# OD Matrix specification format

## Standardization

When transferring data related to movements, this standard can be used to specify the data. The specification is simple and normally easy to generate. The ODM set consists out of 3 parts:

1. Description file
2. Geography specification
3. The OD values file(s)

These 3 parts can be zipped together in one file. The ‘.zip’ extension must be replaces by ‘.odz’.

The ‘base’ filename will be used in all parts, and must not contain a dash, nor other invalid path characters.

## Description file

The description file is a JSON file, explained here:

*{   
 “****unit****”: “unit of the values, like ‘trips’, ‘duration’. This must be repeated in the OD-values-file”,  
 “****geography\_id****” : “the field in the geography specification that identifies the geography”,  
 “****aggregation\_period****”: { “****start****”: “start date-time in* [*RFC 3339, section 5.6*](https://xml2rfc.tools.ietf.org/public/rfc/html/rfc3339.html#rfc.section.5.6)*(= ISO 8601) )”,  
 “****end****”: “end date-time in* [*RFC 3339, section 5.6*](https://xml2rfc.tools.ietf.org/public/rfc/html/rfc3339.html#rfc.section.5.6)*(= ISO 8601) )” },  
 “****generation\_date****”: “generation date-time in* [*RFC 3339, section 5.6*](https://xml2rfc.tools.ietf.org/public/rfc/html/rfc3339.html#rfc.section.5.6)*(= ISO 8601) )”,  
 “****trip\_value\_at****”: “start\_of\_trip|end\_of\_trip|part\_of\_trip”,  
 “****value\_files****”: [ {  
 “****file\_name****”: “the name of the value file”,  
 “****purpose****”: [“the purpose for the movement, like ‘school’, ‘work’, ‘shopping’, see   
 appendix A - purposes”],  
 “****mode****”: [“the mode of transport, see appendix A - modes”],  
 “****aggregation\_function****”: [“the unit of the values, like count, max, min, avg, mean,   
 percentile85, etc. , see appendix A – aggregation functions”],* ***“aggregation\_day\_bucket”****: “aggregation level, like quarterly, monthly, weekly etc. ,   
 see appendix A - buckets”,   
 “****date\_bucket****”: [number representing the year, quarter, month, week etc. (>=1)],* ***“aggregation\_time\_bucket”****: “aggregation level, like hourly, day\_parts, quarters, etc.   
 see appendix A - buckets”   
 “****time\_bucket****”: [number representing the hour, quarter index or day\_part (>=0)]  
 } ],  
 “****daypart\_definitition****” : [ {  
 “****name****”: “name of the daypart”,   
 “****time\_bucket\_index****”: index used in the ‘time\_bucket’ for this daypart  
 “****start****”: “start partial-time in* [*RFC 3339, section 5.6*](https://xml2rfc.tools.ietf.org/public/rfc/html/rfc3339.html#rfc.section.5.6)*(= ISO 8601) )”,  
 “****end****”: “end partial-time in* [*RFC 3339, section 5.6*](https://xml2rfc.tools.ietf.org/public/rfc/html/rfc3339.html#rfc.section.5.6)*(= ISO 8601) )” } ]  
}*

The file ends with ‘.odd’ (= Origin Destination Definition ).

## Geography specification

For this part ESRI shapefiles (https://www.esri.com/content/dam/esrisites/sitecore-archive/Files/Pdfs/library/whitepapers/pdfs/shapefile.pdf) can be used or a GeoJSON file (https://geojson.org/). In the files there must be a property as named in the description file and a valid geometry. The projection to use is WGS84.

ESRI files should comply with the specification, but the name of the files must be the same as the description file (.shp, .shx, .dbf).

If Geojson is used, the extension of the file must be ‘.geojson’.

## OD Values

The actual data can be stored in a semicolon separated csv file in this format:

Cell 1,1: this cell must contain (dash-delimited) a) the UNIT, b) the aggregation function c) value of the date bucket and d) the value of the time bucket.

Cell 1,2 … 1,n: the values of the ids (see ‘geography\_id’) in the geography specification

Cell 2,1.. n,1: the values of the ids (see ‘geography\_id’) in the geography specification

Other cells: the values of the relation, where the it must be read as follows: the geography identified by the first cell in the row must be considered as ‘Origin’ and the value in the header of that cell must be considered as the ‘Destination’.

