

ORDNANCE SURVEY GB

OS Open Linked Identifiers - Getting Started Guide

Version	Date	Description
1.0	01/07/2020	Initial release

Purpose of this Document

This is the Getting Started Guide (from now on referred to as the 'Guide') for the OS Open Linked Identifiers product. This Guide provides information on the contents and structure of this product. For greater insight into the product and its potential applications, please refer to the OS Open Linked Identifiers Overview and Technical Specification.

The terms and conditions on which OS Open Linked Identifiers is made available to you and your organisation are contained within [Open Government Licence](#) which allows you to use OS Open Linked Identifiers in any way and for any purpose. We simply ask that you acknowledge us with "Contains OS data © Crown Copyright (2020)".

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

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

Copyright in this Guide

© Ordnance Survey Limited 2020. This Guide, (including for the avoidance of doubt, any mapping images reproduced within it), is protected by copyright and apart from the rights expressly granted within this document to use the content, all rights are reserved. Any part of this Guide may be copied for use internally in your organisation or business so that you can use OS Open Linked Identifiers under the terms of your licence (but not otherwise).

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

Trademarks

Ordnance Survey, OS, the OS Logos are registered trademarks and OS Open Linked Identifiers is a trademark of Ordnance Survey, Britain's mapping agency.

Contact details

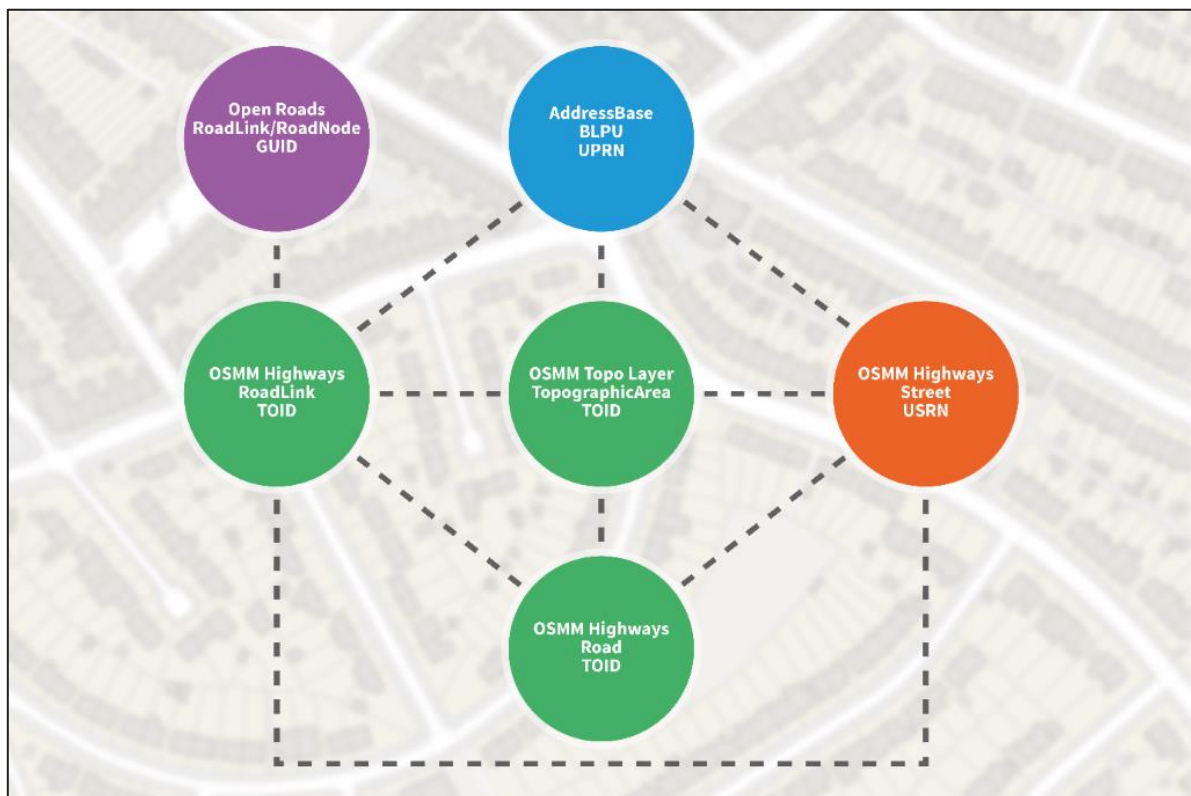
<https://www.ordnancesurvey.co.uk/contact-us>

