



# **Digital Durstlöcher Database**

## **- Technical Manual -**

Version 2020.1

## Content

1	Digital Durstlöcher Database .....	3
1.1	Introduction .....	3
1.2	Purpose .....	3
1.3	Basic Metrics .....	3
1.4	Further Information .....	3
2	Recorded Data .....	3
2.1	Pictures .....	3
2.2	Data File .....	4
2.3	Collected Data .....	4
2.3.1	ID .....	4
2.3.2	File .....	4
2.3.3	GPSLongitude .....	4
2.3.4	GPSLatitude .....	4
2.3.5	Type .....	4
2.3.6	City .....	5
2.3.7	Place .....	5
2.3.8	Quality .....	6
2.3.9	Design .....	6
2.4	Future Data .....	6
2.4.1	Straw .....	6
2.4.2	Cohab .....	7
3	R Scripts .....	7
3.1	DDDmap.R .....	7
3.1.1	readLoescherTable() .....	7
3.1.2	createLoescherTable() .....	7
3.1.3	plotMap() .....	7
3.1.4	plotMainArea() .....	8
3.1.5	addPointsCol() .....	8
3.1.6	addPointsIf() .....	8
3.1.7	addPointsAll() .....	8
3.1.8	addPointsDate() .....	9
3.1.9	getMonthData() .....	9
3.2	DDDpeach.R .....	9
3.2.1	peachQuot() .....	9
3.2.2	yearTotal() .....	9
3.2.3	compQuot() .....	9

# 1 Digital Durstlöscher Database

## 1.1 Introduction

The Digital Durstlöscher Database (DDD) is an annotated collection of photos of wasted boxes of the *Durstlöscher* brand soft drink produced in Germany by the companies *riha WeserGold GmbH & Co. KG* and *Lekkerland Deutschland GmbH & Co. KG* (QuickVit brand). Most Photos were taken in the city of Duisburg and the surrounding cities. Together with the photos, metadata for each picture is collected in a *.csv file*.

The DDD was started in autumn of 2018. Over the time the format of the database changed, as new information was added as metadata.

This manual describes the specification of the 2020.1 version.

## 1.2 Purpose

The DDD is supposed to enable people to do research on occurrence and interaction of *Durstlöscher* especially in an urban environment. For this raw photos and as much metadata as possible will be stored in the DDD.

For easy analysis of the data, *R* scripts are being developed and provided.

A lot of useful research could already be conducted. Results are presented on the projects homepage and Instagram channel.

## 1.3 Basic Metrics

End of 2020 the DDD consisted from XXXX individual *Durstlöscher*s. XXXX Of those where taken before 2020 and still follow the old annotation.

## 1.4 Further Information

Homepage: <https://www.moronaut.de>

Instagram: <https://instagram.com/grottenbacher>

GitHub: <https://github.com/moronaut/DigitalDurstloescherDatabase>

# 2 Recorded Data

## 2.1 Pictures

Pictures taken of *Durstlöscher*s are the foundation of DDD. They are taken with GPS Tracking, to enable geological surveys regarding *Durstlöscher*. All pictures are saved in space saving format with at least 1 megapixel resolution.

Geodata is extracted from the pictures using a *R* script (available on GitHub).

Before 2020 the picture information was not stored and is not available.

## 2.2 Data File

The extracted geodata gets stored in a *.csv file* and gets enriched with additional information about every single *Durstlöscher*. Each *Durstlöscher* in the database will have its own ID, which is a running number. Multiple *Durstlöscher*s in one picture will have the same geodata, but individual annotation and ID.

## 2.3 Collected Data

In the *.csv file*, different characteristics of each *Durstlöscher* is saved. This section is supposed to help with the interpretation of the stored data.

### 2.3.1 ID

Running number to identify each individual *Durstlöscher*.

### 2.3.2 File

The file name of the original picture that the *Durstlöscher* was sampled from. Before 2020 these are pseudo names, generated from the date and the type of *Durstlöscher*.

### 2.3.3 GPSLongitude

Longitude of the original picture, taken from GPS.

### 2.3.4 GPSLatitude

Latitude of the original picture, taken from GPS.