The file must have the extension ‘.odv’ (= Origin Destination Values), and should have the same name as the other files, optionally extended. For convenience reasons, the dimensions could be included in the filename: like *example\_odmatrix-ALL-ALL-COUNT-ALL-ALL.odv* (dash-separated). That means the dash must not be in the initial filename to make it easily processable!

The order should of dimensions should be:

1. Purpose
2. Mode
3. Aggregation function
4. Date bucket, and if not ‘ALL’, it must be extended with # and the bucket number. For instance, exchanging information about February: MONTH#2
5. Time bucket, and if not ‘ALL’, it must be extended with # and the bucket number. For instance, exchanging information about the 3rd hour of the day: HOUR#3

Advanced usage – combined values

If you want to exchange multiple values per OD-pair, these values can be combined into one cell, separated by pipes. Like:

DURATION-ALL-ALL-SUM|AVG|MIN|MAX-ALL-ALL;324AC234;349AB347

324AC234;34298|34|2|57;34234|67|12|92

349AB347;94344|55|34|74;93422|48|30|70

This construct can be applied at only 1 of the dimensions and has to be specified in cell 0,0. The order of the values of the aggregation functions must correspond with the order of the aggregation function in cell 0,0 and in the “aggregation\_function” field of the ODD file.

The in the file name the pipes must be replaced by a underscore.

## Simple example

### Example\_odmatrix.odd

{

“unit”: “TRIPS”,  
 “geography\_id” : “id”,  
 “aggregation\_period”: { “start”: “2017-06-01T00:00:00.000”,  
 “end”: “2017-07-01T00:00:00.000” },  
 “generation\_date”: “2017-07-01T03:00:00.000Z”,  
 “value\_files”: [  
 {

“file\_name”: “example\_odmatrix.odv”,  
 “purpose”: [“ALL”],  
 “mode”: [“ALL”],  
 “aggregation\_function”: [“COUNT”],  
 “aggregation\_date\_bucket”: “ALL”,  
 “aggregation\_time\_bucket”: “ALL”  
 } ]  
}

### Example\_odmatrix.geojson

{ "type": "FeatureCollection",

"features": [

{ "type": "Feature",

"geometry": { "type": "Polygon", "coordinates":[

[[100.0, 0.0], [101.0, 0.0], [101.0, 1.0],[100.0, 1.0], [100.0, 0.0]]]},

"properties": { "id": "324AC234", "name": "First place" }

},

{ "type": "Feature",

"geometry": { "type": "Polygon", "coordinates":[

[[200.0, 0.0], [201.0, 0.0], [201.0, 2.0],[200.0, 2.0], [200.0, 0.0]]]},

"properties": { "id": "349AB347", "name": "Second place" }

}

]}

### Example\_odmatrix.odv

TRIPS-ALL-ALL-COUNT-ALL-ALL;324AC234;349AB347

324AC234;2;342

349AB347;94;9

## Notes

It is not mandatory that every geometry in the ‘geometry specification’ is enlisted in the .odv file.

## Example multiple value files

### Multiple.odd

{

“unit”: “TRIPS”,  
 “geography\_id” : “id”,  
 “aggregation\_period”: { “start”: “2017-06-01T00:00:00.000”,  
 “end”: “2017-07-01T00:00:00.000” },  
 “generation\_date”: “2017-07-01T03:00:00.000Z”,  
 “value\_files”: [  
 {

“file\_name”: “multiple-ALL-ALL-COUNT-MONTH#6-ALL.odv”,  
 “purpose”: [“ALL”],  
 “mode”: [“ALL”],  
 “aggregation\_function”: [“COUNT”],  
 “aggregation\_date\_bucket”: “MONTH”,  
 “date\_bucket”: [6],  
 “aggregation\_time\_bucket”: “ALL”  
 },  
 {

“file\_name”: “multiple-ALL-ALL-COUNT-MONTH#7-ALL.odv”,  
 “purpose”: [“ALL”],  
 “mode”: [“ALL”],  
 “aggregation\_function”: [“COUNT”],  
 “aggregation\_date\_bucket”: “MONTH”,  
 “date\_bucket”: [7],  
 “aggregation\_time\_bucket”: “ALL”  
 } ]  
}