Contents

- 1. Introduction..... 3
- 2. Linked Identifiers..... 4
 - 2.1 Additional Support Information 6
- 3. Instructions to install/load into a database application..... 6
 - 3.1 Pre-Requisites to using the Linked ID data 6
 - 3.2 PostgreSQL Database..... 6
 - 3.2.1 Preparation..... 6
 - 3.2.2 Create Schema..... 6
 - 3.2.3 Create Table 7
 - 3.2.4 Loading the data 9
 - 3.2.5 Viewing/Indexing the data 10

1. Introduction

OS Open Linked Identifiers is a dataset containing the authoritative relationships between Unique Property Reference Numbers (UPRNs), Unique Street Reference Numbers (USRNs) and Topographic Identifiers (TOIDs), and metadata. Identifiers are labels that are assigned to representations in a dataset. They are at the heart of how data can be effectively published, retrieved, reused and linked.



OS Open Linked Identifiers enables you to connect the variety of relationships between UPRN, USRNs and TOIDs. These are synchronised with comprehensive premium products to provide you with the most up-to-date feature metadata sourced from AddressBase, OS MasterMap Highways Network and OS MasterMap Topography Layer. The more an identifier is used in other datasets, the more valuable the primary data becomes.

For example, OS Open Identifiers enables visualisation of data linked to the Highways RoadLink and Open Roads products. By sharing the road link mapping between Highways and Open Roads it enables data that has been collected and shared against the Highways' premium products to be visualised in the open geometry.

Identifiers are crucial to the process of sharing information and linking together datasets, allowing them to fit into many workflows in many different types of workplace to provide powerful insights and support analytics.

2. Linked Identifiers

OS Open Linked Identifiers is aimed for use within database products only and consists of join tables that are supplied in the Comma Separated Variable (CSV) format. These files are not suitable for GIS packages as they are very large and do not contain any spatial information. This guide will cover the loading of an example linked ID CSV file into a typical database application.

Upon downloading the data, you will receive zip files of the 11 Linked Identifier relationships (see table below). The generalised naming convention for each Relationship ID is:

{Data Identifier 1}_{Feature Identifier 1}_{Data Identifier 2}_{Feature Identifier 2}_{CorrelationType Number}

