RESEARCH

2009-33

Development of Data Warehouse and Applications for Continuous Vehicle Class and Weigh-in-Motion Data

Take the steps... Research...Knowledge...Innovative Solutions!

Transportation Research

Technical Report Documentation Page

1. Report No.	2.	3. Recipients Accession No.
MN/RC 2009-33		

4. Title and Subtitle	5. Report Date
Development of Data Warehouse and Applications for Continuous Vehicle Class and Weigh-in-Motion Data	October 2009
	6.
7. Author(s)	8. Performing Organization Report No.
Taek M. Kwon	
9. Performing Organization Name and Address	10. Project/Task/Work Unit No.
Transportation Data Research Laboratory Department of Electrical and Computer Engineering University of Minnesota Duluth 271 Marshall W. Alworth Hall 1023 University Drive Duluth, Minnesota 55812	11. Contract (C) or Grant (G) No. (c) 89261 (wo) 41
12. Sponsoring Organization Name and Address	13. Type of Report and Period Covered
Minnesota Department of Transportation 395 John Ireland Boulevard Mail Stop 330 St. Paul, Minnesota 55155	Final Report
	14. Sponsoring Agency Code

15. Supplementary Notes
http://www.lrrb.org/pdf/200933.pdf

16. Abstract (Limit: 250 words)

Presently, the Office of Transportation Data & Analysis (TDA) at the Minnesota Department of Transportation (Mn/DOT) manages 29 Vehicle Classification (VC) sites and 12 Weigh-in-Motion (WIM) sites installed on various Minnesota roadways. The data is collected 24/7 from all sites, resulting in a large amount of data. The total amount of data is expected to substantially grow with time due to the continuous accumulation of data from the present sites and future expansion of sites. Therefore, there is an urgent need to develop an efficient data management strategy for dealing with the present needs and future growth of this data. The solution proposed in this research project is to develop a centralized data warehouse from which all applications can acquire the data. The objective of this project was to develop software for creating a VC/WIM data warehouse and example applications that utilize it. This project was successfully completed by developing the software necessary to build the VC/WIM data warehouse and the application software packages that utilize the data. The main contribution of this project is that it provides a single access point for querying all of the Mn/DOT's WIM and VC data, from which many more applications can be developed without concerns of proprietary binary formats.

17. Document Analysis/Descriptors	18. Availability Statement
Data warehouses, Weigh in motion, Vehicle classification, Mechanistic design, Pavement design, MEPDG, Data reports, Reports	No restrictions. Document available from: National Technical Information Services, Springfield, Virginia 22161

19. Security Class (this report)	20. Security Class (this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	106	

Development of Data Warehouse and Applications for Continuous Vehicle Class and Weigh-in-Motion Data

Final Report

Prepared by

Taek Mu Kwon

Transportation Data Research Laboratory
Department of Electrical and Computer Engineering
University of Minnesota Duluth

October 2009

Published by

Minnesota Department of Transportation
Research Services Section
395 John Ireland Boulevard, MS 330
St. Paul, Minnesota 55155-1899

ACKNOWLEGEMENTS

This research was supported by the Minnesota Department of Transportation (Mn/DOT). The author would like to thank the technical advisory panel members for this project: Bill Martinson, Mark Novak, Matt Oman, Chu Wei and Tom Nelson. They provided many valuable suggestions that made the deliverables of this project directly applicable for vehicle classification and weigh-in-motion data. Research associate Ryan Weidemann helped to initially formulate pdf pages.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

Presently, the Office of Transportation Data & Analysis (TDA) at the Minnesota Department of Transportation (Mn/DOT) manages 29 Vehicle Classification (VC) sites and 12 Weigh-in-Motion (WIM) sites installed on various Minnesota roadways. The data is collected 24/7 from all sites, resulting in a large amount of data. The total amount of data is expected to substantially grow with time due to the continuous accumulation of data from the present sites and future expansion of sites. Therefore, there is an urgent need to develop an efficient data management strategy for dealing with the present needs and future growth of this data. The solution proposed in this research project is to develop a centralized data warehouse from which all applications can acquire the data. The objective of this project was to develop software for creating a VC/WIM data warehouse and example applications that utilize it.

The data polled from the VC and WIM sites is in proprietary binary formats that can only be read and processed by the proprietary software packages supplied by device manufacturers. Since Mn/DOT uses several different types and models of VC and WIM data collection devices, the binary formats are diverse and no single software package is presently able to read and process the data. This creates a problem, especially when a data summary report or a statistical analysis for all sites is needed. In order to solve this problem, this project developed two standardized unified formats: one for WIM and another for VC data. The unified formats were developed using ASCII comma-separated values (csv) and can be read by any text editors or spreadsheet programs. The data warehouse was constructed using the daily data files formatted according to the unified formats. Since the data warehouse is built on sharable network storage and the data formats are standardized, it provides a single access point to any application that needs a query of the VC and WIM data. In order to provide efficiency in data query, a tree data structure was used for the organization of the data warehouse.

Along with the development of a VC/WIM data warehouse, three application software packages were developed. They are BullReport, BullGuide, and BullPiezo, all of which can be installed and run from any personal computer (PC) as long as it has network access to the data warehouse. These applications maximally utilize the data warehouse and were developed according to the present and future needs specified by the Mn/DOT TDA. The BullReport was developed as an analysis tool for the WIM data and includes a load spectra analysis utility for single, tandem, tridem, quadem, and steer axles; TMG (Traffic Monitoring Guide) reporting and analysis tools; and 21 data summary reporting utilities. The BullGuide was developed to generate MEPDG (Mechanistic-Empirical Pavement Design Guide) input data for Mn/DOT’s WIM sites, which provides a new way of designing pavement. The BullPiezo was developed as a VC data application and is capable of generating seasonal adjustment factors for short-count stations, AADT (Annual Average Daily Traffic), and monthly average truck traffic.

This project was successfully completed by developing the software necessary to build the VC/WIM data warehouse and the application software packages that utilize the data. The data warehouse was successfully built at Mn/DOT and utilized in daily tasks. The main contribution of this project is that it provides a single access point for querying all of the Mn/DOT’s WIM and VC data, from which many more applications can be developed without concerns of proprietary binary formats.

