildingname,buildingnumber_streetdescription['.]=("APARTMENT " to "FLAT")add[/.'.]
method217	postcode	"FLAT "buildingnumber,streetdescription[/.'.]=add[/.'.]
method218	postcode	subbuildingname,streetdescription[/.'.]="FLAT "_add[/.'.]
method219	postcode	saotext,streetdescription[/.'.]="FLAT "_add[/.'.]
method220	postcode	subbuildingname,buildingnumber_streetdescription[/.'.]="APARTMENT "_add[/.'.]
method221	postcode	subbuildingname,streetdescription[]=add[]
method222	postcode	ss,paotext,paostartnumber-paoendnumber_streetdescription[/.'.]=add[/.'.] (remove the FLAT)
method223	postcode	saotext,paotext[/.'.]="APARTMENT "_add[/.'.]
method224	postcode	pp,streetdescription[/.'.]=add[/.'.](remove the FLAT)
method225	postcode	"APARTMENT "_ss,paotext,pp_streetdescription['.]=add['.]
method226	postcode	subbuildingname_buildingname_streetdescription[,]="FLAT "_add[,]
method227	postcode	"FLAT "_subbuildingname,paotext,buildingnumber_streetdescription[,]=add[,]
method228	postcode	buildingname_saotext(remove the "APARTMENT ")[pu the last word string in the first]=add(before first comma)
method229	postcode	"FLAT "ss_pp_streetdescription=add
method230	postcode	subbuildingname(change "FLAT" to "APT")_buildingname_buildingnumber_streetdescription=add

[Type here]

method231	postcode	subbuildingname_buildingname_buildingnumber_streetdescription['.]=add[,'.]
method232	postcode	buildingname_streetdescription=add
method233	postcode	buildingnumber,locality['.]=add['.]
method234	postcode	paotext,townname['./,]=add['./,]
method235	postcode	buildingname,locality['.]=add['.]
method236	postcode	saotext_buildingname_streetdescription,dependentlocality['./]=add1,add2['./]
method237	postcode	saotext_buildingnumber_streetdescription,locality['./]=add1,add2['./]
method238	postcode	buildingnumber,streetdescription[-]=add[-]
method239	postcode	saotext_pp_streetdescription,locality['./]=add1,add2['./]
method240	postcode	pp_streetdescription[-.]=add[-.'],
method241	postcode	paotext []="FLAT "_add[,]
method242	postcode	subbuildingname_paotext[]="FLAT "_add[]
method243	postcode	"FLAT "_subbuildingname,paotext,buildingnumber_streetdescription[]=add[]
method244	postcode	saotext(change "FLAT" to "APARTMENT"),paotext,pp_streetdescription['.]=add['.]
method245	postcode	saotext(change "FLAT" to "APARTMENT")_paotext,pp_streetdescription[]=add[]
method246	postcode	saotext_paotext[]="FLAT "_add1[]
method247	postcode	"FLAT "_saotext[]= add(frist word)[,]
method248	postcode	saotext,pp_streetdescription[remove "FLAT"/. ']=add['. ']
method249	postcode	saotext_pp_streetdescription,dependentlocality['. ']=add1,add2['. ']
method250	postcode	saotext,paotext['. ']=add1[,],add2['. ']
method251	postcode	subbuildingname_paotext,buildingnumber,streetdescription['. ']=add['. ']
method252	postcode	subbuildingname_buildingname_streetdescription,locality['. ']=add1,add2['. ']

[Type here]

method253	postcode	saotext,pp_streetdescription[/.']=add1,add3[/.']
method254	postcode	pp_streetdescription,locality[/.']=add1[/.']
method255	postcode	buildingname_streetdescription[/.']=add1[/.']
method256	postcode	buildingnumber_streetdescription[/.']=add1[/.']
method257	postcode	saotext_pp_streetdescription[/.']=add1[/.']
method258	postcode	subbuildingname_paotext_buildingnumber,streetdescription[/.']=add[/.']
method259	postcode	buildingnumber,streetdescription[.']=add[.']
method260	postcode	ss,streetdescription[]=add[]
method261	postcode	buildingnumber,paotext[]=add[]
method262	postcode	buildingnumber,streetdescription[-]=add[-]
method263	postcode	buildingnumber_streetdescription[]=add1[,]
method264	postcode	pp_streetdescription[]=add1[]
method265	postcode	buildingname_streetdescription[]=add1[,]
method266	postcode	subbuildingname_buildingnumber_streetdescription[/.']=add(change "APARTMENT" to "FLAT")[/.']
method267	postcode	subbuildingname_paotext,streetdescription[]="APARTMENT "_add[]
method268	postcode	subbuildingname_buildingname_buildingnumber_streetdescription="A PARTMENT "_add
method269	postcode	subbuildingname_buildingname,buildingnumber_streetdescription[/.']="APARTMENT "_add[/.']
method270	postcode	ss_paotext[]=add1[]
method271	postcode	saotext_paotext[/.']=add1[/.' ,]
method272	postcode	subbuildingname_paotext[/.']=add1[/.' ,]
method273	postcode	saotext_buildingname[/.']=add1[/.' ,]
method274	postcode	subbuildingname_buildingname[/.']=add1[/.' ,]
method275	postcode	buildingnumber,paotext,streetdescription[-.]=add[-.]

[Type here]

method276	postcode	saotext,paotext,streetdescription[/.']=add[/.']
method277	postcode	saotext_paotext,streetdescription[]=add[]
method278	postcode	saotext,pp_streetdescription[/.']="APARTMENT "add[/.']
method279	postcode	subbuildingname,buildingname,streetdescription[/.']=add(change "APT" to "FLAT")[/.']
method280	postcode	subbuildingname,buildingname_buildingnumber_streetdescription[/.']=add[/.']
method281	postcode	saotext[]=add1(first word)[,]
method282	postcode	ss,streetdescription[/.']=add[/.']
method283	postcode	paostartnumber_saotext,streetdescription[]=add[]
method284	postcode	paostartnumber_saotext(remove "FLAT")_streetdescription=add
method285	postcode	pp_saotext_streetdescription[/.']=add[/.']
method286	postcode	saotext(change "STUDIO" to "FLAT")[/.']=add1(first word)[/.' ,]
method287	postcode	saotext_paotext(change "APARTMENT" to 0)[/.']=add1[/.' ,]
method288	postcode	pp[]=add1(frist word)[,]
method289	postcode	subbuildingname[]=add1(frist word)[,]
method290	postcode	buildingnumber[]=add1(frist word)[,]
method291	postcode	saostartnumber_saostartsuffix[/.']=add1(frist word)[/.' /]
method292	postcode	"APARTMENT "subbuildingname['./]=add1(first word)['./ /]
method293	postcode	"APARTMENT "ss[]=add1(first word)[,]
method294	postcode	saotext(remove "CHALET ")[]=add1(first word)[,]
method295	postcode	"FLAT "_ss_paotext[]=add1_add2['.]
method296	postcode	"FLAT "_ss_paotext[]=add1_add2['. ,]
method297	postcode	"FLAT "_ss_pp_streetdescription=add1_add2
method298	postcode	"FLAT "_ss_paostartnumber-paoendnumber_streetdescription[/.']=add1_add2(/.' ,)
method299	postcode	"FLAT "_ss_pp_streetdescription[/.']=add1_add3[/.' ,]