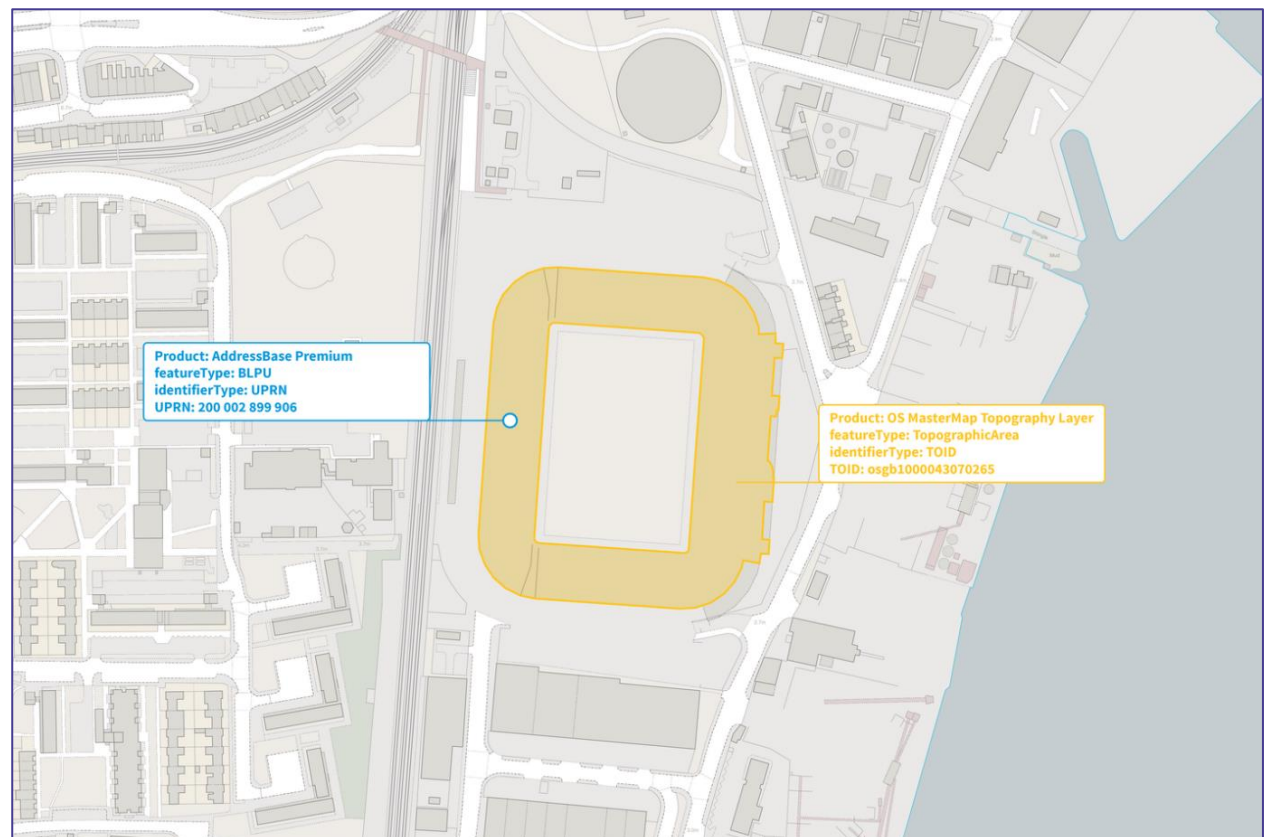
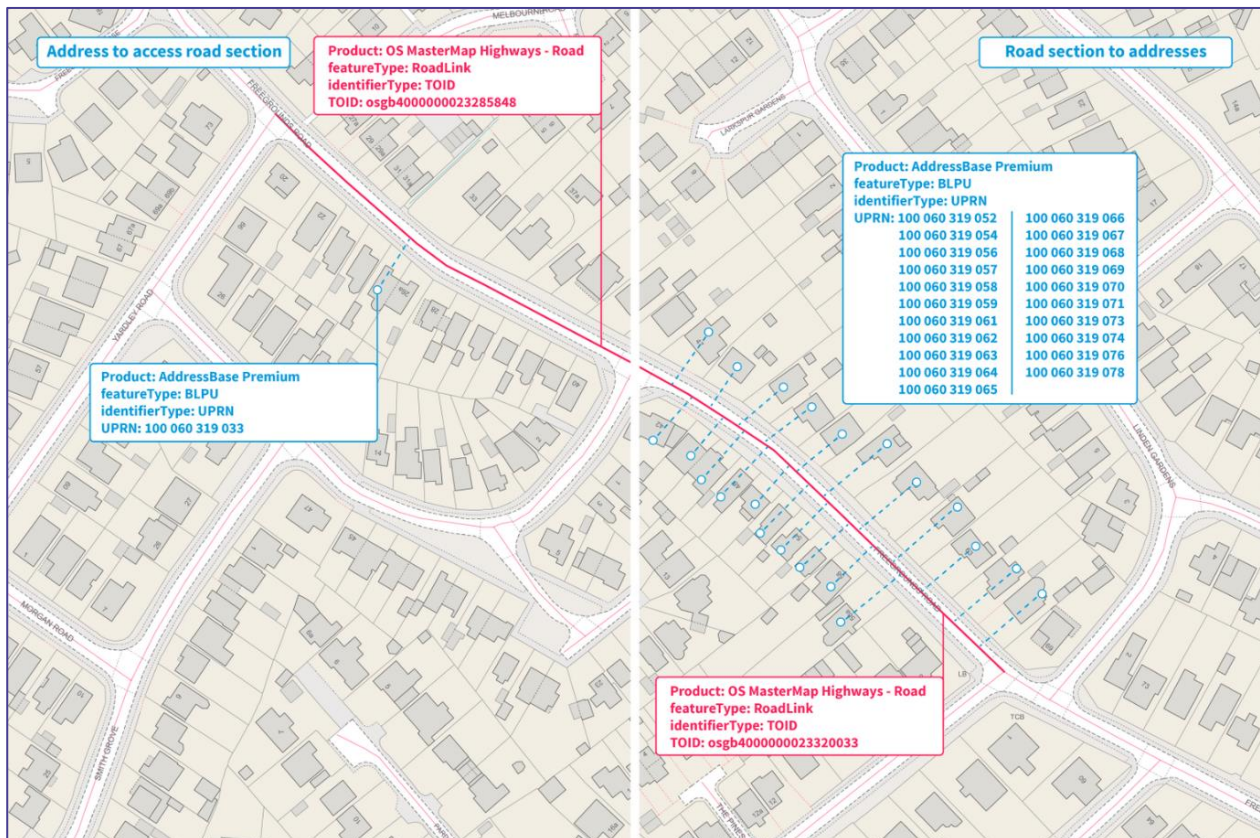
For example: *BLPU_UPRN_RoadLink_TOID_9*