## Example combined values

### Combined.odd

{

“unit”: “TRIPS”,  
 “geography\_id” : “id”,  
 “aggregation\_period”: { “start”: “2017-06-01T00:00:00.000”,  
 “end”: “2017-07-01T00:00:00.000” },  
 “generation\_date”: “2017-07-01T03:00:00.000Z”,  
 “value\_files”: [  
 {

“file\_name”: “combined-ALL-BIKE|MOPED-COUNT-ALL-DAY\_PART#1.odv”,  
 “purpose”: [“ALL”],  
 “mode”: [“BIKE”, “MOPED”],  
 “aggregation\_function”: [“COUNT”],  
 “aggregation\_date\_bucket”: “ALL”,  
 “aggregation\_time\_bucket”: “DAY\_PART”,  
 “time\_bucket”: [1]

},

{

“file\_name”: “combined-ALL-BIKE\_MOPED-COUNT-ALL-DAY\_PART#2.odv”,  
 “purpose”: “ALL”,  
 “mode”: [“BIKE”, “MOPED”],  
 “aggregation\_function”: [“COUNT”],  
 “aggregation\_date\_bucket”: “ALL”,  
 “aggregation\_time\_bucket”: “DAY\_PART”,  
 “time\_bucket”: [2]

} ],

“daypart\_definitition” : [ {  
 “name”: “morning rush hour”,   
 “time\_bucket\_index”: 1

“start”: “07:00:00”,  
 “end”: “09:00:00” },  
 {  
 “name”: “evening rush hour”,   
 “time\_bucket\_index”: 2

“start”: “16:30:00”,  
 “end”: “18:30:00” } ]  
}

### Combined-ALL-BIKE\_MOPED-COUNT-ALL-DAY\_PART#1.odv

TRIPS-ALL-BIKE|MOPED-COUNT-ALL-DAY\_PART#1;324AC234;349AB347

324AC234;2;342

349AB347;94;9

# Appendix A

Each of the dimensions is a proposed list of possibilities. We don’t think the possibilities are limited. That’s why we make a proposal for each list, but when it is not possible to find a suitable value in the list, you are allowed to specify other values. Be aware of the fact that it makes it hard or impossible to use the data processable by consuming parties.

## Purposes

enum: [ALL,WORK, SCHOOL, SHOPPING, HOME, MULTIPLE, LEASURE, VACATION, FAMILY\_VISIT,  
OTHER]

## Modes

The modes for the OD matrix are derived from NeTEx, and extended with extra modes for micro mobility, as stated in the specification of the TOMP-API (https://github.com/TOMP-WG/TOMP-API/blob/master/TOMP-API.yaml)

enum: [AIR,BUS,TROLLEYBUS,TRAM,COACH,RAIL,INTERCITYRAIL,URBANRAIL,METRO,WATER,CABLEWAY,

FUNICULAR,TAXI,SELFDRIVE,FOOT,BICYCLE,MOTORCYCLE,CAR,SHUTTLE,OTHER,PARKING,MOPED,   
STEP,ALL]

## Aggregation functions

enum: [COUNT, MAX, MIN, AVG, MEAN, PERCENTILE85, PERCENTILE90, PERCENTILE95]

## Buckets

In the aggregation period, you can aggregate per ‘bucket’. This means, it can be aggregated per year, half a year, quarter, month, week, day, weekday/weekend day/working day/holiday, per hour (0-23), per quarter of an hour (0-95), per day part (‘morning rush hour’, ‘evening rush hour’, ‘evening hours’, ‘nightly hours’, and ‘rest of the day’). The exact times for the last type have to be specified in the odd file.

In case there is no aggregation at day or time level, ‘ALL’ should be specified.

### Date bucket:

enum: [YEAR, HALF\_YEAR, QUARTER, MONTH, WEEK, DAY, DAY\_PER\_TYPE, ALL]

### Time bucket:

enum: [HOUR, QUARTER\_OF\_HOUR, DAY\_PART, ALL]