



Working Paper

## Creation of vehicle diaries using C++

**Author(s):**

Singhi, Pranay

**Publication Date:**

2001

**Permanent Link:**

<https://doi.org/10.3929/ethz-a-004231917> →

**Rights / License:**

[In Copyright - Non-Commercial Use Permitted](#) →

This page was generated automatically upon download from the [ETH Zurich Research Collection](#). For more information please consult the [Terms of use](#).

---

tripno	studyc	cityc	hnnr	pnr	vechno	muser	nop	dorp	tnr	tdep	tarr	type
1	2	1	76	1	1	1	1	1	2	48600	49200	5
2	2	1	76	1	1	1	1	1	3	50400	51000	5
3	2	1	76	1	1	1	1	1	6	80400	81600	5
4	2	1	76	1	1	1	1	2	1	25800	27300	5
5	2	1	76	1	1	1	1	2	2	27600	29100	5
6	2	1	76	1	1	1	1	9	3	66300	68400	5
7	2	1	76	1	1	1	1	18	4	79200	80400	5
8	2	1	76	1	1	1	1	19	1	36000	36600	5
9	2	1	76	1	1	1	1	19	2	37200	38100	5
10	2	1	76	1	1	1	1	19	3	46800	48000	5
11	2	1	76	1	1	1	1	19	4	61200	62400	5
12	2	1	76	1	1	1	1	22	5	84600	85800	5
13	2	1	76	1	1	1	1	24	3	64200	65700	5
14	2	1	76	1	1	1	1	26	5	71100	72000	5
15	2	1	76	1	1	1	1	28	4	61200	62700	5

---

## Creation of Vehicle Diaries using C++

Pranay Singhi

Arbeitsbericht  
Verkehrs- und Raumplanung

90  
October 2001

## Creation of Vehicle Diaries using C++

Pranay Singhi  
IVT  
ETH  
CH-8093 Zürich

Telefon: +41-1-633 79 47  
Telefax: +41-1-633 10 57  
eMail: [n8004022@ccs.iitb.ernet.in](mailto:n8004022@ccs.iitb.ernet.in), [axhausen@ivt.baug.ethz.ch](mailto:axhausen@ivt.baug.ethz.ch)

October 2001

### Abstract

Models that consider the environmental impacts of vehicular traffic require the use of trip data of vehicles. The trip data for the vehicle is stored in the form of vehicle diaries, which provide the information regarding the time of use of the vehicle, occupancy and main user of the vehicle. The German research project Mobidrive, funded by the Federal Ministry of Education and Research, implemented a six-week continuous travel diary, which now yields a unique database for creation of vehicle diaries of the household that took part in the survey. The paper describes the C++ program, which was used for the creation of vehicle diaries within the database.

### Keywords

Vehicle Diaries, C++ programming, Mobidrive, ETH Zürich – Institut für Verkehrsplanung und Transporttechnik, Strassen- und Eisenbahnbau (IVT)

### Preferred citation

Singhi, P. (2001) Analysis of joint trips using C++ in Mobidrive, *Arbeitsbericht Verkehrs- und Raumplanung*, **90**, Institut für Verkehrsplanung und Transporttechnik, Strassen- und Eisenbahnbau, ETH, Zürich.

## 1. Introduction

The German research project MobiDrive, funded by the German ministry of Research and Education, is an ambitious study to observe and analyse the structure as well as the determinants of temporal aspects in individual travel behaviour. The analysis is mainly based on a six-week travel survey, which was conducted to observe the longitudinal phenomena of behaviour of travel, such as rhythms in the behaviour of the respondents (Axhausen, Zimmermann, Schönfelder, Rindsfuser and Haupt, 2000; PTV AG, Fell, Schönfelder and Axhausen, 2000; Schlich, König, Aschwanden, Kaufmann and Axhausen, 2000). This analysis can be further extended to observe the travel pattern of vehicles from the travel survey data.

The survey was performed in the German cities of Karlsruhe and Halle both with about 270,000 inhabitants, in the fall of 1999. A total of 317 persons over 6 years of age in 139 households participated in the main study (Axhausen *et al.*, 2000).

The working paper describes a method to create vehicle diaries from the travel data survey. The vehicle diaries give the information regarding the time for which the vehicle was in use, the number of people occupying the vehicle, the main user of the vehicle and other trip characteristics. A program written in C++ programming language creates the vehicle diaries. The paper describes the algorithm of the program, the input and the output files, which are used and the variables employed in the program.