Relationship IDs	Relationship
RoadLink_TOID_TopographicArea_TOID_2	RoadLink <-> TopographicArea
Road_TOID_TopographicArea_TOID_3	Road <-> TopographicArea
Street_USRN_TopographicArea_TOID_4	Street <-> TopographicArea
BLPU_UPRN_TopographicArea_TOID_5	BLPU <-> TopographicArea
RoadLink_TOID_Road_TOID_7	RoadLink <-> Road
RoadLink_TOID_Street_USRN_8	RoadLink <-> Street
BLPU_UPRN_RoadLink_TOID_9	BLPU <-> RoadLink
Road_TOID_Street_USRN_10	Road <-> Street
BLPU_UPRN_Street_USRN_11	BLPU <-> Street
ORRoadLink_GUID_RoadLink_TOID_12	Open Roads RoadLink <-> OSMM Highways RoadLink
ORRoadNode_GUID_RoadLink_TOID_13	Open Roads RoadNode <-> OSMM Highways RoadLink

It is up to the end user to choose which Linked ID they wish to use. For further information for the purpose of each join table, please refer to the *OS Linked Identifiers Overview* document. This will depend on which dataset they wish to link to elements within OS data. It is NOT recommended to attempt to open the uncompressed .CSV files in an application such as Microsoft Excel, as the application will simply either hang or not load the data at all.

This is also the case if the user tries to load the Linked IDs directly into a GI application. The GI application will quite likely hang because the Linked IDs are large and contain no spatial information (geometry) and will therefore not be spatially indexed.

Using these Linked Identifiers from within a database will allow the user to achieve end results such as shown below. The results will depend on which user data they wish to link, and which OS dataset is being linked.



The use of Linked Identifiers will also be made available through an [API service](#) which is not covered in this guide.

2.1 Additional Support Information

Within each ZIP file containing the Linked Identifier data in CSV format, there is a JSON file. This file contains important information providing the epoch or publication date of the differing datasets used to create the Linked Identifiers. This information can be used to ensure synchronicity between the Ordnance Survey datasets that are being linked.

3. Instructions to install/load into a database application

3.1 Pre-Requisites to using the Linked ID data

It is highly recommended that the Linked ID data be loaded into a database application for use. It is not possible to illustrate within this guide the loading of these datasets into every database application available. Therefore, users should familiarise themselves with the operations of both loading the data and then indexing it so that performance can be improved.

For the purposes of this guide, an example will be shown illustrating the loading of the Road_TOID_Street_USRN Linked ID CSV file into a PostgreSQL (Open Source) database.

3.2 PostgreSQL Database

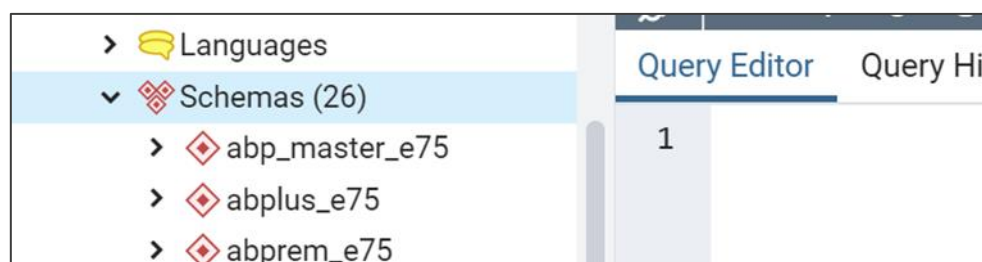
3.2.1 Preparation

In this example version 11 of PostgreSQL is being used. It is expected that a later release (version 12) should also behave in a similar manner. Pg_Admin 4 or any other free database manager can be used to carry out the procedures. Pg_Admin 4 is the database manager front-end application installed with PostgreSQL, and PostGIS for tables with geometry, will be used to illustrate the procedures.