### 2.3.5 Type

Type (Taste) of *Durstlöscher*. Same Type in database can be a different type, depending on Design. Some Types only exist with specific Design.

It is worth mentioning, that more types of *Durstlöscher* exist, that are not represented in DDD.

Class	Design A/B	Design 0/C
AOZ	Apfel-Orange-Zitrone	Apfel-Orange-Zitrone
Blau	Blaubeer-Vanille (only B)	
Cola	Cola	Cola
Energy		Eistee Energy
ErdRhab	Erdbeer-Rhababer	
Frütee	Früchtetee (only A)	
Granat	Granatapfel-Zitrone	Granatapfel-Zitrone
Him	Erdbeer-Himbeer	Himbeer
Kiba	A: Banane-Sauerkirsch, B: Kirsch-Banane	Banane-Sauerkirsche

<i>Kizi</i>	Sauerkirsch-Zitrone	Sauerkirsche-Zitrone
<i>Mango</i>		Eistee Mango
<i>Marsh</i>		Blueberry-Marshmellow (only C)
<i>Mate</i>		Mate
<i>Multi</i>	Multivitamin 12 Frucht	Multivitamin 12-Frucht
<i>Ora</i>	Orange	Orange
<i>Pf</i>	Eistee Pfirsich	Eistee Pfirsich
<i>Pf-T</i>	Pfirsich mit Traubenzucker (only B)	
<i>Wald</i>	Waldmeister	Waldmeister
<i>Wasser</i>	Wassermelone	Wassermelone
<i>Zit</i>	Eistee Zitrone	Eistee Zitrone

### 2.3.6 City

City that the *Durstlöscher* was found in.

Class	City
<i>D</i>	City of Düsseldorf
<i>Do</i>	City of Dortmund
<i>Du</i>	City of Duisburg
<i>E</i>	City of Essen
<i>Gre</i>	Grevenbroich, County of Rhein-Kreis Neuss
<i>Krefeld</i>	City of Krefeld
<i>Me</i>	Hilden, County of Mettmann
<i>Mh</i>	City of Mülheim
<i>Mo</i>	Moers, County of Wesel
<i>Ob</i>	City of Oberhausen
<i>W</i>	City of Wuppertal

### 2.3.7 Place

The environment, that the *Durstlöscher* was found in. This is a generalized attribute. Multiple environments might be summarized in one class.

Class	Place(s)
<i>Dirt</i>	Any type of dirt ground, that is not in other class: Earth, gravel, construction side, etc.
<i>Green</i>	Any type of park, park benches, plant bed, planter, tree, etc.
<i>House</i>	Inside or next to a building, on a stair, wall, fence, any type of built structure that does not fall into another class.
<i>Pave</i>	Pavement, footpath, bike lane
<i>Street</i>	Street
<i>Train</i>	Any train related surrounding, inside a train, on a platform, on the tracks, etc.
<i>Trash</i>	Inside or next to a trash can

<i>Utilitybox</i>	On or next to a utility box, phone switch box, park meter, etc.
<i>Vehicle</i>	In or on a vehicle. NOT next to a vehicle (this would be <i>Pave</i> or <i>Street</i> ).
<i>Water</i>	In or under water.

### 2.3.8 Quality

A scale from -2 to 2, without zero, stating the grade of degradation of the Durstlöscher. This attribute can be a bit subjective, because no real rule was implemented on how to grade the quality. Only a rough outline is given.

Class	Quality
2	Box is completely intact. No sign of deterioration.
1	Slight deformation of box.
-1	Box is compressed, deformed or teared.
-2	Box is completely flat, deteriorated, or destroyed

### 2.3.9 Design

The design of the *Durstlöscher* box.

Class	Design
0	Early design of <i>riha</i> Durstlöscher (until ca early 2018)
A	Early design of <i>QuickVit</i> Durstlöscher (until ca mid-2019)
B	Late design of <i>QuickVit</i> Durstlöscher (from mid-2017)
C	Late design of <i>riha</i> Durstlöscher (from early 2019)
X	Other Durstlöscher, different manufacturer. Only 2 ever recorded.

## 2.4 Future Data

In the next version(s) of DDD, additional attributes will be collected. Features listed below will be added to the next version of DDD.