## 2. C++ Program Description

The generation of the vehicles diaries from the Mobidrive data was done using C++ as a programming tool (Lafore, 1991). The joint trip file (Singhi, 2001) and the vsum file of the Mobidrive data was used as the primary source reference for the analysis.

### 2.1 Input Description

The input for the program has two sources. The information for the trips is taken from the joint trip file and the user of the program enters the information for the vehicle.

### 2.1.1 Input for the trip

The input file is of .txt format with the following variables (Singhi, 2001):

1. Studyc: the study code for Main study or the Pre test study.
2. Cityc: the city code of Halle or Karlsruhe.
3. Hhnr: the household number of the person taking the the trip.
4. Pnr: the person number in the household.
5. Jtrip: indicates whether the trip is a joint trip or not.
6. Nop: the number of persons in the trip.
7. Tdep: the time of departure of the trip taken by the person.
8. Tarr: the time of arrival of the trip taken by the person.
9. Tmmb: the main mode used in the trip by speed.
10. Dorp: the day of reporting period of the person taking the trip.
11. Tnr: the trip number of the trip.

### 2.1.2 Input for the vehicle

1. Num: the number of the vehicle.
2. Type: the type of the vehicle in use (car, bike, cycle or any other vehicle).
3. mu: the main user of the vehicle
4. u1: the second user in preference of the vehicle after the main user.
5. u2: the third user in preference of the vehicle after main user and u1.

## 2.2 Algorithm of the program

The algorithm of the program proceeds as follows:

- The program reads the input data from the input file as described before. The program creates two classes by the name info and info1. The class info has the information about a particular trip from the joint trip file which includes the study code, city code, household number, person number, day of reporting, departure and arrival times of the trip, trip main mode by speed, trip number, whether it is a joint trip number. The info

class also has objects for indicating the vehicle in use, the main user in the vehicle and the type of the vehicle. The class info1 has the information about the vehicle, which is given by the user of the program. The class info1 has the information about the vehicle number in the household, vehicle type, main user of the vehicle, the other users in priority of the use of the vehicle. A check is performed of vehicle type and in case of a four-wheeler or bike a maximum of two users (u1 and u2) are given after the main user. In case of a bicycle only one user after the main user is entered.

- The program then creates vehicle diaries depending on the type of the vehicle. It first creates the vehicle diary for a car. Among all the trips the program identifies which trips have the main mode by speed as car. Then it assigns which vehicle was used for the trip amongst all the cars present in the household. Then by identifying the vehicle in use it then finds out the main user of the vehicle by comparing it with the person number of the trip in serial order. It checks the main user of the vehicle by seeing the priority of use of the vehicle by the different members of the household. Once the main user is identified then it finds out whether that trip was a joint trip or not. In case of joint trip it also stores the same vehicle information and the main user information in the other joint trips.
- In case there are two persons who use the same vehicle at the same time for different trips then the program corrects the flaw and assigns different vehicles for different trips. The assignment is done depending upon the priority of the users of the vehicles. Also in case of joint trips the other trips are modified suitably.
- Then the program creates vehicle diaries for motorbikes and cycles in the similar manner as that of cars by performing the check that the trip main mode by speed is that of a bike or cycle respectively. In case of public transport or walking the trip is assigned zero vehicle number.
- The output of the program is created in the form of vehicle diaries depending upon the total number of vehicles in the household. The output is stored in the .txt format and numbered according to the vehicle number in the v\_sum file.

## 2.3 Output of the program

The output of the program is in the form of vehicle diaries, which is in the .txt format. The vehicle diaries contain the following variables.

1. Trip no: the trip number of the vehicle in use.
2. Studyc: the study code for Main study or the Pre test study.
3. Cityc: the city code of Halle or Karlsruhe.
4. Hhnr: the household number of the person taking the the trip.
5. Pnr: the person number in the household.
6. Vecho: the number of the vehicle.
7. Muser: the main user of the vehicle
8. Nop: the number of persons in the vehicle.
9. Dorp: the day of reporting period of the person taking the trip.
10. Tnr: the trip number of the trip.
11. Tdep: the time of departure of the trip taken by the person.
12. Tarr: the time of arrival of the trip taken by the person.
13. Type: the type of the vehicle.

## 3. Sample Vehicle Diary

A sample vehicle diary as created by the program is shown in Figure 1.