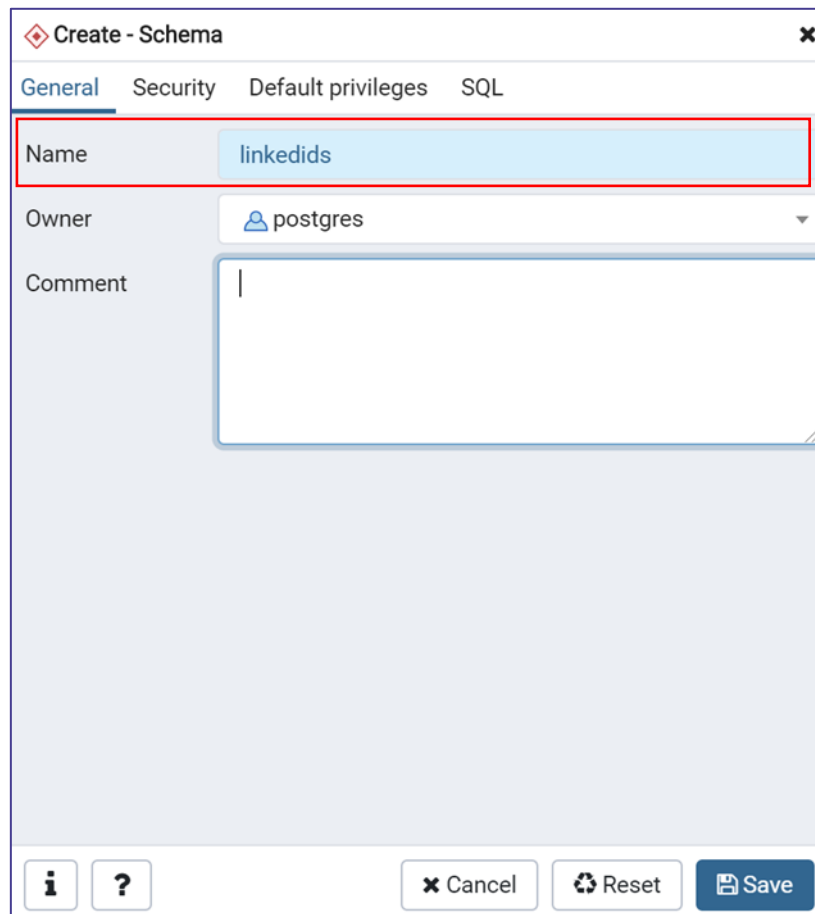
3.2.2 Create Schema

It is assumed that the end user has already created a database within Postgres to hold other data. The other data held by the user will be held in different schemas. It is recommended that a new schema be created to hold the Linked ID data, especially if the user intends to use most or all the Linked IDs available.

1. In Pg_Admin right-click on Schemas under the database listing



2. Select 'Create Schema'.



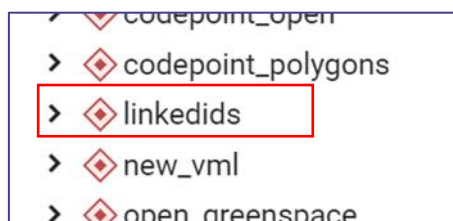
The screenshot shows a 'Create - Schema' dialog box with the following details:

- Title:** Create - Schema
- Tabs:** General (selected), Security, Default privileges, SQL
- Name:** linkedids (highlighted with a red box)
- Owner:** postgres
- Comment:** (empty text area)
- Buttons:** i, ?, Cancel, Reset, Save

3. Give the schema a name in the Name box.

4. Click 'Save'

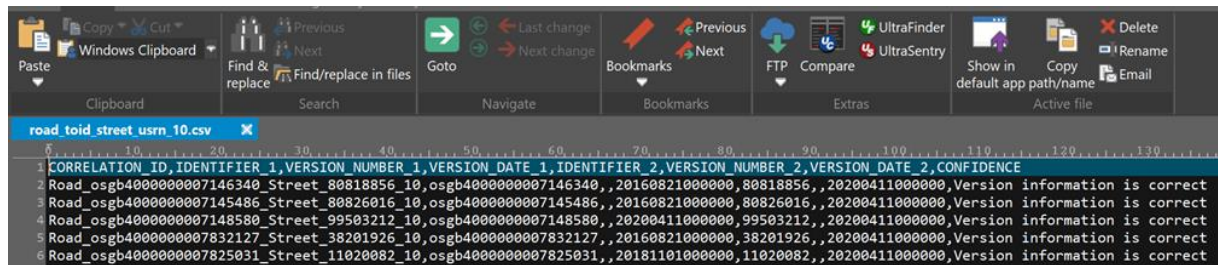
The new schema now appears in the listing of schemas within the database.