CHAPTER 1: INTRODUCTION

1.1 Background

The main idea of this project was born out of the successes in the previous project entitled "TMC Traffic Data Automation for Mn/DOT's Traffic Monitoring Program," through which remote continuous count data (also called ATR data) and short-duration count (also called short count) data services were established [1]. One of the plausible aspects of this system is that the Office of Transportation Data and Analysis (TDA) at the Minnesota Department of Transportation (Mn/DOT) can create ATR and short count stations at arbitrary locations without actually installing vehicle counters, as long as ITS generated loop data is available. To facilitate this data service, the Northland Advanced Transportation Systems Research Laboratories (NATSRL) at the University of Minnesota Duluth (UMD) provided funding for building a Data Center (DC) which can house a data warehouse for TC traffic and statewide Road Weather Information System (RWIS) data. Presently, Mn/DOT TDA defines ATR and short count stations using a TC loop-detector map and then sends the station definitions to the UMD DC through an on-line data entry table. The UMD DC then generates and delivers the requested data to Mn/DOT TDA through an on-line automated delivery system. ATR data is continuously generated 24/7 once it is defined while the short count data is generated per year. In this model, only the final processed data sets, which are customized for Mn/DOT TDA data needs, are generated and delivered through the Internet, while the computing software and raw data are maintained by the UMD DC. This model combines the software development capabilities of UMD researchers and the practical application knowledge of Mn/DOT employees to share and utilize the available resources. The implemented system is still in operation and has worked well for both UMD and Mn/DOT, providing data research opportunities to UMD and operational benefits to Mn/DOT.

Mn/DOT TDA recognized the benefits of the UMD DC operation and proposed to develop a similar process for their Weigh-in-Motion (WIM) and continuous vehicle classification (VC or also called piezo data at Mn/DOT) data, by creating a data warehouse for WIM and VC data at the UMD DC. Once a VC/WIM data warehouse is established, the UMD DC could develop and run WIM and VC data applications whose specifications are developed by Mn/DOT. Similarly to the ATR and short count data, Mn/DOT then would receive the processed data through the Internet. With the potential growth of the number of VC and WIM sites at Mn/DOT and the corresponding growth in data size, utilization of a large networked data warehouse and high performance computers at UMD DC is an attractive solution. Consequently, the original proposal of this project was developed based on the needs described above. However, this remote data model was later modified to directly build the data warehouse at Mn/DOT instead of UMD DC and also to run the applications directly from Mn/DOT. The main reason for this change was due to the uncertainty of the UMD DC in securing future funds for maintaining the facility and researcher supports. NATSRL has been providing the funds for the UMD DC, but the policy recently changed to only support pure research related activities. It decided not to support any on-going non-research related activities, such as the present data support to Mn/DOT. UMD DC requested operational funds for the present data service, but Mn/DOT also could not secure the funds for the present data services. This created problems for this project. Finally, the scope of the project was modified to establish a data warehouse on Mn/DOT network storage and run the software directly from Mn/DOT desktop PCs. This modified approach has opposing aspects. The positive aspect is that the dependence to UMD DC is totally removed, i.e., Mn/DOT can freely generate the processed data at any time. Also, communication problems such as Internet failures are no longer a concern. On the other hand, the negative aspect is that Mn/DOT now has to maintain the software after the project is completed. For example, if WIM or VC vendors change their data formats, the software must be modified to incorporate the format changes. Also, if a new data utility or modification to existing functions of applications is needed, Mn/DOT is now responsible for the changes, instead of UMD DC researchers. To mitigate the future software upgrade needs, the PI and Technical Advisory Panel (TAP) of this project put their best efforts to ensure that the final software packages delivered incorporate present and future needs and that the software is tested thoroughly.

There are four main tasks in this project: Task-1, Analysis of raw data; Task-2, Development of remote data export systems; Task-3, Development of data warehouse; and Task-4, development of reporting functionalities. The rest of tasks are writing the draft and final reports. In order to incorporate the changes in the UMD DC policy, Task-3 was modified to build a data warehouse at Mn/DOT. Below each task is further described.

In Task-1, WIM and VC raw binary data were analyzed to develop decoding logics. There were three different types of WIM systems used at Mn/DOT which have unique binary formats. VC systems had several versions of binary formats within the same type of systems. Mn/DOT did not have any information on binary formats from any of the products they use, and the information was not available from the companies either. Therefore, the binary formats were decoded purely based on the analysis of binary bit patterns and then verified using the outputs of the software that the companies sold to Mn/DOT. After decoding logics were established, the binary data was translated into unified formats by creating one standard for WIM and another one for VC data. After translation into a unified format, the format differences between manufacturers or different hardware versions no longer exist. The data sets organized using standard formats generally improve the query efficiency because it eliminates the decoding time and does not require use of multiple proprietary software packages for different data collection devices. The VC/WIM data warehouse was thus created using the standardized data formats. The structure used for the data warehouse is also important for query efficiency and maintenance of data. A hierarchical tree structure was implemented to allow fast searches. The binary data decoding software was written to create an organized tree structure at the specified network storage as soon as decoding is completed. Thus, the translated data is immediately organized into a tree structure data warehouse as the raw binary data is decoded. The details on binary formats are discussed in Chapter 3. The VC/WIM data warehouse structure is discussed in Chapter 2.

Task-2 was to develop several pieces of data transport software to create a data warehouse at UMD DC. Three different pieces of software were created, all of which were designed for scheduled runs without any human intervention. The first piece transfers the raw binary data from Mn/DOT sources to a UMD DC server; the second piece organizes the raw data, translates it into a unified format, and then transfers the formatted data to a data server; the third piece transfers processed data from a UMD DC data server to an export server to deliver the data to Mn/DOT. Each piece was thoroughly tested to ensure that they are fully functional. These software pieces created an identical VC/WIM data warehouse at both UMD DC and Mn/DOT with the result that reporting software (Task-4) can be developed and tested from UMD

DC. Because of the identical data structure used at both sides, several software applications created at UMD DC were able to run from Mn/DOT without any modification.
Task-3 was to build a data warehouse. The data tree is automatically constructed when the decoding software developed in Task-1 runs. Mn/DOT successfully installed the decoding software called BullConverter and PeakConverter, and created a data warehouse for WIM and VC data in a network storage. The data can be accessed from desktop PCs as long as the network drive is mounted.

Task-4 was to develop reporting software utilizing the data warehouse previously created. Three main pieces of software were developed, which are BullReport, BullGuide, and BullPiezo. BullReport reads in standardized WIM data and provides three main functions: generation of various data summary reports (21 different report types are presently implemented), TMG formatted data for FHWA submission, and graphical analysis tools for load spectra and TMG formatted data. BullGuide also reads in the standardized WIM data but produces data inputs for MEPDG pavement design. BullPiezo reads in standardized VC data and generates seasonal adjustment factors, AADT for each vehicle type, and monthly traffic distribution. The software was developed as installation packages that can be installed on any PC and can be run as long as the PC has access to the data warehouse. The details on the software packages are described in Chapter 4.