Figure 1 Sample Vehicle Diary

---

tripno	studyc	cityc	hnnr	pnr	vechno	muser	nop	dorp	tnr	tdep	tarr	type
1	2	1	76	1	1	1	1	1	2	48600	49200	5
2	2	1	76	1	1	1	1	1	3	50400	51000	5
3	2	1	76	1	1	1	1	1	6	80400	81600	5
4	2	1	76	1	1	1	1	2	1	25800	27300	5
5	2	1	76	1	1	1	1	2	2	27600	29100	5
6	2	1	76	1	1	1	1	9	3	66300	68400	5
7	2	1	76	1	1	1	1	18	4	79200	80400	5
8	2	1	76	1	1	1	1	19	1	36000	36600	5
9	2	1	76	1	1	1	1	19	2	37200	38100	5
10	2	1	76	1	1	1	1	19	3	46800	48000	5
11	2	1	76	1	1	1	1	19	4	61200	62400	5
12	2	1	76	1	1	1	1	22	5	84600	85800	5
13	2	1	76	1	1	1	1	24	3	64200	65700	5
14	2	1	76	1	1	1	1	26	5	71100	72000	5
15	2	1	76	1	1	1	1	28	4	61200	62700	5
16	2	1	76	1	1	1	1	32	3	63600	64560	5
17	2	1	76	1	1	1	1	32	4	73200	73500	5
18	2	1	76	1	1	1	1	32	5	81000	81600	5
19	2	1	76	1	1	1	1	33	3	58800	59100	5
20	2	1	76	1	1	1	1	33	4	61200	61500	5
21	2	1	76	1	1	1	1	35	5	76200	76800	5
22	2	1	76	1	1	1	1	36	5	70500	71400	5
23	2	1	76	1	1	1	1	36	6	80400	80700	5
24	2	1	76	1	1	1	1	36	7	84600	85500	5
25	2	1	76	1	1	1	1	39	3	64200	65100	5
26	2	1	76	1	1	1	1	39	4	73200	73500	5
27	2	1	76	1	1	1	1	39	5	81000	81900	5

---

## 4 Acknowledgements

The author gratefully acknowledges the support of Mr. K. W. Axhausen, Mr. S Schönfelder, and Mr. A König for their constant guidance and help in analysing and understanding the data and preparing the report.



## 5 Literature

- Axhausen, K.W., A. Zimmermann, S. Schönfelder, G. Rindsfuser and T. Haupt (2000) Observing the rhythms of daily life: A six-week travel diary, *Arbeitsberichte Verkehrs - und Raumplanung*, **25**, Institut für Verkehrsplanung und Transporttechnik, ETH, Zurich.
- Lafore, R. (1991) *Object oriented programming in C++*, Galgotia , New Delhi.
- PTV AG, B. Fell, S. Schönfelder and K.W. Axhausen (2000) Mobidrive Questionnaires, *Arbeitsberichte Verkehrs - und Raumplanung*, **52**, Institut für Verkehrsplanung und Transporttechnik, ETH, Zurich.
- Schlich, R. A. König, A. Aschwanden, A. Kaufmann und K.W. Axhausen (2000) Mobidrive Data format guide, *Arbeitsberichte Verkehrs - und Raumplanung*, **35**, Institut für Verkehrsplanung und Transporttechnik, ETH, Zurich
- Singhi, P. (2001) Analysis of joint trips using C++ in Mobidrive, *Arbeitsbericht Verkehrs- und Raumplanung*, **89**, Institut für Verkehrsplanung und Transporttechnik, Strassen- und Eisenbahnbau, ETH, Zürich.