[Type here]

method300	postcode	saotext,paotext,pp_streetdescription[/.']="FLAT "_add[/.']
method301	postcode	"STUDIO "_saotext,pp_streetdescription[/.']=add1,add3[/.']
method302	postcode	saotext,paotext,paostartnumber- paoendnumber_streetdescription[/.']="APARTMENT "add1[-],add2[/.']
method303	postcode	"APARTMENT "ss,paotext,streetdescription,townname[/.']=add['./]
method304	postcode	saotext_buildingname,buildingnumber_streetdescription[-]=add1[/.'],a dd2,add3[/.']
method305	postcode	"FLAT "_buildingname_streetdescription[/.']=add[/.' ,]
method306	postcode	"STUDIO "subbuildingname_paotext_buildingnumber_streetdescription[/.']=ad d[-/']
method307	postcode	subbuildingname_paotext,buildingnumber,streetdescription[/.']=add[- /']
method308	postcode	subbuildingname,buildingname,streetdescription[/.']=add ("STUDIO" to "FLAT")[/.']
method309	postcode	subbuildingname,buildingname_streetdescription[/.']=add("STUDIO" to "FLAT")[/.']
method310	postcode	subbuildingname,buildingnumber_streetdescription[/.']=add("STUDIO" to "FLAT")[/.']
method311	postcode	subbuildingname,pp_streetdescription[/.']=add("STUDIO" to "FLAT")[/.']
method312	postcode	saotext,paotext,pp_streetdescription[/.' ,]=add("FLAT" to "APARTMENT")[/.' ,]
method313	postcode	subbuildingname_buildingnumber,streetdescription[/.' ,]=add1,add2 ("FLAT" to "APARTMENT")[/.' ,]
method314	postcode	subbuildingname_paotext,buildingnumber,streetdescription[/.' ,]=add[/ .' ,]
method315	postcode	subbuildingname_paotext,buildingnumber,streetdescription[/.']=add[/. ']
method316	postcode	saotext_paotext_streetdescription_townname[/.']="APARTMENT "add[/.' ,]
method317	postcode	saotext,pp_streetdescription[/.']=add("FLAT" to "APARTMENT")[/.']

[Type here]

method318	postcode	"FLAT "_buildingname[/.']=add1_add2[/.']
method319	postcode	saotext_paotext[/.' ,]= "APARTMENT "add1[/.' /]
method320	postcode	saotext,pp_streetdescription[/.']= "APARTMENT "add1,add3[/.' /]
method321	postcode	subbuildngname_streetdescription[/.']=add1_add3("APARTMENT" to "FLAT")[/.' ,]
method322	postcode	saotext,pp_streetdescription[/.']=add1,add3("FLAT" to "APARTMENT")[/.' ,]
method323	postcode sector	pp,streetdescription[/.']=add[/.']
method324	postcode sector	buildingnumber,streetdescription,townname[/.']=add[/.']
method325	postcode sector	pp,streetdescription,townname[/.']=add[/.']
method326	postcode sector	buildingname,streetdescription[/.']=add[/.']
method327	postcode sector	buildingnumber,streetdescription,dependentlocality[/.']=add[/.']
method328	postcode sector	pp,streetdescription,dependentlocality[/.']=add[/.']
method329	postcode sector	buildingnumber,streetdescription,locality[/.']=add[/.']
method330	postcode sector	pp,streetdescription,locality[/.']=add[/.']
method331	postcode sector	buildingname,streetdescription,dependentlocality[/.']=add[/.']
method332	postcode sector	buildingname,streetdescription,locality[/.']=add[/.']
method333	postcode sector	paotext,streetdescription,dependentlocality[/.']=add[/.']
method334	postcode sector	paotext,streetdescription,locality[/.']=add[/.']
method335	postcode sector	subbuildingname_buildingname[/.']=add[/.']

[Type here]

method336	postcode sector	buildingname,dependentlocality[/.']=add[/.']
method337	postcode sector	buildingname,locality[/.']=add[/.']
method338	postcode sector	subbuildingname_buildingname,streetdescription[/.']=add[/.']
method339	postcode sector	saotext_buildingname,streetdescription[/.']=add[/.']
method340	postcode sector	subbuildingname_ss,paotext,streetdescription[/.']=add[/.']
method341	postcode sector	paotext,streetdescription[.']=add[.']
method342	postcode sector	subbuildingname,buildingnumber,streetdescription[/.']=add[/.']
method343	postcode sector	saotext,buildingnumber,streetdescription[/.']=add[/.']
method344	postcode sector	subbuildingname,buildingnumber,streetdescription,dependentlocality[/.']=add[/.']
method345	postcode sector	saotext,buildingnumber,streetdescription,dependentlocality[/.']=add[/.']
method346	postcode sector	subbuildingname,buildingnumber,streetdescription,locality[/.']=add[/.']
method347	postcode sector	subbuildingname_buildingname,buildingnumber,streetdescription[/.']=add[/.']
method348	postcode sector	buildingname,paotext[/.']=add[/.']
method349	postcode sector	buildingnumber,streetdescription[/.']=add[/.']
method350	postcode sector	saotext,paotext,streetdescription[/.']=add[/.']
method351	postcode sector	saotext,paotext,buildingnumber, streetdescription[/.']=add[/.']
method352	postcode sector	saotext,paotext,buildingnumber_streetdescription[/.']=add[/.']

[Type here]

method353	postcode sector	saotext,paotext,pp_streetdescription[/.']=add[/.']
method354	postcode sector	saotext,paotext[/.']=add[/.']
method355	postcode sector	saotext,paotext,paostartnumber- paoendnumber[/.']=add1,add2,add3(first word)[/.']
method356	postcode sector	saotext_streetdescription[/.']=add[/.']
method357	postcode sector	saotext_paotext[,./]="APARTMENT "add1[,.']
method358	postcode sector	subbuildingname,paotext,streetdescription[/.']=add[/.']
method359	postcode sector	"STUDIO "_subbuildingname_paotext_buildingnumber_streetdescription[,./]=a dd[-/.']
method360	postcode sector	saotext,paotext[/.']="FLAT "add1[,./]
method361	postcode sector	saotext,buildingname,streetdescription,dependentlocality[/.']=add[/.']
method362	postcode sector	subbuildingname,buildingnumber_streetdescription,dependentlocality[/.']=add[]
method363	postcode sector	ss,streetdescription[]=add[]
method364	postcode sector	saotext,paotext,paostartnumber- paoendnumber_streetdescription[/.']=add[/.']
method365	postcode sector	subbuildingname,paotext,paostartnumber- paoendnumber_streetdescription[/.']=add[/.']
method366	postcode sector	saotext,"NO "_pp_streetdescription[/.']=add[/.']
method367	postcode sector	saotext,pp_streetdescription[/.']=add1,add3[/.']
method368	postcode sector	saotext,buildingnumber_streetdescription[/.']=add[/.']