1.2 Overall System Model

The term data warehouse was coined by William Inmon in the 1990s [2], which he defined in the following way: “A data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management’s decision making process.” Subject oriented refers to data that gives information about a particular subject instead of the organization’s ongoing operations. Time-variant indicates all data in the data warehouse is identified with a particular time period. Non-volatile implies that data is stable and should not be modified. This definition is reasonably accurate to describe the UMD DC since it houses a collection of traffic, WIM, VC, and statewide RWIS data with the data storage exceeding 4 terabytes. In general, data warehouse refers to a collection of many types of databases structured for query and analysis. If a data warehouse holds only a single-subject data, it is often referred to as a “data mart” instead of a data warehouse. Therefore, the data warehouse created at Mn/DOT might be better suited with the name, data mart. Nevertheless, the term data warehouse is used for the Mn/DOT data to minimize confusion.

Regardless of the definition of the terminologies, the originally proposed data system is depicted in Figure 1. At Mn/DOT, raw binary data is collected from VC or WIM sites through modem lines, which are then stored to a network storage. This raw data is transferred to the UMD DC through an on-line automation. UMD DC then reorganizes this raw data and translates them into unified data formats, which are stored into the main network storage where the data is organized for efficient query and analysis. This organized data is called the data warehouse. An application server processes the data to produce various reports or data summaries which are transferred to Mn/DOT through an export server. In this model, Mn/DOT simply requests the type of data they need, and the UMD DC computers produce and deliver the requested data. In effort to minimize the recurring cost of maintenance, all transactions are automated through networking software.

The modified data system model due to the uncertainty of the future of UMD DC policy is shown in Figure 2, which simplifies and moves the function of UMD DC to Mn/DOT. In this model, decoding software packages translate the raw data into unified formats and then organize them into a WIM/VC data warehouse for query and analysis. Application software packages are directly installed on Mn/DOT PCs, from which it directly queries and analyzes the unified format data to produce the final intended results. The application software packages that can be installable on Mn/DOT PCs are supplied by the UMD DC. The responsibility of Mn/DOT is first to regularly run the decoding software to maintain the translated data at the Mn/DOT data warehouse and then to run the application software to produce the final results.

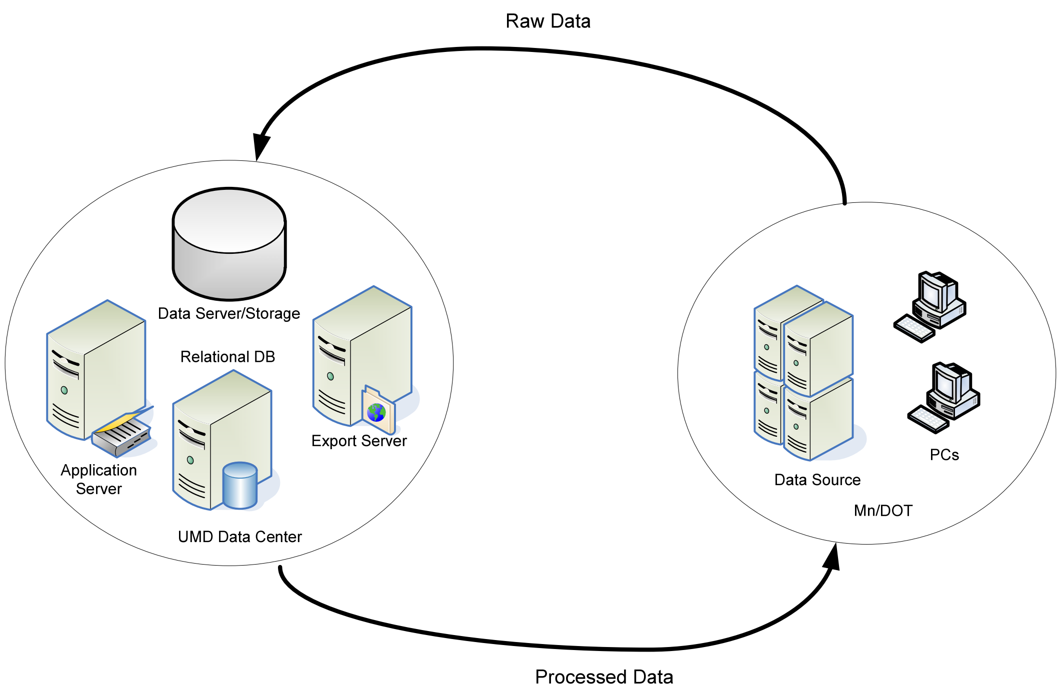


Figure 1: Originally proposed data system model

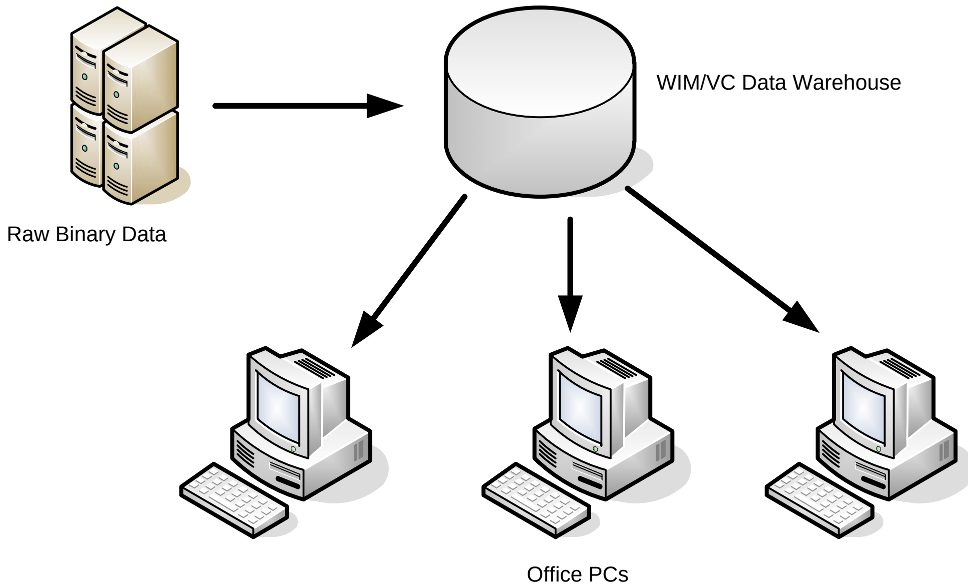


Figure 2: Modified data system model

CHAPTER 2: DATA FORMAT STANDARDIZATION AND ARCHIVE

2.1 Data Standardization

The raw data produced by WIM/VC data collection devices are mostly encoded in proprietary binary formats that can only be understood by the vendor specific software. Even within the same product line, binary formats often change over time because of changes in the operating system, microprocessors, etc. Such discrepancies and protective nature of binary formats limit development of a new application because of unavailability of data decoding information. Therefore, there is a need for unifying the various binary formats into one standardized format that is easy to understand and can remain unchanged for a long time.