## Appendix A: Source Code of the program

```
#include <iostream.h>
#include <fstream.h>
#include <iomanip.h>
```

```
class info
{
public:
    int no;
    int hhnr;
    int pnr;
    int tdep;
    int tarr;
    int jtrip;
    int tmms;
    int nop;
    int dorp;
    int cityc;
    int studyc;
    int vechno;
    int tnr;
    int muser;
    int test;
    int type;
};
```

```
class info1
{
public:
    int num;
    int type;
    int mu;
    int u1;
    int u2;
};
```

```
const int MAX = 1000;
```

```
const int max = 8;           //total no of vehicles in the households
```

```
void main()
{
    info arr[MAX];

    info1 vech[max];

    int i,j,k,h,no1,no2,no3,no4,no5,no6,no7,no8,test=0;

    for(i=0;i<MAX;i++)          // initializing the array
    {
        arr[i].hhnr=0;
        arr[i].pnr=0;
        arr[i].tarr=0;
        arr[i].tdep=0;
        arr[i].dorp=0;
        arr[i].tmmbs=0;
        arr[i].cityc=0;
        arr[i].jtrip=0;
        arr[i].vechno=0;
        arr[i].tnr=0;
        arr[i].muser=0;
        arr[i].test=0;
    }

    ifstream inFile;           // file input

    inFile.open("h105.txt",ios::in); // reading the file input from the joint trip file

    for(i=0;i<MAX;i++)
    {
        inFile >> arr[i].studyc >> arr[i].cityc >> arr[i].hhnr >> arr[i].pnr >> arr[i].jtrip >>
        arr[i].nop >> arr[i].tdep >> arr[i].tarr >> arr[i].tmmbs >> arr[i].dorp >> arr[i].tnr;
    }

    for(i=0;i<max;i++)
    {
        cout << " enter the details for the " << i+1 << " vehicle no \n" ; // entry for
                                                //each vehicle user

        vech[i].num=i+1;
        cout << vech[i].num << "\n";
        cout << " enter the type of vehicle \n";
        cin >> vech[i].type;
    }
}
```

```

    if(vech[i].type==5 || vech[i].type==6 || vech[i].type==2 || vech[i].type==3 ||
vech[i].type==4)
    {
        cout << " enter the main user \n";           //case of four wheeler or bike
        cin >> vech[i].mu;
        cout << " enter the second user in preference \n";
        cin >> vech[i].u1;
        cout << " enter the third user in preference \n" ;
        cin >> vech[i].u2;
    }

    if(vech[i].type==1)                               //case of bicycle
    {
        cout << " enter the main user \n";
        cin >> vech[i].mu;
        cout << " enter the second user in preference \n";
        cin >> vech[i].u1;
    }

}

// vehicle diary for a car

for(i=0;i<MAX;i++)
{
    if(arr[i].tmbs==4) // checks the mode is car
    {
        for(j=0;j<max;j++)
        {
            if(vech[j].type==5 || vech[j].type==6) //checks the type of the
                                                    //vehicle
            {
                if(vech[j].u2==arr[i].pnr) //incase the second user matches the
                                                    //person number
                {
                    arr[i].vechno=vech[j].num; //stores the vehicle
                                                    // information in the trip
                    arr[i].muser=vech[j].mu;
                    arr[i].type=vech[j].type;
                    arr[i].test=1;
                }
            }
        }
    }
}

```

```

        if(arr[i].jtrip==1)
        {
            for(k=0;k<MAX;k++) //checks if it is a joint trip and
                               // stores the same vehicle information in
                               // the joint trips
            {
                if(arr[k].tdep==arr[i].tdep && arr[k].pnr!=arr[i].pnr &&
                    arr[k].dorp==arr[i].dorp && arr[k].tmmbs==5)
                {
                    arr[k].vechno=vech[j].num;
                    arr[k].muser=vech[j].mu;
                    arr[k].type=vech[j].type;
                }
            }
            break;
        }
    }
}

```

```

for(j=0;j<max;j++)
{
    if(vech[j].type==5 || vech[j].type==6) //checks the type of the
                                             //vehicle
    {
        if(vech[j].u1==arr[i].pnr) //incase the first user matches the
                                     //person number
        {
            arr[i].vechno=vech[j].num; //stores the vehicle
                                         // information in the trip
            arr[i].muser=vech[j].mu;
            arr[i].type=vech[j].type;
            arr[i].test=1;

            if(arr[i].jtrip==1)
            {
                for(k=0;k<MAX;k++) //checks if it is a joint trip
                                     // and stores information in them
                {
                    if(arr[k].tdep==arr[i].tdep && arr[k].pnr!=arr[i].pnr &&
                        arr[k].dorp==arr[i].dorp && arr[k].tmmbs==5)
                    {
                        arr[k].vechno=vech[j].num;
                        arr[k].muser=vech[j].mu;
                        arr[k].type=vech[j].type;
                    }
                }
            }
        }
    }
}