3.2.3 Create Table

The next procedure will show you how to create a table within the schema to hold the Linked ID data for the individual Linked ID that the user is seeking to load. The table will need to have columns created corresponding to the fields within the Linked ID .CSV file. The user will either need to use an advanced text editor (e.g. UltraEdit) to view the CSV data, or refer to the technical specification document to see what fields are contained within the Linked ID data, and what are the data types within the fields. In this example, we will define all the fields within the Linked ID data as text fields.

- Using UltraEdit Text editor, which can open very large CSV files, we can view the column headings within the CSV file. The example below shows the opening of the Road_Toid_street_USRN_10.CSV file, which is the example Linked ID file we will be using to illustrate the loading procedure in this guide.

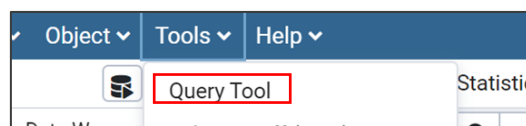


- We have identified the column headers and can now use a script like the one below which will carry out the table creation operation. This includes both dropping any pre-existing table of the same name, and then creating a table to hold the data.

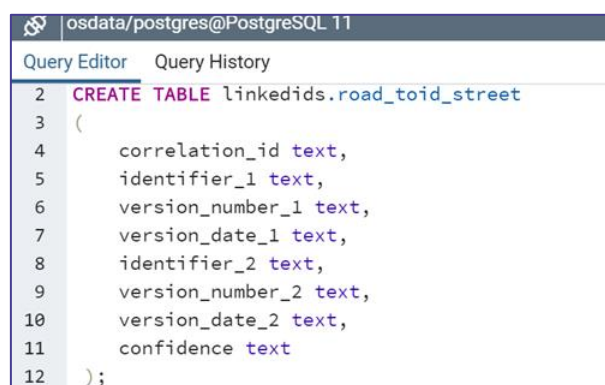
Please note this code will need to be modified to account for the variables in the user's own environment, but the column names and data types will remain the same.

```
DROP TABLE IF EXISTS linkedids.road_toid_street;
CREATE TABLE linkedids.road_toid_street
(
    correlation_id text,
    identifier_1 text,
    version_number_1 text,
    version_date_1 text,
    identifier_2 text,
    version_number_2 text,
    version_date_2 text,
    confidence text
);
```

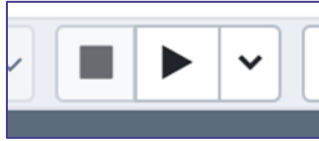
- In Pg_Admin, under 'Tools', select 'Query Tool' to open a SQL query window



- Copy and paste the code from a Text file (recommended) into the SQL window.



5. When satisfied, click the 'run' button to execute the query.



The query should now run, creating the table, and loading the data into the created table. A success message should appear when the query has run.

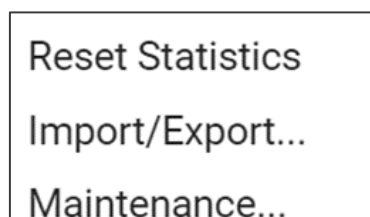
3.2.4 Loading the data

In Pg_Admin, under the LinkedIDs schema a new table should now be visible.



In this example the new table is called road_toid_street. We will now use the import data function within Pg_Admin_4 to load the data into the created table.

1. Right-Click on the road_toid_street table just created and select 'import/export'.



Import/Export data - table 'road_toid_street'

Options Columns

Import/Export **Import**

File Info

Filename C:\Data\Temp\road_toid_street_usrn_10.csv

Format csv

Encoding UTF8

Miscellaneous

OID No

Header **Yes**

Delimiter ,

Specifies the character that separates columns within each row (line) of the file. The default is a tab character in text format, a comma in CSV format. This must be a single one-byte character. This option is not allowed when using binary format.