[Type here]

method369	postcode sector	subbuildingname,buildingnumber_streetdescription[/.']=add1,add2[/.']
method370	postcode sector	saotext,pp_streetdescription[/.']=add[/.']
method371	postcode sector	subbuildingname,buildingnumber_streetdescription[/.']=add[/.']
method372	postcode sector	subbuildingname_paotext,buildingnumber_streetdescription[/.']=add[/.']
method373	postcode sector	saotext,paostartnumber-paoendnumber_streetdescription[/.']=add[/.']
method374	postcode sector	ss_paotext,streetdescription[/.']=add[/.']
method375	postcode sector	saotext,paotext,locality[/.']=add[/.']
method376	postcode sector	saotext,pp_streetdescription[/.']="FLAT "add[/.']
method377	postcode sector	saotext,pp_streetdescription[/.']="APARTMENT "add[/.']
method378	postcode sector	subbuildingname,buildingnumber_streetdescription[/.']=add(change "ROOM" to "FLAT")[/.']
method379	postcode sector	subbuildingname,buildingnumber_streetdescription[/.']=add(change "FLAT" to "ROOM")[/.']
method380	postcode sector	subbuildingname,buildingnumber_streetdescription[/.']=add(change "STUDIO" to "FLAT")[/.']
method381	postcode sector	"FLAT subbuildingname,paotext,buildingnumber_streetdescription[/.']=add[/.']
method382	postcode sector	saotext,buildingname_streetdescription[/.']=add(change "UNIT" to "FLAT")[/.']
method383	postcode sector	subbuildingname,paotext,buildingnumber_streetdescription[/.']=add(change "STUDIO" to "FLAT")[/.']
method384	postcode sector	subbuildingname,buildingnumber_streetdescription[/.']=add[/.']

[Type here]

method385	postcode sector	subbuildingname,buildingnumber_streetdescription[/.']=add[/.']
method386	postcode sector	saotext,pp_streetdescription[/.']=add(change "APARTMENT" to "FLAT")[/.']
method387	postcode sector	ss,paotext,streetdescription[/.']=add[/.']
method388	postcode sector	buildingname[/.']=add[,./]
method389	postcode sector	subbuildingname,buildingnumber_streetdescription[/.']="FLAT" add1(first word),add2
method390	postcode sector	saotext,pp_streetdescription[/.']=add1,add3(change "APARTMENT" to "FLAT")[/.']
method391	postcode sector	subbuildingname,paostartnumber- paoendnumber_streetdescription[/.']=add1,add3[./]
method392	postcode sector	saotext_paotext,pp_streetdescription[/.']=APARTMENT"add[./]
method393	postcode sector	saotext_paotext,pp_streetdescription[/.']=add[./]
method394	postcode sector	subbuildingname_buildingname,buildingnumber,streetdescription[/.']= add[./]
method395	postcode sector	subbuildingname,paotext,pp,streetdescription[/.']=add[./]
method396	postcode sector	subbuildingname,paotext,pp_streetdescription[/.']=add[./]
method397	postcode sector	"FLAT "buildingname,streetdescription[/.']=add[./]
method398	postcode sector	"FLAT "subbuildingname,paostartnumber- paoendnumber_streetdescription[/.']=add[./]
method399	postcode sector	"FLAT "buildingname,streetdescription[/.']=add[./]
method400	postcode sector	saotext,pp_streetdescription[/.']=add[./]
method401	postcode sector	buildingname,subbuildingname,pp_streetdescription[/.']=add[/.']

[Type here]

method402	postcode sector	subbuildingname_buildingname,streetdescription[/.']=add[/.']
method403	postcode sector	subbuildingname,paostartnumber- paoendnumber,streetdescription[/.']=add[/.']
method404	postcode sector	ss_paotext,streetdescription[/.']=add1,add2[/.']
method405	postcode sector	saotext,pp_paotext,streetdescription[/.']=add[/.']
method406	postcode sector	subbuildingname,buildingname,buildingnumber_streetdescription[-,/.'] =add[-,/.']
method407	postcode sector	subbuildingname,buildingname,streetdescription[/.']=add(change "STUDIO" to "FLAT")[/.']
method408	postcode sector	saotext_paotext,streetdescription,locality[/.']="FLAT"_add[/.']
method409	postcode sector	subbuildingname,buildingname,locality,townname[/.']=add[/.']
method410	postcode sector	saotext_buildingnumber_streetdescription[/.']=add1[/.']
method411	postcode sector	subbuildingname,pp_streetdescription[/.']=add[/.']
method412	postcode district (First 3 characters)	subbuildingname,pp_streetdescription[/.']=add[/.']
method413	postcode district (First 2 characters)	subbuildingname,pp_streetdescription[/.']=add[/.']
method414	postcode sector	saotext,buildingname,buildingnumber_streetdescription[/.']=add1[-,a dd2,add3[/.']
method415	postcode sector	saotext,pp_streetdescription(change "STUDIO FLAT" to "APARTMENT" or change "ROOM"to "APARTMENT")[/.']=add[/.']
method416	postcode sector	saotext_streetdescription[/.']="FLAT"_add[/.']
method417	postcode sector	saotext,pp_streetdescription[/.']=add1,add3[/.']

[Type here]

method418	postcode sector	"CHALET"_ss,paotext,streetdescription,townname[/.']=add[/.']
method419	postcode sector	pp_streetdescription[/.']=add2[/.']
method420	postcode sector	"FLAT"_buildingname_streetdescription[/.']=add1[/.']
method421	postcode sector	saotext_paotext[,./]=add1[,./]
method422	postcode sector	saotext_streetdescription[/.']=add1[,./]
method423	postcode sector	"APARTMENT"_buildingname_streetdescription[/.']=add1_add2_add3[/.']
method424	postcode sector	pp_streetdescription[,./]=add[,./]
method425	postcode sector	buildingnumber_streetdescription,townname[,./]=add[,./]
method426	postcode sector	saotext,paotext,pp_streetdescription[/.']=add1[,],add2[]
method427	postcode sector	buildingname(frist word)_subbuildingname_buildingname(all string after first word),pp,streetdescription[,./]=add1[,],add2[/.' ,]
method428	postcode sector	buildingname(frist word),streetdescription[,./]=add1,add2[/.' ,]
method429	postcode sector	buildingname,streetdescription[/.']=add[/.']
method430	postcode sector	saotext,paotext,paostartnumber- paoendnumber_streetdescription[]=add1[,],add2[]
method431	postcode sector	saotext_streetdescription[/.']=add1_add3[/.']
method432	postcode	paotext(last word)_paotext(first two words),locality,townname[/.']=add1[,],add2,add3[/.']
method433	postcode	saotext_paotext_streetdescription_townname[,./]=add[,./]
method434	postcode	buildingname(last word)_subbuildingname_buildingname(frist two words),pp,streetdescription,pp,streetdescription[]=add[]

[Type here]

method435	postcode	saotext,paotext,pp_streetdescription[/.']=add(change "FLAT" to "APARTMENT")[/.']
method436	postcode	subbuildingname,buildingnumber,streetdescription[/.']=add1(first two words),add2[,.']
method437	postcode	saotext,paotext,pp_streetdescription[/.']="FLAT" _add[/.' ,]
method438	postcode	saotext,pp,streetdescription[/.']=add(change "FLAT" to "ROOM")[/.']
method439	postcode	saotext,paotext,pp_streetdescription[/.']=add(change "FLAT" to "ROOM")[/.']
method440	postcode	saotext,paostartnumber-paoendnumber_streetdescription[/.']=add(change "STUDIO" to "STUDIO FLAT" or change "CLUSTER" to "CLUSTER FLAT")[/.']
method441	postcode	saotext,paostartnumber-paoendnumber_streetdescription[/.']=add(change "FLAT" to "ROOM")[/.']
method442	postcode	saotext,paostartnumber-paoendnumber_streetdescription[/.']=add(change "APARTMENT" to "FLAT")[/.']
method443	postcode	saotext,streetdescription[/.']="APARTMENT" _add1[,],add2[.' /]
method444	postcode	"FLAT" _paostartsuffix_paostartnumber,streetdescription[/.']=add1,add 3[.' /]
method445	postcode	paotext(last word),townname[/.']=add1,add3[/.']
method446	postcode	paotext(last word)_paotext(first two word),townname[/.']=add1,add3[/.']