```

```

        }
        break;
    }
}

for(j=0;j<max;j++)
{
    if(vech[j].type==5 || vech[j].type==6) //checks the type of the
        //vehicle
    {
        if(vech[j].mu==arr[i].pnr) //incase the main user matches the
            //person number
        {
            arr[i].vechno=vech[j].num; //stores the vehicle
            // information in the trip
            arr[i].muser=vech[j].mu;
            arr[i].type=vech[j].type;
            arr[i].test=1;

            if(arr[i].jtrip==1)
            {
                for(k=0;k<MAX;k++) //checks if it is a joint trip and
                    //stores the same vehicle
                    // information in the joint trips
                {
                    if(arr[k].tdep==arr[i].tdep && arr[k].pnr!=arr[i].pnr &&
                        arr[k].dorp==arr[i].dorp && arr[k].tmmbs==5)
                    {
                        arr[k].vechno=vech[j].num;
                        arr[k].muser=vech[j].mu;
                        arr[k].type=vech[j].type;
                    }
                }
            }
            break;
        }
    }
}

for(i=0;i<MAX;i++) //this part of the program checks if two persons are using the same
{
    //vehicle for diffenet trips at the same time then it corrects the flaw
    // and assigns different vehicle for different trips at the same time

    for(j=i;j<MAX;j++)
    {

```

```

if(i!=j && arr[j].tmmbs==4)
{
if(arr[j].jtrip==0) //checks for two trips with the same vehicle at the
//same time
{
if(arr[i].tdep==arr[j].tdep && arr[i].vechno==arr[j].vechno &&
arr[i].dorp==arr[j].dorp)
{
if(vech[arr[j].vechno-1].mu==arr[i].pnr) //if the main user
//is in the trip[I]
{
// then vehicle for the trip[j] is changed

for(k=0;k<max;k++)
{
if(vech[k].type==5 && vech[k].num != arr[i].vechno)
{
//assigns a different vehicle to the trip[j]
arr[j].vechno=vech[k].num;
arr[j].muser=vech[k].mu;
arr[j].type=vech[k].type;

for(h=0;h<MAX;h++)
{
if(arr[h].tdep==arr[j].tdep &&
arr[h].pnr!=arr[j].pnr && arr[h].dorp==arr[j].dorp && arr[h].tmmbs==5)
{
arr[h].vechno=vech[j].num;
arr[h].muser=vech[j].mu; //modifies the vehicle information for the joint trip
arr[h].type=vech[j].type;
}
}
}
break;
}
break;
}

if(vech[arr[j].vechno-1].mu==arr[j].pnr) //if the main user
//is in the trip[i]
{
//then vehicle for the trip[j] is changed

for(k=0;k<max;k++)
{
if(vech[k].type==5 && vech[k].num != arr[j].vechno)
{
//assigns a different vehicle to the trip[i]
arr[i].vechno=vech[k].num;
arr[i].muser=vech[k].mu;

```

```
        arr[i].type=vech[k].type;

        for(h=0;h<MAX;h++)
        {
            if(arr[h].tdep==arr[i].tdep &&
arr[h].pnr!=arr[i].pnr && arr[h].dorp==arr[i].dorp && arr[h].tmmbs==5)
            {
                arr[h].vechno=vech[i].num;
                arr[h].muser=vech[i].mu; //modifies the vehicle information for the joint trip
                arr[h].type=vech[i].type;
            }
        }
        break;
    }
    break;
}
}
}
}

//vehicle diary for the motorbike
for(i=0;i<MAX;i++)
{
    if(arr[i].tmmbs==3) //checks the mode used is bike
    {
        for(j=0;j<max;j++)
        {
            if(vech[j].type==2 || vech[j].type==3 || vech[j].type==4) //checks
            // if the vehicle used is a bike
            {
                if(vech[j].u2==arr[i].pnr) //incase the second
            //matches the
            // the person no
                {
                    arr[i].vechno=vech[j].num;
                    arr[i].muser=vech[j].mu; //stores vehicle
            //information
                    arr[i].type=vech[j].type; //in the trip
                    arr[i].test=1;

                    if(arr[i].jtrip==1) //checks for
            //the joint trip
            // and stores the same vehicle information in the joint trip
                    {
```



```

                                for(k=0;k<MAX;k++)
                                {
                                if(arr[k].tdep==arr[i].tdep &&
arr[k].pnr!=arr[i].pnr && arr[k].dorp==arr[i].dorp && arr[k].tmmbs==5)
                                {
                                arr[k].vechno=vech[j].num;
                                arr[k].muser=vech[j].mu;
                                arr[k].type=vech[j].type;
                                }
                                }
                                }
                                break;
                                }
                                }
                                }

for(j=0;j<max;j++)
{
    if(vech[j].type==2 || vech[j].type==3 || vech[j].type==4) //checks
        // if the vehicle used is a bike
    {
        if(vech[j].u1==arr[i].pnr) //incase the first user
            //matches the person no
        {
            arr[i].vechno=vech[j].num;
            arr[i].muser=vech[j].mu; //stores vehicle
            //information
            arr[i].type=vech[j].type; //in the trip
            arr[i].test=1;

            if(arr[i].jtrip==1) //checks for
                // joint trip and stores
                // the same vehicle information in the joint trip

            {
                for(k=0;k<MAX;k++)
                {
                if(arr[k].tdep==arr[i].tdep &&
arr[k].pnr!=arr[i].pnr && arr[k].dorp==arr[i].dorp && arr[k].tmmbs==5)
                {
                arr[k].vechno=vech[j].num;
                arr[k].muser=vech[j].mu;
                arr[k].type=vech[j].type;
                }
                }
                }
                break;
            }
        }
    }
}