Cancel OK

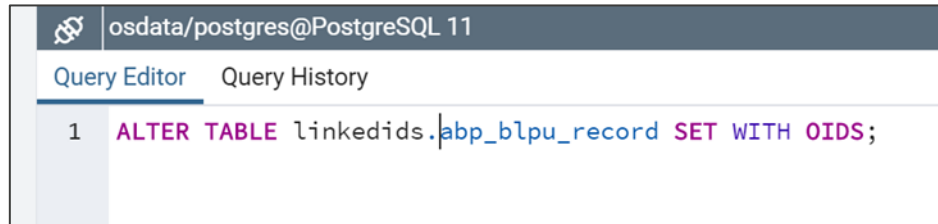
2. In the window that appears, select the import/export to 'import'.
3. Select the filename of the CSV file which is being imported.
4. Set the encoding to UTF-8.
5. Set the header box to 'Yes' because we know that the CSV file contains header information.
6. Select the delimiter to comma.
7. Click 'OK' when finished. The data will now load into the created table.

3.2.5 Viewing/Indexing the data

1. Right-clicking on the road_toid_street table and viewing the top 100 rows shows a result similar to that shown below. This indicates that the Linked ID data for the Road_Toid_street_USRN_10.CSV has been successfully loaded into the database.

	correlation_id	identifier_1	version_number	version_date_1	identifier_2	version_number_2	version_date_2	confidence
	text	text	text	bigint	text	text	bigint	text
1	Road_osgb4000000...	osgb400000000...	[null]	20160821000000	80818856	[null]	20200411000000	Version informa...
2	Road_osgb4000000...	osgb400000000...	[null]	20160821000000	80826016	[null]	20200411000000	Version informa...
3	Road_osgb4000000...	osgb400000000...	[null]	20200411000000	99503212	[null]	20200411000000	Version informa...
4	Road_osgb4000000...	osgb400000000...	[null]	20160821000000	38201926	[null]	20200411000000	Version informa...
5	Road_osgb4000000...	osgb400000000...	[null]	20181101000000	11020082	[null]	20200411000000	Version informa...
6	Road_osgb4000000...	osgb400000000...	[null]	20160821000000	11706260	[null]	20200411000000	Version informa...
7	Road_osgb4000000...	osgb400000000...	[null]	20160821000000	8200575	[null]	20200411000000	Version informa...
8	Road_osgb4000000...	osgb400000000...	[null]	20160821000000	28812783	[null]	20200411000000	Version informa...

- The data should now be indexed, which can be done by adding an index to one of the fields in the data or by adding OIDs as unique IDs to the data. This can be achieved by typing in a code string like the one shown below into the query window and then running the query.



```
1 ALTER TABLE linkedids.add_bp_record SET WITH OIDS;
```

Adding this new field will enable any joining and querying to be achieved much more quickly. Looking at the data table now, we can see that PostgreSQL has added a unique OID field to the data.

	oid	correlation_id	identifier_1	version_number	version_date_1	identifier_2	version_number_2	version_date_2	confidence
	oid	text	text	text	bigint	text	text	bigint	text
1	10012...	Road_osgb400000...	osgb40000000...	[null]	20160821000000	80818856	[null]	20200411000000	Version informa...
2	10012...	Road_osgb400000...	osgb40000000...	[null]	20160821000000	80826016	[null]	20200411000000	Version informa...
3	10012...	Road_osgb400000...	osgb40000000...	[null]	20200411000000	99503212	[null]	20200411000000	Version informa...
4	10012...	Road_osgb400000...	osgb40000000...	[null]	20160821000000	38201926	[null]	20200411000000	Version informa...
5	10012...	Road_osgb400000...	osgb40000000...	[null]	20181101000000	11020082	[null]	20200411000000	Version informa...
6	10012...	Road_osgb400000...	osgb40000000...	[null]	20160821000000	11706260	[null]	20200411000000	Version informa...
7	10012...	Road_osgb400000...	osgb40000000...	[null]	20160821000000	8200575	[null]	20200411000000	Version informa...
8	10012...	Road_osgb400000...	osgb40000000...	[null]	20160821000000	28812783	[null]	20200411000000	Version informa...

Other database applications will have similar methods for the creation and loading of data tables. The data is now ready for use and table-joins can now be made with the end user's own data held either within a database or outside the database in a GI application