Binary formats are compact, so they are often used in data collection devices to cope with the limited storage spaces in the device. Since the objective in this project is to build a data warehouse in a large storage, the storage size is no longer a concern. Hence, it is proposed that ASCII (American Standard Code for Information Exchange) characters are used for all data formats, instead of binary codes. ASCII characters require more storage space, but they are a universally accepted character set that can be read from virtually all computers in any operating system. The data can be either numeric values or text, both of which can be expressed in ASCII codes. Both WIM and VC data can be expressed as a table consisting of multiple columns and rows. It is proposed that each data item, which corresponds to a cell in a row, is separated by a comma and each row is separated by a line break (carriage return and line feed). This format is often referred to as a csv (comma-separated values) format and widely used and accepted by many databases and spreadsheet programs. Another principle used in the standardization is inclusion of information from all manufacturers, i.e., the information encoded in the binary data from all manufacturers should be translated into the standardized ASCII csv format as much as possible. The detailed format definitions of WIM and VC data are described in the subsequent subsections.

2.1.1 Standard WIM data format

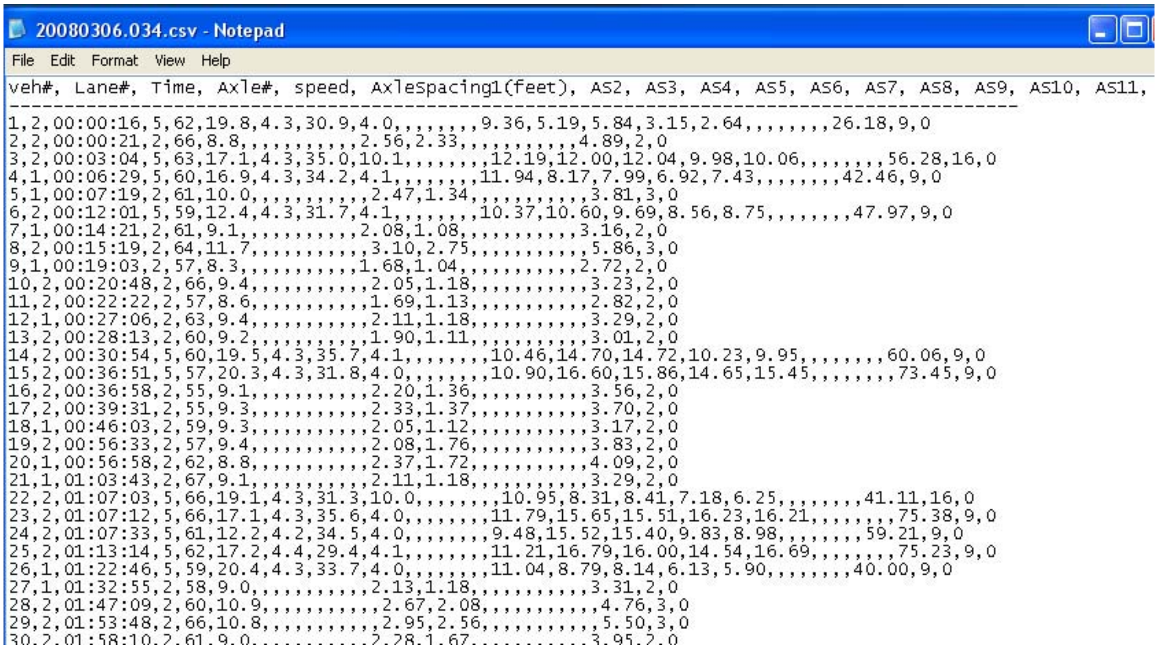
WIM data is recorded as a collection of vehicle records. The important information of a vehicle record is the lane of the vehicle traveled, speed, time, axle spacing, axle weights, and error conditions. Standardized WIM files are ASCII text files and can be read by any text editor, such as a Notepad or a spreadsheet program like Microsoft Excel. The columns of the standard WIM format used in this project are summarized in Table 1. The first line of the file is the column headings; and the second line is a marking line that should be ignored in reading it. The error codes are summarized in Table 2. Equivalent Single Axle Loads (ESALs) are not included as one of the columns, but it can be easily computed using the axle spacing, weights, and pavement type data. Figure 3 shows a screen capture of a WIM data file read by the Notepad program.

Table 1: Column Description of Standardized Vehicle Record File

Column #	Column Heading	Description
1	Veh#	Count or index of vehicles from 12:00AM (midnight).
2	Lane#	Lane number (1, 2, 3, or 4, etc)
3	Time	Time at which vehicle recorded. It is written using the format, hour(0-23):min(0-59):sec (0-59).
4	Axle#	Number of axles of the vehicle.
5	Speed	Speed of the vehicle in miles per hour
6-16 (11 columns)	AxleSpacing (AS1,AS2,…,AS11)	Axle spacing in feet. This distance data is filled between Column 6 and Column 16. Eleven columns are allocated for axle spacing. The unfilled columns are left empty. **Example:** If a vehicle has 3 axles, it has two axle spacings, which will look like: 10,12,,,,,,,,, and indicates that the spacing between the first and second axle is 10 feet and the spacing between the second to third axle is 12 feet
17-28 (12 columns)	AxleWeight (AW1,AW2,…,AW12)	Axle weights in Kips (Kilo-pounds). This weight data is filled between Column 17 and Column 28. Twelve columns are allocated for axle weights, and the unfilled columns are left empty. **Example:** If a vehicle has 3 axles, and the axle weighs are 4.5, 3.6, 7.5 Kips, then the columns should look like: 4.5,3.6,7.5,,,,,,,,,
29	GVW	Gross Vehicle Weight (GVW) is the total load of the vehicle in Kips. It is the sum of each axle weights. In the above example, GVW is 15.6 Kips
30	Class	Vehicle type is displayed in this column. Presently, Mn/DOT uses 1-16.
31	ERR	Shows the error codes in numeric number