```

```
        }
    }
    for(j=0;j<max;j++)
    {
        if(vech[j].type==2 || vech[j].type==3 || vech[j].type==4) //checks
            // if the vehicle used is a bike
        {
            if(vech[j].mu==arr[i].pnr) //incase the main
                //user matches
                // the person no
            {
                arr[i].vechno=vech[j].num;
                arr[i].muser=vech[j].mu; //stores vehicle
                //information
                arr[i].type=vech[j].type; //in the trip
                arr[i].test=1;

                if(arr[i].jtrip==1) //checks for
                    //the joint trip and stores
                    // same vehicle information in the joint trip
                {
                    for(k=0;k<MAX;k++)
                    {
                        if(arr[k].tdep==arr[i].tdep &&
arr[k].pnr!=arr[i].pnr && arr[k].dorp==arr[i].dorp && arr[k].tmmbs==5)
                        {
                            arr[k].vechno=vech[j].num;
                            arr[k].muser=vech[j].mu;
                            arr[k].type=vech[j].type;
                        }
                    }
                }
                break;
            }
        }
    }
}
}
```

```
// vehicle diary for the cycle
for(i=0;i<MAX;i++)
{
```

```

if(arr[i].tmmbs==2)    //checks if the mode is cycling
{
    for(j=0;j<max;j++)
    {
        if(vech[j].type==1)    //checks if the vehicle type is cycle
        {
            if(vech[j].u1==arr[i].pnr) //checks in case for the first user
            {
                arr[i].vechno=vech[j].num; //assigns the vehicle
                arr[i].muser=vech[j].mu;    //to the trip
                arr[i].type=vech[j].type;
                break;
            }
        }
    }

    for(j=0;j<max;j++)
    {
        if(vech[j].type==1)
        {
            if(vech[j].mu==arr[i].pnr) //checks in case for the main
            //user
            {
                arr[i].vechno=vech[j].num; //assigns the vehicle
                arr[i].muser=vech[j].mu;    //to the trip
                arr[i].type=vech[j].type;
                break;
            }
        }
    }
}

for(i=0;i<MAX;i++)    //incase of public transport or walking mode
{
    if(arr[i].tmmbs==1||arr[i].tmmbs==6||arr[i].tmmbs==7||arr[i].tmmbs==8||arr[i].tmmbs==9)
    {
        arr[i].vechno=0;
    }
}