[Type here]

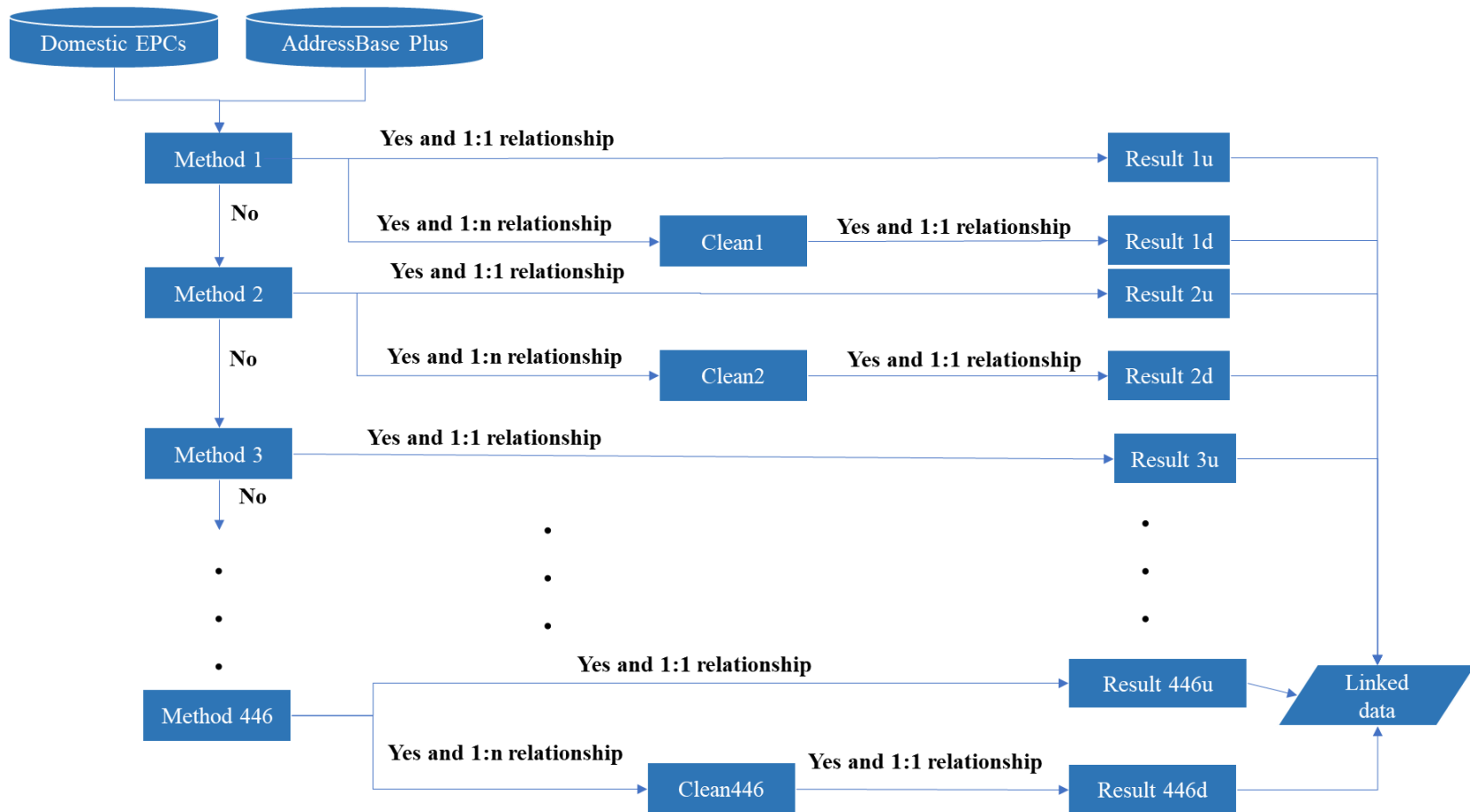


Figure B1 A Brief view of the workflow of data linkage between Domestic EPCs and AddressBase Plus

[Type here]

Table B2 The cleaning method for each linkage

Cleaning ID	Clean method
clean1	keep the residential uprn
clean2	keep the residential uprn and saostartnumber is null
clean4	keep the residential uprn
clean5	keep the residential uprn
clean6	keep the residential uprn
clean7	keep the residential uprn
clean8	keep the residential uprn
clean9	keep the residential uprn
clean10	keep the residential uprn
clean11	keep the residential uprn
clean12	keep the residential uprn
clean15	keep the residential uprn
clean20	keep the residential uprn
clean25	keep the residential uprn
clean30	keep the residential uprn
clean32	keep the residential uprn
clean34	keep the residential uprn
clean35	keep the residential uprn
clean38	keep the residential uprn
clean44	keep the residential uprn
clean45	keep the residential uprn
clean58	buildingnumber does not contain number
clean73	keep the residential uprn
clean74	keep the residential uprn
clean75	keep the residential uprn
clean78	keep the residential uprn
clean79	keep the residential uprn
clean82	keep the residential uprn
clean87	keep the residential uprn
clean96	keep the residential uprn
clean97	keep the residential uprn
clean101	keep the residential uprn
clean103	keep the residential uprn
clean104	keep the residential uprn
clean107	keep the residential uprn
clean108	keep the residential uprn
clean109	keep the residential uprn

[Type here]

clean113	keep the residential uprn or buildingnumber is equal to paostartnumber
clean115	keep the residential uprn
clean123	keep the residential uprn
clean124	keep the residential uprn
clean125	keep the residential uprn
clean142	keep the residential uprn
clean143	the last word in add2 is equal to last word in subbuildingname, or the first word in add2 is equal to first word in buildingnumber, or the first word in add2 is equal to last word of subbuildingname, or residential uprn, or first three word in add2 is equal to combine the subbuildingname and buildingnumber with a blank space, or add2=saotext_streetdescription
clean144	add1(last word)=subbuildingname(last word);
clean145	keep the residential uprn
clean146	keep the residential uprn
clean147	add2(first word)=paostartnumber
clean151	add3(first word)=paostartnumber
clean153	add2(first word)=ss;
clean161	add3(first word)=saostartnumber
clean169	add2(first word)=pp or add2(last word)=saostartnumber or add2(first word)=buildingnumber or add2(first word)=saostartnumber or add2(first word)=ss
clean171	add2(first word)=paostartnumber-paoendnumber
clean172	add2(first two words)=buildingname or add2=saostartnumber_paotext or add2(last two words)=subbuildingname
clean175	keep the residential uprn
clean180	add2(first word)=paostartnumber and add2 starts with number strings
clean196	add1[,](strings after the first word)=paotext or add1[,](first second and third words)=paotext or
clean197	add1[,](strings after the first word)=paotext
clean202	add2=buildingname or add2(first three words)=paotext(first three words)
clean203	add2[,](first word)=buildingnumber
clean214	add1[,](text after first word)=paotext
clean217	keep the residential uprn
clean234	keep the residential uprn
clean236	keep the residential uprn
clean237	keep the residential uprn
clean239	keep the residential uprn
clean240	keep the residential uprn
clean246	add2[,](first word)=paostartnumber or add2[,](first word)=paostartnumber-paoendnumber