Table 2: WIM Vehicle Record Error Code

Error Code	Description
0	Normal, no error
101	Upstream loop failure
102	Downstream loop failure
103	Upstream & downstream loop failure
104	Loop in wrong order
105	Signal idle level too high or too low
106	Maximum number of axles exceeded, max=12
107	Zero axle detected by both up- and downstream axle load sensors
108	Unequal axle counts. Up- and downstream axle counts are not equal
109	Zero axle detected by upstream axle load sensor
110	Axle sensors in wrong order
111	Axle spacing too short
112	Zero axle found by downstream axle load sensor
113	Vehicle too slow
14	Memory buffer failure, queue overflow
15	Not defined
16	Axle on sensor too long
17	Vehicle too fast (likely an error)
18	Loop bounce
19	One axle detected
20	Unknown error
31	Off-scale hit
32	Over height (from height sensor if exists)
33	Significant speed change
34	Significant weight difference
35	Vehicle headway too short
36	Unequal axle detected
37	Wrong lane
38	Tailgating
39	On-scale missed



```
20080306.034.csv - Notepad
File Edit Format View Help
veh#, Lane#, Time, Axle#, speed, AxleSpacing1(feet), AS2, AS3, AS4, AS5, AS6, AS7, AS8, AS9, AS10, AS11,
--------------------------------------------------------------------------------------------------
1,2,00:00:16,5,62,19.8,4.3,30.9,4.0,,,,,,,,9.36,5.19,5.84,3.15,2.64,,,,,,,,26.18,9,0
2,2,00:00:21,2,66,8.8,,,,,,,,,,,2.56,2.33,,,,,,,,,,,4.89,2,0
3,2,00:03:04,5,63,17.1,4.3,35.0,10.1,,,,,,,,12.19,12.00,12.04,9.98,10.06,,,,,,,,56.28,16,0
4,1,00:06:29,5,60,16.9,4.3,34.2,4.1,,,,,,,,11.94,8.17,7.99,6.92,7.43,,,,,,,,42.46,9,0
5,1,00:07:19,2,61,10.0,,,,,,,,,,,2.47,1.34,,,,,,,,,,,3.81,3,0
6,2,00:12:01,5,59,12.4,4.3,31.7,4.1,,,,,,,,10.37,10.60,9.69,8.56,8.75,,,,,,,,47.97,9,0
7,1,00:14:21,2,61,9.1,,,,,,,,,,,2.08,1.08,,,,,,,,,,,3.16,2,0
8,2,00:15:19,2,64,11.7,,,,,,,,,,,3.10,2.75,,,,,,,,,,,5.86,3,0
9,1,00:19:03,2,57,8.3,,,,,,,,,,,1.68,1.04,,,,,,,,,,,2.72,2,0
10,2,00:20:48,2,66,9.4,,,,,,,,,,,2.05,1.18,,,,,,,,,,,3.23,2,0
11,2,00:22:22,2,57,8.6,,,,,,,,,,,1.69,1.13,,,,,,,,,,,2.82,2,0
12,1,00:27:06,2,63,9.4,,,,,,,,,,,2.11,1.18,,,,,,,,,,,3.29,2,0
13,2,00:28:13,2,60,9.2,,,,,,,,,,,1.90,1.11,,,,,,,,,,,3.01,2,0
14,2,00:30:54,5,60,19.5,4.3,35.7,4.1,,,,,,,,10.46,14.70,14.72,10.23,9.95,,,,,,,,60.06,9,0
15,2,00:36:51,5,57,20.3,4.3,31.8,4.0,,,,,,,,10.90,16.60,15.86,14.65,15.45,,,,,,,,73.45,9,0
16,2,00:36:58,2,55,9.1,,,,,,,,,,,2.20,1.36,,,,,,,,,,,3.56,2,0
17,2,00:39:31,2,55,9.3,,,,,,,,,,,2.33,1.37,,,,,,,,,,,3.70,2,0
18,1,00:46:03,2,59,9.3,,,,,,,,,,,2.05,1.12,,,,,,,,,,,3.17,2,0
19,2,00:56:33,2,57,9.4,,,,,,,,,,,2.08,1.76,,,,,,,,,,,3.83,2,0
20,1,00:56:58,2,62,8.8,,,,,,,,,,,2.37,1.72,,,,,,,,,,,4.09,2,0
21,1,01:03:43,2,67,9.1,,,,,,,,,,,2.11,1.18,,,,,,,,,,,3.29,2,0
22,2,01:07:03,5,66,19.1,4.3,31.3,10.0,,,,,,,,10.95,8.31,8.41,7.18,6.25,,,,,,,,41.11,16,0
23,2,01:07:12,5,66,17.1,4.3,35.6,4.0,,,,,,,,11.79,15.65,15.51,16.23,16.21,,,,,,,,75.38,9,0
24,2,01:07:33,5,61,12.2,4.2,34.5,4.0,,,,,,,,9.48,15.52,15.40,9.83,8.98,,,,,,,,59.21,9,0
25,2,01:13:14,5,62,17.2,4.4,29.4,4.1,,,,,,,,11.21,16.79,16.00,14.54,16.69,,,,,,,,75.23,9,0
26,1,01:22:46,5,59,20.4,4.3,33.7,4.0,,,,,,,,11.04,8.79,8.14,6.13,5.90,,,,,,,,40.00,9,0
27,1,01:32:55,2,58,9.0,,,,,,,,,,,2.13,1.18,,,,,,,,,,,3.31,2,0
28,2,01:47:09,2,60,10.9,,,,,,,,,,,2.67,2.08,,,,,,,,,,,4.76,3,0
29,2,01:53:48,2,66,10.8,,,,,,,,,,,2.95,2.56,,,,,,,,,,,5.50,3,0
30,2,01:58:10,2,61,9.0,,,,,,,,,,,2.28,1.67,,,,,,,,,,,3.95,2,0
```

Figure 3: Screen capture of a WIM vehicle record file read by Notepad.

2.1.2 Standard VC data formats

Vehicle classification (VC) data at Mn/DOT refers to binary files that are collected from the data collection devices that operate based on piezoelectric wires, developed by the Peek Traffic Corporation. The data file may contain a number of possible combinations of data, e.g., volume only, volume and speed, or volume, speed, and classification. In the standardized format, each file contains only one of the following types: volume, speed, or classification. Each standardized VC file consists of two header lines followed by data lines. The first header line describes the site and data information. It consists of five fields: siteID, numOfLanes, dataType, date, and lane-by-lane flag, each separated by a comma. Each field of the site header line is summarized in Table 3.