```

```
ofstream outFile,outfile1,outfile2,outfile3,outfile4,outfile5,outfile6,outfile7; //file output
```

```
outFile.open("v1.txt", ios::out || ios::app); //creates eight vehicle diaries in the .txt format
outfile1.open("v2.txt", ios::out || ios::app);
outfile2.open("v3.txt", ios::out || ios::app);
outfile3.open("v4.txt", ios::out || ios::app);
outfile4.open("v5.txt", ios::out || ios::app);
outfile5.open("v6.txt", ios::out || ios::app);
outfile6.open("v7.txt", ios::out || ios::app);
outfile7.open("v8.txt", ios::out || ios::app);
```

```
    outFile << setw(4) << "tripno" << setw(8) << "studyc" << setw(10) << "cityc" <<
    setw(8) << "hnr" << setw(8) << "pnr" << setw(8) <<"vechno" << setw(6) << "muser" <<
    setw(6) << "nop" << setw(6) << "dorp" << setw(6) << "tnr" << setw(8) << "tdep" << setw(8)
    << "tarr" << setw(6) << "type" << "\n";
```

```
    outfile1 << setw(4) << "tripno" << setw(8) << "studyc" << setw(10) << "cityc" <<
    setw(8) << "hnr" << setw(8) << "pnr" << setw(8) <<"vechno" << setw(6) << "muser" <<
    setw(6) << "nop" << setw(6) << "dorp" << setw(6) << "tnr" << setw(8) << "tdep" << setw(8)
    << "tarr" << setw(6) << "type" << "\n";
```

```
    outfile2 << setw(4) << "tripno" << setw(8) << "studyc" << setw(10) << "cityc" <<
    setw(8) << "hnr" << setw(8) << "pnr" << setw(8) <<"vechno" << setw(6) << "muser" <<
    setw(6) << "nop" << setw(6) << "dorp" << setw(6) << "tnr" << setw(8) << "tdep" << setw(8)
    << "tarr" << setw(6) << "type" << "\n";
```

```
    outfile3 << setw(4) << "tripno" << setw(8) << "studyc" << setw(10) << "cityc" << setw(8)
    << "hnr" << setw(8) << "pnr" << setw(8) <<"vechno" << setw(6) << "muser" << setw(6)
    << "nop" << setw(6) << "dorp" << setw(6) << "tnr" << setw(8) << "tdep" << setw(8) <<
    "tarr" << setw(6) << "type" << "\n";
```

```
    outfile4 << setw(4) << "tripno" << setw(8) << "studyc" << setw(10) << "cityc" <<
    setw(8) << "hnr" << setw(8) << "pnr" << setw(8) <<"vechno" << setw(6) << "muser" <<
    setw(6) << "nop" << setw(6) << "dorp" << setw(6) << "tnr" << setw(8) << "tdep" << setw(8)
    << "tarr" << setw(6) << "type" << "\n";
```

```
    outfile5 << setw(4) << "tripno" << setw(8) << "studyc" << setw(10) << "cityc" <<
    setw(8) << "hnr" << setw(8) << "pnr" << setw(8) <<"vechno" << setw(6) << "muser" <<
    setw(6) << "nop" << setw(6) << "dorp" << setw(6) << "tnr" << setw(8) << "tdep" << setw(8)
    << "tarr" << setw(6) << "type" << "\n";
```

```
    outfile6 << setw(4) << "tripno" << setw(8) << "studyc" << setw(10) << "cityc" <<
    setw(8) << "hnr" << setw(8) << "pnr" << setw(8) <<"vechno" << setw(6) << "muser" <<
    setw(6) << "nop" << setw(6) << "dorp" << setw(6) << "tnr" << setw(8) << "tdep" << setw(8)
    << "tarr" << setw(6) << "type" << "\n";
```

```

outfile7 << setw(4) << "tripno" << setw(8) << "studyc" << setw(10) << "cityc" <<
setw(8) << "hhnr" << setw(8) << "pnr" << setw(8) << "vechno" << setw(6) << "muser" <<
setw(6) << "nop" << setw(6) << "dorp" << setw(6) << "tnr" << setw(8) << "tdep" << setw(8)
<< "tarr" << setw(6) << "type" << "\n";

```

```
no1=1;no2=1;no3=1;no4=1;no5=1;no6=1;no7=1,no8=1;
```

```

for(i=0;i<MAX;i++) //output of the program
{
cout << "no. " << i+1 << " number of vechicle " << arr[i].vechno << "\n ";

if(arr[i].vechno==1 && arr[i].test==1)
{
outfile << setw(4) << no1 << setw(8) << arr[i].studyc << setw(10) << arr[i].cityc <<
setw(8) << arr[i].hhnr << setw(8) << arr[i].pnr << setw(8) << arr[i].vechno << setw(6) <<
arr[i].muser << setw(8) << arr[i].nop << setw(6) << arr[i].dorp << setw(6) << arr[i].tnr <<
setw(8) << arr[i].tdep << setw(8) << arr[i].tarr << setw(6) << arr[i].type << "\n";
no1++;
}

if(arr[i].vechno==2 && arr[i].test==1)
{
outfile1 << setw(4) << no2 << setw(8) << arr[i].studyc << setw(10) << arr[i].cityc
<< setw(8) << arr[i].hhnr << setw(8) << arr[i].pnr << setw(8) << arr[i].vechno << setw(6) <<
arr[i].muser << setw(8) << arr[i].nop << setw(6) << arr[i].dorp << setw(6) << arr[i].tnr <<
setw(8) << arr[i].tdep << setw(8) << arr[i].tarr << setw(6) << arr[i].type << "\n";
no2++;
}

if(arr[i].vechno==3 && arr[i].test==1)
{
outfile2 << setw(4) << no3 << setw(8) << arr[i].studyc << setw(10) << arr[i].cityc
<< setw(8) << arr[i].hhnr << setw(8) << arr[i].pnr << setw(8) << arr[i].vechno << setw(6) <<
arr[i].muser << setw(8) << arr[i].nop << setw(6) << arr[i].dorp << setw(6) << arr[i].tnr <<
setw(8) << arr[i].tdep << setw(8) << arr[i].tarr << setw(6) << arr[i].type << "\n";
no3++;
}

if(arr[i].vechno==4 && arr[i].test==1)
{
outfile3 << setw(4) << no4 << setw(8) << arr[i].studyc << setw(10) << arr[i].cityc
<< setw(8) << arr[i].hhnr << setw(8) << arr[i].pnr << setw(8) << arr[i].vechno << setw(6) <<
arr[i].muser << setw(8) << arr[i].nop << setw(6) << arr[i].dorp << setw(6) << arr[i].tnr <<
setw(8) << arr[i].tdep << setw(8) << arr[i].tarr << setw(6) << arr[i].type << "\n";
no4++;
}