[Type here]

clean253	add2(first word)=ss or add2=paotext after correct some add2; or add2=paotext(text after first word)
clean254	add2[,](first two words)=saotext or add2[,](last two words)=saotext;
clean255	add2[,](first word)=buildingnumber
clean271	add2[,](first word)=saostartnumber or add2[,](first word)=buildingnumber or add2[,](first word)=paostartnumnber
clean272	add2[,](first word)=paostartnumber
clean281	add2(first word)=saostartnumber or add2(first word)=paostartnumber or add2(first word)=saostartnumber-saoendnumber or add2(first word)=saostartnumber or add2[,](first word)=ss or add2(first word)=paostartnumber-paoendnumber or add2=paotext or add1[,](text afte the second word)=saostartnumber_paotext or add2[,](first word)=pp or add2[,](first two words)="BLOCK" saostartnumber or add2[,](second word)=paostartnumber or add3[,](first word)=paotext(last word) or add3[,](first word)=paostartnumber or add2[,](first word)=saostartnumber or add3[,](first word)=paostartnumber-paoendnumber or add3[,](first word)=saostartnumber-saoendnumber or add1(text after second word)="BLOCK"_saostartnumber or add1(text after second word)=buildingnumber or add1(text after second word)=ss or add1(first two words after second word)=paotext(first two word) or add1(first two words after second word)=paotext(second and third words) or add1[,](the third word)=paostartnumber-paoendnumber or add1[,](third and forth words)=paotext(first two words) or add1(third word)=paotext(first word)
clean288	add1(text after first word)=saotext_streetdescription or add1[,](text after first word)=subbuildingname_saotext or (add1[,](second word)=paostartnumber-paoendnumber and add1(first word)=saostartnumber) or add1[,](second word)=paotext(first word) or add1[,](frist word)=ss or correct some address string then add2(second word)=streetdescription(first word) or residential uprn or add1[,](second word)=saotext(last word) or add1[,](second and third word)=saotext(last two words) or add1[,](second word)=saotext(first word)
clean289	add1(second and third words)=buildingname and correct the "CHEPLING HOUSE"to "CHEPING HOUSE"; or add1[,](first three words)=saotext or add2=paostartnumber_streetdescription or add1=saostartnumber_paotext
clean291	add2[,](frist word)=pp or add1[,](second word)=buildingnumber or add1[,](second word)=paostartnumber-paoendnumber or add1[,](second word)=paostartnumber or add1[,](second word)=paotext(second word) or add1[,](second word)=paotext(first word) or add1[,](second word)=paotext(second word) or add1[,](second word)=paotext(first word) or add1[,](text after first word)=paotext or add1[,](second and third word)=paotext(first two words)
clean292	add2=paotext or add2[,](first two words)=paotext(frist two words)
clean293	add2(first word)=paostartnumber or add1(third and forth words)=paotext(first two words) or add2=paotext or add2=pp_streetdescription or
clean294	keep the residential uprn or add2(first word)=ss or add2[,](first word)=paostartnumber or add1[,](second word)=saostartnumber

[Type here]

clean295	keep the residential uprn or add3[,](first word)=paostartnumber
clean296	keep the residential uprn
clean301	add2(first word)=paotext(first word)
clean319	add2(first word)=paostartnumber or add3(first word)=buildingnumber
clean321	add2[,](first word)=buildingnumber or add2[,](last word)=buildingnumber
clean324	keep the residential uprn
clean348	keep the residential uprn
clean349	keep the residential uprn
clean357	add2[,](first word)=paostartnumber
clean367	add2[,](first word)=ss
clean390	add2=paotext or add2(first word)=saostartnumber
clean391	add2=buildingname(first two words)
clean404	keep the residential uprn
clean419	keep the residential uprn
clean420	add2[,](frist word)=paostartnumber
clean422	keep the residential uprn
clean431	add2=paotext or add2(second and third words)=paotext(first two words)
clean436	add1[,](third word)=paostartnumber or add1[,](first two words)= subbuildingname and add2(first word)=paostartnumber or add1[,](third word)=paotext
clean444	add2=paotext or add2(first two words)=paotext(second and third words)
clean446	keep the residential uprn

Appendix C

Table C1 New address variables created from existing address field in LR PPD and OS AddressBase Plus

Type	New variable	Method	Data
Combine	SAONPAON	Combine SAON and PAON with a blank space	LR PPD
	PAONSTREET	Combine PAON and street with a blank space	LR PPD
	SAONSTREET	Combine PAON and street with a blank space	LR PPD
	bb	Combine buildingname and buildingnumber, using a comma	OS AddressBase Plus
	pp	Combine paostartnumber and paostartsuffix	OS AddressBase Plus
	subbname	combine subbuildingname and builidngname with a blank space	OS AddressBase Plus
	pp2	Combining paotext and pp fields using a comma	OS AddressBase Plus

[Type here]

pp4	Combine paostartnumber and paostartsuffix using a hyphen	OS AddressBase Plus
ppp	Combine paotext and pp4 with a blank space	OS AddressBase Plus
ss	Combine saostartnumber and saostartsuffix	OS AddressBase Plus
ss1	Combine saostartsuffix and saostartnumber	OS AddressBase Plus
ss2	Combine saostartnumber and saoendnumber with a hyphen	OS AddressBase Plus
subss	Combine subbuildingname and ss with a blank space	OS AddressBase Plus
saopp	Combine saotext and pp with a comma and a blank space	OS AddressBase Plus
sp	Combine ss and paotext fields using a blank space	OS AddressBase Plus
ssp	Combine saotext and saostartnumber fields with a comma and a blank space, then combine with paotext	OS AddressBase Plus
saobui	Combine saotext and buildingname fields using a blank space	OS AddressBase Plus
SAONSTREET	Combine SAON and street fields with a comma and a blank space	LR PPD
paostartnumber1	Combine the paostartnumber and string1	OS AddressBase Plus
newp	Combine pp and string1 fields and remove all the blank spaces	OS AddressBase Plus
ssend	Combine the saoendnumber and saoendsuffix	OS AddressBase Plus
sspaotext	Combine the ss and paotext with blank space	OS AddressBase Plus
subbuildingnamepaotext	Combine the subbuildingname and paotext	OS AddressBase Plus
psao	Combine the paostartnumber and saotext1	OS AddressBase Plus

[Type here]