Table 3: Site Header

Header Name	Description
siteID	Three digit site ID
numOfLanes	Number of lanes in the data file
dataType	It is one of vol, spd, or cls.
date	Date expressed in yyyymmdd.
lane-by-lane	True, if the data was collected lane-by-lane. False, if the data was collected without distinction of each lane.

Each item in the first header line is expressed using an equal sign (=) as shown in the following example.

```
SiteID=054,numOfLanes=2,dataType=spd,date=20070114,lane-by-lane=False
```

The second line contains the headers of the data that immediately follows. The three different data types are illustrated one by one using an example.

Volume data

Volume data is recorded as hourly volume with each line starting with the corresponding hour followed by the volumes of each lane. Twenty four data lines, corresponding to each hour of the day, are in the volume data file. An example data file that consists of four lanes is shown in Figure 4. As shown in the first line, the data was collected on Jan 15, 2006 from Site 101 which has four lanes. Presently, all Mn/DOT volume data are collected lane-by-lane, thus the lane-by-lane flag in the first line header appears as true. The time is the beginning time of each hour period.

```
SiteID=101,numOfLanes=4,dataType=vol,date=20060116,lane-by-lane=True
Time,Lane1,Lane2,Lane3,Lane4
00:00,75,52,70,32
01:00,35,35,64,29
02:00,27,26,53,23
03:00,36,12,29,25
04:00,40,11,41,48
05:00,132,43,201,95
06:00,245,144,430,247
07:00,461,374,710,384
08:00,365,261,557,358
09:00,316,249,448,328
10:00,329,252,462,334
11:00,412,297,493,401
12:00,390,312,518,345
13:00,418,370,510,346
14:00,478,446,595,422
15:00,549,607,599,430
16:00,562,672,608,464
17:00,508,566,555,378
18:00,345,323,337,205
19:00,235,213,202,137
20:00,195,174,210,113
21:00,189,162,166,73
22:00,140,126,182,76
23:00,110,111,94,19
```

Figure 4: Volume data example

Speed data

Speed data contains the volume of a specific speed range (speed bins). The range of speed bins are summarized in Table 4. An example is shown in Figure 5. The data header specifies the speed bins in Table 4. The speed bin ranges are set during the device setup.

Table 4: Speed Bin Range

Speed bin	Range
0	0-40 mph
40	40-45 mph
45	45-50 mph
50	50-55 mph
55	55-60 mph
60	60-65 mph
65	65-70 mph
70	70-75 mph
75	75-80 mph
80	80-85 mph
85	85-100 mph
100	100-111 mph
111	111-above mph

```
siteID=054,numOfLanes=2,dataType=spd,date=20070114,lane-by-lane=False
Time,0,40,45,50,55,60,65,70,75,80,85,100,111
00:00,0,0,0,1,1,0,1,0,0,1,0,0,0
01:00,0,1,2,0,0,1,1,0,0,0,0,0,0
02:00,0,0,0,1,1,1,0,0,0,0,0,0,0
03:00,0,0,0,1,0,0,0,0,0,0,0,0,0
04:00,0,0,0,0,0,0,0,0,0,0,0,0,0
05:00,0,0,0,2,0,1,0,0,0,0,0,0,0
06:00,0,0,0,0,0,0,0,0,0,0,0,0,0
07:00,0,0,1,2,0,0,3,0,0,0,0,0,0
08:00,0,0,0,1,1,3,0,0,0,0,1,0,0
09:00,0,0,1,0,3,1,1,2,0,0,0,0,0
10:00,0,1,1,4,6,5,1,1,0,0,0,0,0
11:00,1,0,3,0,11,4,1,0,0,0,0,0,0
12:00,1,0,1,2,5,5,1,0,0,0,0,0,0
13:00,0,1,0,5,4,2,4,0,0,0,0,0,0
14:00,0,1,3,8,3,2,2,0,0,0,0,0,0
15:00,4,3,8,4,3,1,0,0,0,0,0,0,0
16:00,3,2,4,4,8,0,0,0,0,0,0,0,0
17:00,1,3,1,0,2,0,0,0,0,0,0,0,0
18:00,5,1,5,1,1,0,0,0,0,0,0,0,0
19:00,6,0,1,1,0,0,0,0,0,0,0,0,0
20:00,2,1,1,0,0,0,0,0,0,0,0,0,0
21:00,1,1,1,0,0,0,0,0,0,0,0,0,0
22:00,1,2,0,0,0,0,0,0,0,0,0,0,0
23:00,0,0,0,0,0,0,0,0,0,0,0,0
```

Figure 5: Speed data example

Notice from Figure 5 that the lane-by-lane flag is false. If this is a lane-by-lane data file, a Lane column is added right after the Time column. The data is then recorded one line per each lane for the same hour.

Classification data

Classification data contains the volume of each vehicle type at each hour. The type corresponds to FHWA classification which defines types 1 trough 15. An example of classification data expressed in lane-by-lane is shown in Figure 6. This site has four lanes, thus it consists of a total of 96 data lines. A classification algorithm is directly implemented into the data collection device and not adjustable from the data.

```
SiteID=204,numOfLanes=4,dataType=cls,date=20080117,lane-by-lane=True
Time,Lane#,Type1,Type2,Type3,Type4,Type5,Type6,Type7,Type8,Type9,Type10,Type11,Type12,Type13,Type14,Type15
00:00,1,0,15,6,0,0,0,0,0,1,0,0,0,0,0,0
00:00,2,0,4,1,0,0,0,0,0,0,0,0,0,0,0,0
00:00,3,0,3,2,0,0,0,0,0,0,0,0,0,0,0,0
00:00,4,0,15,6,0,1,0,0,0,0,0,0,0,0,0,0
01:00,1,0,11,1,0,0,0,0,0,0,0,0,0,0,0,0
01:00,2,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0
01:00,3,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0
01:00,4,0,8,7,0,1,0,0,0,1,0,0,0,0,0,0
02:00,1,0,7,1,0,0,0,0,0,2,0,0,0,0,0,0
02:00,2,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0
02:00,3,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0
02:00,4,0,8,4,0,1,0,0,0,1,0,0,0,0,0,0
03:00,1,0,3,4,0,0,0,0,0,0,0,0,0,0,0,0
03:00,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
03:00,3,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0
03:00,4,0,7,5,0,2,0,0,0,4,0,0,0,0,0,0
04:00,1,0,5,6,0,0,1,0,0,3,0,0,0,0,0,0
04:00,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
04:00,3,0,2,0,0,0,0,0,0,1,0,0,0,0,0,0
04:00,4,0,24,7,0,1,0,0,0,2,0,0,0,0,0,0
05:00,1,0,22,11,0,1,0,0,0,8,1,0,0,0,0,0
05:00,2,0,8,3,0,2,1,0,0,1,0,0,0,0,0,0
05:00,3,0,9,2,0,0,0,0,0,0,0,0,0,0,0,0
05:00,4,0,30,13,0,2,0,0,3,4,0,0,0,0,0,0
06:00,1,0,26,32,0,7,1,0,0,5,0,0,0,0,0,0
06:00,2,0,14,10,0,4,0,0,0,1,0,0,0,0,0,0
06:00,3,0,11,1,0,1,0,0,0,3,0,0,0,0,0,0
```