### 2.4.1 Straw

Type of straw belonging to the box.

Class	Straw
B	Bent straw
N	None
S	Straight straw

X	Straw next to the box
---	-----------------------

### 2.4.2 Cohab

Cohabitation. Other Trash surrounding the *Durstlöscher* box. These will be saved as a string, as multiple objects can cohabitate one *Durstlöscher*.

Class	Cohab
B	Booze
C	Cigarette
D	Drinks (not capri sun)
F	Food Waste (packaging and actual food)
G	General Waste
L	General Waste, Plastic
N	Napkins and toilet paper
P	Paper waste, Newspaper, Magazine
S	Capri Sun
X	None

## 3 R Scripts

Together with the DDD, a R scripts are in development, which allow easy extraction of data from the database.

### 3.1 DDDmap.R

The R script “*DDDmap.R*” contains several functions for geo data analysis of DDD data.

#### 3.1.1 readLoescherTable()

Function: Reads the DDD .csv file into a dataframe. Needed for all other functions of *DDDmap.R*.

Input: - csv = csv file, containing DDD data

Output: dataframe containing DDD data

#### 3.1.2 createLoescherTable()

Function: Reads *exif* data of photos to create a DDD .csv file for new photos of *Durstlöscher* boxes.

Input: - folder = folder containing

Output: .csv file in the input folder, containing DDD data for new photos

#### 3.1.3 plotMap()

Function: Plots map of the DDD area of operation. Paths to shapefiles need to given in source code.

Shapefiles are provided with DDD.

```
map <- outlines of area of operation
```

```
int <- internal boundaries
```

```
riv <- rivers
```

Output: Plot of area of operation.

### **3.1.4 plotMainArea()**

Function: Creates an empty rectangular plot of the main area of investigation. Coordinates are hard coded.

Output: Plot of main area of investigation.

### **3.1.5 addPointsCol()**

Function: Adds color coded points to the plot. Category for coloring can be chosen. Has to be executed after plotMap() or plotMainArea().

Input: - data = dataframe containing DDD data.

Input: - cat = column of DDD dataframe to color code by.

Output: Points on plot.

### **3.1.6 addPointsIf()**

Function: add only points with particular attribute. Has to be executed after plotMap() or plotMainArea().

Input: - data = dataframe containing DDD data

Input: - cat = column of dataframe to attribute.

Input: - selector = attribute to filter by.

Input: - fill = color of points to add.

Output: Points on plot.

### **3.1.7 addPointsAll()**

Function: add all points to a map. Has to be executed after plotMap() or plotMainArea().

Input: - data = dataframe containing DDD data

Input: - fill = color of points to add.

Output: Points on plot.



### 3.1.8 addPointsDate()

Function: Adds points with specific date to the plot. Adapted from *addPointsIf()*. Has to be executed after *plotMap()* or *plotMainArea()*.

Input: - data = dataframe containing DDD data.

Input: - date = string with date information.

Input: - fill = color of points to add.

Output: Points on plot.

### 3.1.9 getMonthData()

Function: Prints a summary table of collected DDD data of a specific month (year).

Input: - data = dataframe containing DDD data.

Input: - date = string with date information.

Output: non-machine readable text with summary information.

## 3.2 DDDpeach.R

This script contains a few functions specifically for analysis regarding the *pf* type of *Durstlöschers*. It was used for the 2020 research poster, presented on the <https://moronaut.de> homepage.

### 3.2.1 peachQuot()

Function: Outputs a plot of the annual ratio of *Pf Durstlöschers* to overall *Durstlöschers* as well as the change to the previous year.

Input: Dataframe containing DDD data. Hardcoded

Output: Plot

### 3.2.2 yearTotal()

Function: Outputs a bar plot of the overall annual *Durstlöschers*, as well as the change to the previous year.

Input: Dataframe containing DDD data. Hardcoded

Output: Plot

### 3.2.3 compQuot()

Function: outputs a bar graph of total *Durstlöschers* per manufacturer and their part of *Pf Durstlöschers*.

Input: Dataframe containing DDD data. Hardcoded

Output: Plot