```

```
if(arr[i].vechno==5 && arr[i].test==1)
{
    outfile3 << setw(4) << no5 << setw(8) << arr[i].studyc << setw(10) << arr[i].cityc
<< setw(8) << arr[i].hhnr << setw(8) << arr[i].pnr << setw(8) << arr[i].vechno << setw(6) <<
arr[i].muser << setw(8) << arr[i].nop << setw(6) << arr[i].dorp << setw(6) << arr[i].tnr <<
setw(8) << arr[i].tdep << setw(8) << arr[i].tarr << setw(6) << arr[i].type << "\n";
    no5++;
}

if(arr[i].vechno==6 && arr[i].test==1)
{
    outfile3 << setw(4) << no6 << setw(8) << arr[i].studyc << setw(10) << arr[i].cityc
<< setw(8) << arr[i].hhnr << setw(8) << arr[i].pnr << setw(8) << arr[i].vechno << setw(6) <<
arr[i].muser << setw(8) << arr[i].nop << setw(6) << arr[i].dorp << setw(6) << arr[i].tnr <<
setw(8) << arr[i].tdep << setw(8) << arr[i].tarr << setw(6) << arr[i].type << "\n";
    no6++;
}

if(arr[i].vechno==7 && arr[i].test==1)
{
    outfile3 << setw(4) << no7 << setw(8) << arr[i].studyc << setw(10) << arr[i].cityc
<< setw(8) << arr[i].hhnr << setw(8) << arr[i].pnr << setw(8) << arr[i].vechno << setw(6) <<
arr[i].muser << setw(8) << arr[i].nop << setw(6) << arr[i].dorp << setw(6) << arr[i].tnr <<
setw(8) << arr[i].tdep << setw(8) << arr[i].tarr << setw(6) << arr[i].type << "\n";
    no7++;
}

if(arr[i].vechno==8 && arr[i].test==1)
{
    outfile3 << setw(4) << no8 << setw(8) << arr[i].studyc << setw(10) << arr[i].cityc
<< setw(8) << arr[i].hhnr << setw(8) << arr[i].pnr << setw(8) << arr[i].vechno << setw(6) <<
arr[i].muser << setw(8) << arr[i].nop << setw(6) << arr[i].dorp << setw(6) << arr[i].tnr <<
setw(8) << arr[i].tdep << setw(8) << arr[i].tarr << setw(6) << arr[i].type << "\n";
    no8++;
}
}
}
```

Die *Arbeitsberichte Verkehrs- und Raumplanung* dienen der schnellen Verbreitung der Ergebnisse der Arbeit der Mitarbeitenden und Gäste des Instituts. Die Verantwortung für Inhalt und Gestaltung liegt alleine bei den Autor/innen.

The *Working Papers Traffic and Spatial Planning* are intended for the quick dissemination of the results of the members and guests of the Institut. Their content is the sole responsibility of the authors.

Eine vollständige Liste der Berichte kann vom Institut angefordert werden:

A complete catalog of the papers can be obtained from:

IVT ETHZ  
ETH Hönggerberg (HIL)  
CH - 8093 Zürich

Telephon: +41 (0)1 633 31 05  
Telefax: +41 (0)1 633 10 57  
E-Mail: [hotz@ivt.baug.ethz.ch](mailto:hotz@ivt.baug.ethz.ch)  
WWW: [www.ivt.baug.ethz.ch](http://www.ivt.baug.ethz.ch)

Der Katalog kann auch abgerufen werden von:

The catalog can also be obtained from:

[http://www.ivt.baug.ethz.ch/veroeffent\\_arbeitsbericht.html](http://www.ivt.baug.ethz.ch/veroeffent_arbeitsbericht.html)