Strip	PAON1	Stripping surrounding whitespace from hyphens and the comma in PAON field.	LR PPD
	PAON2	Stripping surrounding whitespace from hyphens in PAON field	LR PPD
	SAON2	Stripping surrounding whitespace in SAON field	LR PPD
	SAON9	Deleting the 'THE ' string in SAON	LR PPD
	pp11	Deleting whitespace from the beginning and end of pp new field	OS AddressBase Plus
	SAON4	Deleting all the full stop in SAON	LR PPD
	PAON4	Deleting all commas in POAN	LR PPD
	buildingname1	Deleting the full stop in buildingname	OS AddressBase Plus
	saotext1	Deleting the 'FLAT ' leading string in saotext	OS AddressBase Plus
Prepend string	FLATSAON	Prepend the SAON with 'FLAT ' string	LR PPD
	FLATPAON	Prepend the PAON with 'FLAT ' string	LR PPD
	UNITPAON	Prepend the PAON with 'UNIT ' string	LR PPD
	flatpao1	Prepend the paostartsuffix with 'FLAT ' string	OS AddressBase Plus
	apsaon	Prepend the SAON with 'APARTMENT ' string	LR PPD
	flatss	Prepend the ss with 'FLAT ' string	OS AddressBase Plus
	flatsub	Prepend the subbuildingname with 'FLAT ' string	OS AddressBase Plus
	unitss	Prepend the ss with 'UNIT ' string	OS AddressBase Plus
	apss	Prepend the saotext with 'APARTMENT ' string	OS AddressBase Plus
	fnewp	Prepend the newp with 'FLAT ' string	OS AddressBase Plus
Replace	paostartnumber1	Prepend the paostartnumber with 'FLAT ' string	OS AddressBase Plus
	subbuildingnamenew	Replace 'UNIT' or 'APARTMENT' string in subbuildingname to 'FLAT ' string	OS AddressBase Plus
	subbuildingnamenew1	Replace 'FLAT ' string in subbuildingname to 'APARTMENT ' string	OS AddressBase Plus

[Type here]

	SAON6	Replace "STORE FLAT" to "FLAT" string in SAON	LR PPD
	saotext5	Replace "FLAT" to "UNIT" string in saotext	OS AddressBase Plus
	saotext6	Replace "FLAT" to "APARTMENT" string in saotext	OS AddressBase Plus
	saotext9	Replace "GARDEN FLAT" to "GARDEN FLOOR FLAT" string in saotext	OS AddressBase Plus
	saotext4	Replace "FLAT FIRST FLOOR" to "FIRST FLOOR FLAT" string in saotext	OS AddressBase Plus
	saotext7	Replace "APARTMENT" to "FLAT" string in saotext	OS AddressBase Plus
Extract	string1	For a address string follow the string pattern start with 'FLAT ' and then follow with one-letter words in saotext , only keep this one-letter word	OS AddressBase Plus
	saotext3	Extract the first two words in saotext	OS AddressBase Plus
	SAON22	Extract the first two words in SAON	LR PPD
	SAON3	Extract the last word in SAON	LR PPD
	subbuildingname3	Extract the last word in subbuildingname	OS AddressBase Plus
	SAON5	Extract the last two words in SAON	LR PPD
	saotext8	Extract the last word in saotext	OS AddressBase Plus
	PAON3	Extract the last word in PAON	LR PPD
	SAON8	Extract the first two word in SAON	LR PPD

[Type here]