Figure 6: Classification data example

Standardized data formats for VC and WIM data shown above are used to build a VC/WIM data warehouse. Because all formats are a csv text format, it is easy to read and significantly reduces application development time. In preparation to efficiently retrieve the data, these csv formatted data files must be well organized, which is discussed in the next section.

2.2 Data Warehouse Organization

After standardized formats are established, it is important to organize the data using an efficient data structure. The objectives set forth in this project for the data structure include:

- Simple to understand and use
- Easy to manage
- Easy to distribute and share large amounts of data
- No cost in adopting the technology

- Fast and easy to retrieve the data
- Adaptable for changes in data collection device location
- Easy to develop data applications
- Robust

There are many ways of organizing the data. The method adopted in this project was a tree structure which is commonly used in file systems. This structure is proven robust and efficient, as demonstrated by many file systems in various operating systems. This structure has been also successfully used at UMD DC [3] for managing a large quantity of data (e.g., terabytes). Since file systems are used in all PCs, many maintenance utilities, such as file backup utilities and monitoring of status are readily available. Therefore, the data warehouse structure for this project was chosen as a tree structure in networked disk storage. A networked storage is required because the data must be shared.

2.2.1 The directory structures of VC and WIM

In a tree structure, the files are organized as directories and subdirectories. At the root of the data warehouse tree, VC and WIM directories are created as shown in Figure 7. Under each root, the data types are divided into three categories: “Processed”, “Raw”, and “Rawcsv”. The “Raw” directory houses all of the Mn/DOT’s raw binary data. The “Rawcsv” directory houses the standardized csv formatted files. In the “Processed” directory, processed data, which are produced by processing the standardized csv data, are stored. The directory structure shown in Figure 7 is used at UMD DC and a similar structure is created at Mn/DOT.

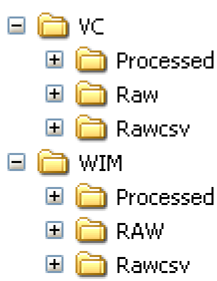


Figure 7: Root directory of VC and WIM

Inside the “Rawcsv” directory, the subdirectories are categorized by site IDs. For VC, six digit site ID numbers were used. Each site directories are automatically created when raw binary data is translated into the standard csv files. Within each site, the data is further divided into year directories. Each year directory contains daily files of that year. This relation is shown in Figure 8.

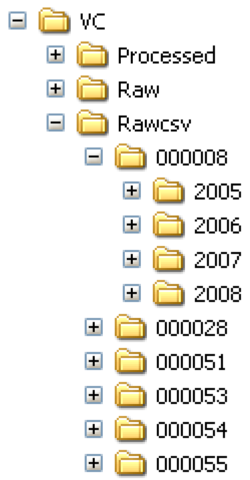


Figure 8: Each site directory of VC Rawcsv data is organized as year directories.

For WIM data, the sites directories incorporate the site names along with the site ID as shown in Figure 9. This was to improve readability of the directories but not necessary.

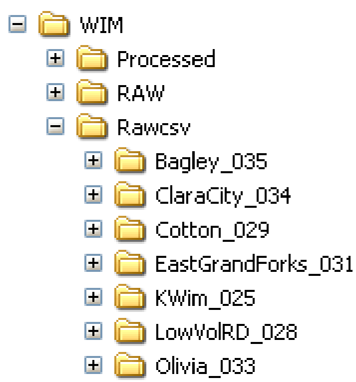


Figure 9: WIM site directory

The “Processed” directory contains different types of processed data which are produced as a result of processing the Rawcsv data. Figure 10 shows examples of “Processed” subdirectories, all of which are based on actual implementation as a part of this project. VC data is often used to create Monthly Adjustment Factors (MAF) for short count stations, which was developed as a part of applications in this project. MEPDG houses the pavement design input files that are processed using the csv WIM data. The details of these applications will be covered in Chapter 4. The “TMG” directory contains the monthly TMG reporting files.

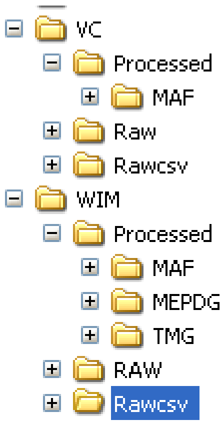


Figure 10: Processed directory example

2.2.2 Filenames

The leaves of the data tree in this project are the csv data files. Each csv data file contains data collected for a single day from the data collection site in a format defined in Section 2.1. The filename consists of three fields: date, site ID, and an extension, as shown below.

yyyymmdd.siteID.ext

yyyy: year
mm: month
dd: day
siteID: numeric site ID
ext: three letter extension. "vol" for volume, "cls" for classification, "spd" for speed, and csv for WIM.

With this filename convention, the date, the site ID, and the data type are immediately recognized from the filename. Also, data files from different sites can be mixed together and still be recognized. Examples of filenames are shown in Table 3.

Table 5: Filename Examples

Data Type	Filename Example	Description
Volume	20060423.000053.vol	Site 53, volume data for April 23, 2006
Speed	20071212.000100.spd	Site 100, speed data for Dec 12, 2007
Classification	20080120.000191.cls	Site 191, classification data for Jan 20, 2008
WIM	20080712.188.csv	Site 188, WIM data for July 12, 2008

CHAPTER 3: BINARY DATA TRANSLATION

3.1 Binary Data

There are a number of reasons that traffic data collection devices use proprietary binary formats. First, binary data formats create a compact format, and thus they reduce the usage of valuable memory space of the device. Second, binary formats are difficult to understand and can hide the format. Proprietary software, developed by the device company, is often required to read the binary data. Third, companies can change binary formats without acknowledging the users but by providing a new version of software. The hidden format changes provide a non-competitive advantage in providing new versions of the software. Fourth, the errors in the software are less exposed since other competing sources of software are not available. Therefore, binary data formats are often not disclosed.