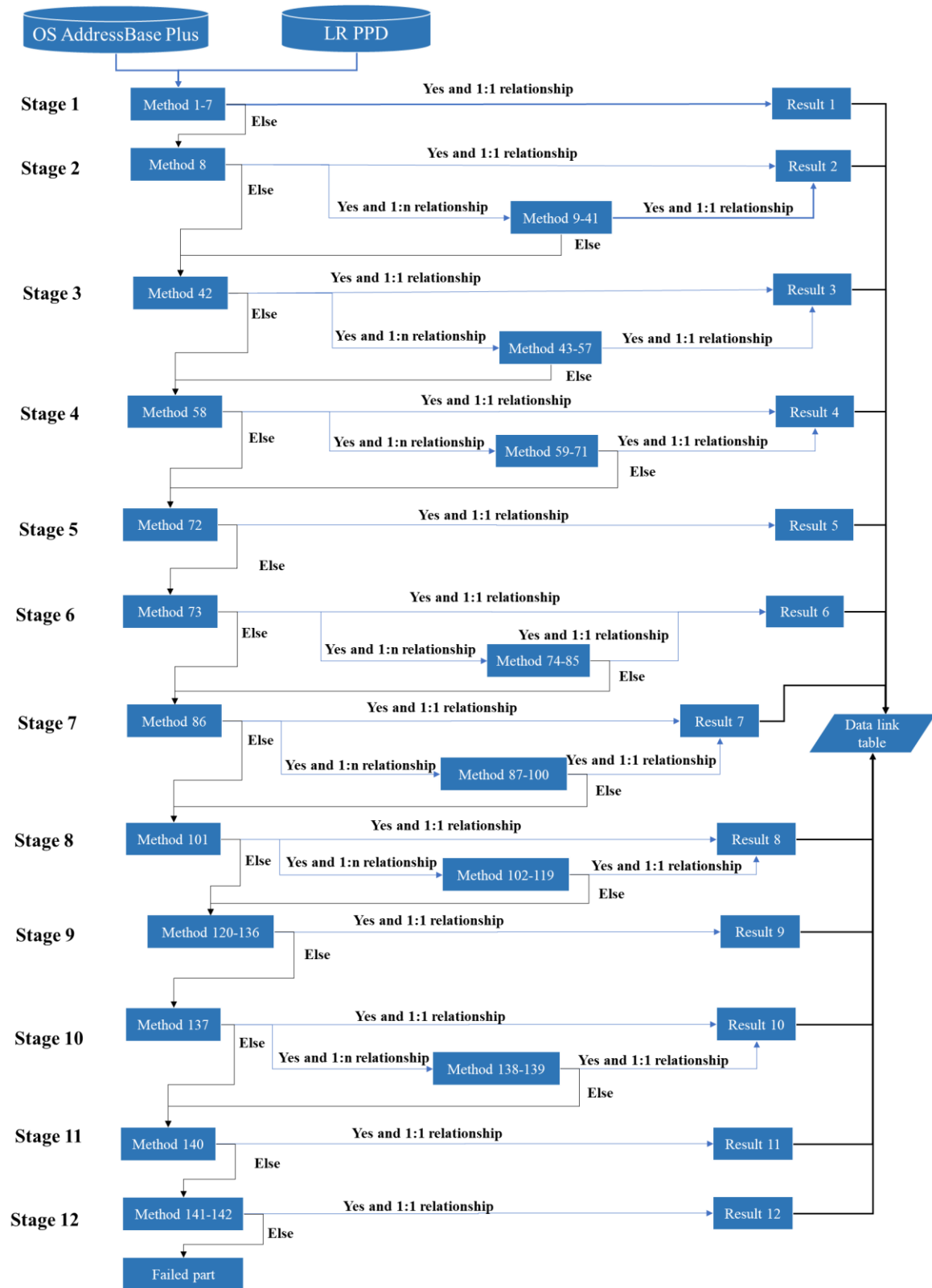


Figure C1 Master workflow for 12 stage data linkage process

Table C2 Details of the 142 matching rules in the 12 stages

Stage No.	Method No.	Conditional	Matching rule
Stage 1	Method 1	PAON is NULL and SAON is not NULL	SAON is equal to pp
	Method 2	PAON is NULL and SAON is not NULL	SAON is equal to paostartnumber1
	Method 3	PAON is NULL and SAON is not NULL	SAON is equal to ss
	Method 4	PAON is NULL and SAON is not NULL	SAON is equal to buildingname (remove blank space for both side)
	Method 5	PAON is NULL and SAON is not NULL	SAON is equal to paotext
	Method 6	PAON is NULL and SAON is not NULL	SAONSTREET is equal to buildingname
	Method 7	PAON is NULL and SAON is not NULL	SAON9 is equal to buildingname (remove blank space for both side)
Stage 2	Method 8		PAON is equal to buildingname or buildingnumber or bb
	Method 9	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is NULL and saotext is equal to buildingname
	Method 10	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to pp and SAON is equal to saotext
	Method 11	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to buildingname and SAON is equal to saotext
	Method 12	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to buildingnumber and SAON is equal to saotext

[Type here]

Method 13	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to bb and SAON is equal to saotext
Method 14	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to buildingname and SAON is equal to subbuildingname
Method 15	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to buildingnumber and SAON is equal to subbuildingname
Method 16	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to bb and SAON is equal to subbuildingname
Method 17	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to paotext and SAON is equal to ss
Method 18	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to bb and SAON is equal to ss
Method 19	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to buildingname and FLATSAON is equal to subbuildingname
Method 20	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to paotext and FLATSAON to saotex
Method 21	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to buildingname and FLATSAON is equal to subbuildingnamenew
Method 22	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to buildingname and SAON is equal to fss

[Type here]

Method 23	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to paotext and SAON is equal to fss
Method 24	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to bb and SAON is equal to fss
Method 25	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	SAONPAON is equal to buildingname
Method 26	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to paotext and SAON is equal to saotext
Method 27	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to buildingname and SAON is equal to ss
Method 28	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	SAONPAON is equal to subbname
Method 29	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to bb and FLATSAON is equal to saotext
Method 30	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to paotext and SAON is equal to pp
Method 31	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to buildingname and SAON is equal to pp11
Method 32	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to buildingname and FLATSAON is equal to subbuildingnamenew1

[Type here]

	Method 33	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to buildingname and SAON is equal to newp
	Method 34	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to buildingname and SAON is equal to fnewp
	Method 35	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to buildingname and SAON is equal to subbuildingname (remove blank space for both side)
	Method 36	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to buildingname and SAON is equal to saotext5 (remove blank space for both side)
	Method 37	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to bb and SAON is equal to apss (remove blank space for both side)
	Method 38	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to buildingname and SAON6 is equal to saotext (remove blank space for both side)
	Method 39	(PAON is equal to buildingname or buildingnumber or bb) and SAON is not NULL	PAON is equal to buildingname and SAON is equal to ssend (remove blank space for both side)
	Method 40	(PAON is equal to buildingname or buildingnumber or bb) and SAON is NULL	PAON is equal to pp
	Method 41	(PAON is equal to buildingname or buildingnumber or bb) and SAON is NULL	PAON is equal to paotext
Stage 3	Method 42	PAON is equal to paostartnumber	one to one linkage result
	Method 43	PAON is equal to paostartnumber	PAON is equal to paostartnumber and SAON is equal to flatpao1

[Type here]

	Method 44	PAON is equal to paostartnumber	PAON is equal to pp and SAON is equal to saotext
	Method 45	PAON is equal to paostartnumber	PAON is equal to pp and FLATSAON is equal to saotext
	Method 46	PAON is equal to paostartnumber	PAON is equal to pp and STREET is equal to streetdescription
	Method 47	PAON is equal to paostartnumber	PAON is equal to pp and SAON2 is equal to subbuildingname
	Method 48	PAON is equal to paostartnumber	PAON is equal to pp and SAON2 is equal to saotext
	Method 49	PAON is equal to paostartnumber	PAON is equal to pp and SAON2 is equal to saotext3
	Method 50	PAON is equal to paostartnumber	PAON is equal to pp and SAON is equal to saotext9
	Method 51	PAON is equal to paostartnumber	PAON is equal to pp and SAON is equal to saotext4
	Method 52	PAON is equal to paostartnumber, SAON contains "SECOND FLOOR FLAT" and saotext contains "SECOND FLOOR"	PAON is equal to pp and SAON is equal to saotext
	Method 53	PAON is equal to paostartnumber, SAON contains "FIRST AND SECOND FLOOR" and saotext contains "1ST AND 2ND FLOOR"	PAON is equal to pp and SAON is equal to saotext
	Method 54	PAON is equal to paostartnumber	PAON is equal to pp and SAON is equal to paotext_ss, then remove all the blank space
	Method 55	PAON is equal to paostartnumber	PAON is equal to pp and SAON is equal to saotext6
	Method 56	PAON is equal to paostartnumber	PAON is equal to pp and SAON is equal to ss
	Method 57	PAON is equal to paostartnumber	PAON is equal to pp and SAON3 is equal to saotext8
Stage 4	Method 58		PAON is equal to pp

[Type here]

	Method 59	PAON is equal to pp	paon is equal to pp , SAON is equal to saotext
	Method 60	PAON is equal to pp	PAON is equal to pp and SAON is equal to subbuildingname
	Method 61	PAON is equal to pp	PAON is equal to pp and SAON is equal to flatss
	Method 62	PAON is equal to pp	PAON is equal to pp and SAON is equal to sspaotext
	Method 63	PAON is equal to pp	PAON is equal to pp and FLATSAON is equal to saotext
	Method 64	PAON is equal to pp	PAON is equal to pp and FLATSAON is equal to subbuildingname
	Method 65	PAON is equal to pp	PAON is equal to pp and SAON is equal to ss
	Method 66	PAON is equal to pp	PAON is equal to pp and SAON is equal to saotext5
	Method 67	PAON is equal to pp	PAON is equal to pp and SAON is equal to saotext6
	Method 68	PAON is equal to pp	PAON is equal to pp and SAON is equal to paoendnumber
	Method 69	PAON is equal to pp	PAON is equal to pp and SAON3 is equal to subbuildingname
	Method 70	PAON is equal to pp	PAON is equal to pp and SAON is equal to subbuildingnamepaotext
	Method 71	PAON is equal to pp	PAON is equal to pp and SAON3 is equal to subbuildingname3
	Method 72		PAON is equal to psao
Stage 6	Method 73		PAON is equal to pp2
	Method 74	PAON is equal to pp2	PAON is equal to pp2 and SAON is equal to saotext
	Method 75	PAON is equal to pp2	PAON is equal to pp2 and SAON is equal to ss
	Method 76	PAON is equal to pp2	PAON is equal to pp2 and SAON is equal to flatss

[Type here]

	Method 77	PAON is equal to pp2	PAON is equal to pp2 and FLATSAON is equal to saotext
	Method 78	PAON is equal to pp2	PAON is equal to pp2 and SAON is equal to unitss
	Method 79	PAON is equal to pp2	PAON is equal to pp2 and SAON is equal to subbuildingname
	Method 80	PAON is equal to pp2	PAON1 is equal to buildingname and SAON is equal to subbuildingname
	Method 81	PAON is equal to pp2	Flat transactions: PAON is equal to pp2 and SAON is equal to ss1
	Method 82	PAON is equal to pp2	PAON is equal to pp2 and SAON4 is equal to saotext
	Method 83	PAON is equal to pp2	PAON is equal to pp2 and SAON5 is equal to saotext
	Method 84	PAON is equal to pp2	PAON is equal to pp2 and SAON is equal to saotext6
	Method 85	PAON is equal to pp2	PAON is equal to pp2 and SAON is equal to saotext5
Stage 7	Method 86		PAON is equal to paotext or PAON is equal to sp
	Method 87	PAON is equal to paotext or PAON is equal to sp	PAON is equal to paotext and SAON is equal to ss
	Method 88	PAON is equal to paotext or PAON is equal to sp	PAON is equal to paotext and FLATSAON is equal to saotext
	Method 89	PAON is equal to paotext or PAON is equal to sp	PAON is equal to paotext and SAON is equal to flatss
	Method 90	PAON is equal to paotext or PAON is equal to sp	PAON is equal to paotext and SAON is equal to saotext
	Method 91	PAON is equal to paotext or PAON is equal to sp	PAON is equal to paotext and SAON is equal to pp
	Method 92	PAON is equal to paotext or PAON is equal to sp	PAON is equal to paotext and SAON is equal to subss
	Method 93	PAON is equal to paotext or PAON is equal to sp	PAON is equal to paotext and SAONPAON is equal to saobui
	Method 94	PAON is equal to paotext or PAON is equal to sp	PAON is equal to paotext and SAON3 is equal to ss

[Type here]

	Method 95	PAON is equal to paotext or PAON is equal to sp	PAON is equal to paotext and UNITSAON is equal to saotext
	Method 96	PAON is equal to paotext or PAON is equal to sp	PAON is equal to paotext and SAON2 is equal to saotext3
	Method 97	PAON is equal to paotext or PAON is equal to sp	PAON is equal to paotext and SAON4 is equal to ss
	Method 98	PAON is equal to paotext or PAON is equal to sp	PAON is equal to paotext and SAON is equal to saotext7
	Method 99	PAON is equal to paotext or PAON is equal to sp	PAON is equal to paotext and apsaon is equal to saotext
	Method 100	PAON is equal to paotext or PAON is equal to sp	PAON is equal to paotext and SAON is equal to ss1
Stage8	Method 101		PAON1 is equal to buildingname or then PAON1 is equal to pp4
	Method 102	PAON1 is equal to buildingname or then PAON1 is equal to pp4	PAON1 is equal to buildingname and SAON is equal to subbuildingname
	Method 103	PAON1 is equal to buildingname or then PAON1 is equal to pp4	PAON1 is equal to buildingname and SAON is equal to saotext
	Method 104	PAON1 is equal to buildingname or then PAON1 is equal to pp4	PAON2 is equal to pp4 and SAON is NULL
	Method 105	PAON1 is equal to buildingname or then PAON1 is equal to pp4	PAON1 is equal to ppp and SAON is equal to ss
	Method 106	PAON1 is equal to buildingname or then PAON1 is equal to pp4	PAON1 is equal to ppp and SAON is equal to flats
	Method 107	PAON1 is equal to buildingname or then PAON1 is equal to pp4	PAON1 is equal to ppp and SAON is equal to saotext
	Method 108	PAON1 is equal to buildingname or then PAON1 is equal to pp4	PAON1 is equal to buildingname and FLATSAON is equal to subbuildingname
	Method 109	PAON1 is equal to buildingname or then PAON1 is equal to pp4	PAON1 is equal to buildingname and SAON is equal to flatsub

[Type here]

	Method 110	PAON1 is equal to buildingname or then PAON1 is equal to pp4	PAON1 is equal to buildingname and SAON is equal to ss
	Method 111	PAON1 is equal to buildingname or then PAON1 is equal to pp4	PAON2 is equal to pp4 and SAON is equal to saotext
	Method 112	PAON1 is equal to buildingname or then PAON1 is equal to pp4	PAON2 is equal to pp4 and FLATSAON is equal to saotext
	Method 113	PAON1 is equal to buildingname or then PAON1 is equal to pp4	PAON2 is equal to pp4 and SAON is equal to subbuildingname
	Method 114	PAON1 is equal to buildingname or then PAON1 is equal to pp4	PAON2 is equal to pp4 and SAON is equal to ssp
	Method 115	PAON1 is equal to buildingname or then PAON1 is equal to pp4	PAON1 is equal to buildingname and SAON3 is equal to saotext8
	Method 116	PAON1 is equal to buildingname or then PAON1 is equal to pp4	PAON1 is equal to buildingname and SAON8 is equal to ss
	Method 117	PAON1 is equal to buildingname or then PAON1 is equal to pp4	PAON is equal to buildingname and SAON3 is equal to saotext8
	Method 118	PAON1 is equal to buildingname or then PAON1 is equal to pp4	PAON is equal to buildingname and SAON2 is equal to saotext3
	Method 119	PAON1 is equal to buildingname or then PAON1 is equal to pp4	PAON2 is equal to buildingname and SAON3 is equal to saotext8
Stage9	Method 120	STREET is equal to paotext	PAON is equal to ss
	Method 121	STREET is equal to paotext	PAONSTREET is equal to buildingname
	Method 122	STREET is equal to paotext	PAONSTREET is equal to paotext
	Method 123	STREET is equal to paotext	PAONSTREET is equal to paotext and SAON is equal to ss

[Type here]

	Method 124	STREET is equal to paotext	PAON is equal to ss and SAON is equal to saotext
	Method 125	STREET is equal to paotext	FLATPAON is equal to subbuildingname
	Method 126	STREET is equal to paotext	UNITPAON is equal to saotext
	Method 127	STREET is equal to paotext	PAON is equal to saotext
	Method 128	STREET is equal to paotext	PAON3 is equal to ss
	Method 129	STREET is equal to paotext	SAONPOAN is equal to saotext
	Method 130	STREET is equal to paotext	SAONPOAN is equal to buildingname
	Method 131	STREET is equal to paotext	PAONSTREET is equal to buildingname1
	Method 132	STREET is equal to paotext	POANSAON is equal to saotext
	Method 133	STREET is equal to paotext	PAON4 is equal to saotext
	Method 134	STREET is equal to paotext	POAN1 is equal to ss2
	Method 135	STREET is equal to paotext	STREET is equal to paotext and POAN is equal to saotext3
	Method 136	STREET is equal to paotext	STREET is equal to paotext and PAON3 is equal to saotext8
Stag10	Method 137		PAON is equal to saopp
	Method 138		PAON is equal to saopp and SAON is equal to flatss
	Method 139		PAON is equal to saopp and SAON is equal to ss
Stage 11	Method 140		SAONPAON is equal to buildingname
stage 12	Method 141		PAON is equal to ss and SAON is NULL

[Type here]

	Method 142		PAONSTREET is equal to buildingname
--	---------------	--	-------------------------------------