For data sharing and application developments, proprietary binary formats are less desirable in several respects. When transportation departments use multiple data collection devices produced by separate vendors, different software packages must be purchased and used for producing the same type of data. This increases the cost, but also presents problems of non-uniform reports or even unavailability of a certain functions in different software packages. Also, when the quality of software is not satisfactory, there is no alternative and departments have to use the same unsatisfactory product. Different versions of a device within the same product line often require a different software package, resulting in inefficiency of developing data applications, such as an analysis of historic trends. Therefore, a standard binary format for data collection devices is more desirable, which is the case in communication and computer industry. However, such industry standards have not been created in traffic data collection devices.

For data warehousing, creating a standard format is imperative to allow data applications to freely query and share the data. This is the reason that this project created standard data formats. It adopted the most widely accepted character encoding standard and the data format, i.e., ASCII character set and csv data format. It should be noted that Mn/DOT has the ownership of all data collected from the devices that are owned by Mn/DOT. However, the data formats are proprietary and owned by the device companies. This means that Mn/DOT has the right to use the data without knowing the data format. This is the problem faced in this project. One solution would be creating a non-disclosure agreement for a specific Mn/DOT use with each and every company for the binary data format. Such an agreement could be legally complicated and may take a long time. Also, the usage limits placed on the non-disclosure agreement may create additional problems. Therefore, a different route was taken in this project. Since Mn/DOT owns the binary data, what to do with it is the right of Mn/DOT. If the binary data is read by Mn/DOT's own software at its own risk, there are no ownership issues for using the binary data. After a few meetings with Mn/DOT and the PI of this project, Mn/DOT granted use of the binary data by UMD DC for translating them to a standard ASCII csv format, as a part of this project. This chapter describes decoding of the binary formats into understandable information on all of the formats used by the present Mn/DOT WIM and VC data collection devices. The formats were decoded by a pattern analysis of binary bits, so the absolute accuracy is not guaranteed. However, the accuracy of data translation was verified against the output of each company's software package that Mn/DOT purchased.

3.2 WIM Binary Data

Mn/DOT presently uses four different types of WIM systems, which are IRD 1068, IRD iSync, PEEK PVR, and Bulldog. The Bulldog system was developed by UMD and directly outputs ASCII csv data, so no translation is required. Therefore, the three binary formats of the IRD 1068, IRD iSync, and PVR are described. The basic unit of binary is a byte which consists of 8-bits. The formats are described byte-by-byte in the order it appears in the binary file.

3.2.1 IRD 1068 format

IRD 1068 was developed using a DOS based embedded PC. Since DOS does not allow multitasking, it runs on a batch mode and can only perform one task at a time. For example, while the data is downloaded, the data collection function must stop. It also uses the old 32-bit DOS timestamp, which starts from Jan 1st, 1970. The binary data consists of a collection of vehicle records, each of which are composed of a timestamp, error number, lane number, record type, speed, number of axles, axle spacing, and axle weights. The data follows the byte order of “Little Endian” which stores the least significant byte at the lowest address. The data format is summarized in Table 6 in the byte order that appears in each file.

Table 6: IRD 1068 Binary Format

Bytes	Description
16 bytes	**Header information**: It represents system type, number of lanes, etc. An example byte pattern is1,8,4,0,0,0,0,0 (or 264193 in integer) or 0,0,0,0,0,0,0,0.
3 bytes	**Index or count** of the vehicle record
4 bytes	**Timestamp** It is a 32-bit integer representing, number of seconds since Jan 1st 1970, 00:00:00 GMT. Hence, direct translation of this timestamp is GMT, and the local time must be obtained by subtracting a proper number of hours. In general, **8 hours** should be subtracted to get the local time because the DOS timestamp routine was originally written in California and uses the PST.
1 byte	**Error number**
1 byte	**Lane number**. This number starts from 0, so the lane number is obtained by adding 1. For example, 2 indicates lane #3.
1 byte	**Status.** This field actually represents error/warning numbers specified in the DOS Office program. If status =0, it means the data is good. If status>=15, it is a warning. If 0<status<15, it is a critical error and the next record starts right after this status byte.

1 byte	**Record Type** 10 = classification only 11 = normal axle weight record, i.e., space 1, space2, weight1, weight2, weight3, … order 12 = split weights record, i.e., space1, space2, …,weight1_left, weight1_right, … 30-34=calibration mode 41-44=diagnostic mode
2 bytes	**Vehicle length in cm.** This is not useful data since the length is not measured but estimated based on axle spacing.
1 byte	**Speed.** Km/hour
1 byte	**Number of axles.** This number determines the subsequent number of data items.
Array of 2 bytes (array of 16-bit integers)	**Axle Spacing in Cm**. (2 bytes per each spacing) The number of bytes is (#_of_axles -1) * 2 bytes
Array of 2 bytes (array of 16-bit integers)	**Weights in Kg** (2 bytes per each axle weight) Data size and order depends on the record types. type 11: #_of_axles * 2 bytes type 12: #_of_axles * 4 bytes (left and right per each axle) type 41, 42, 43, 44: calibration mode NumOfWeightRecords = type - 40 #_of_axles * NumOfWeightRecords * 2 bytes type 30,31,32,33,34: diagnostic mode NumOfWeightRecords = type -30 #_of_axles * NumOfWeightRecords * 2 bytes type 20 : it is a special type of record. This record always shows up before the diagnostic records. If the data byte at the number of axles indicates how many words must advance, e.g. if #_of_axles=12, 12 words must advance and then next record starts. Note that no spacing data appears in this case.
2 bytes	**End of record mark** Most of time, it is 0000.

3.2.2 iSinc format

This format is used by a newer type of WIM system provided by IRD (International Road Dynamics Inc). However, it still uses the old DOS time stamp. The binary ordering follows the "Little Endian" order. The header is clearly different from IRD 1068, but similarities exist in the rest of data. There are several unidentified fields but the critical information such as axle spacing and weights were identified. The summary of binary format is shown in Table 7 in the order of the bytes that appear in a file.

Table 7: IRD iSinc Format

Bytes	Description
16 bytes	**Header information**: System type, number of lanes, etc… Typically, it is 1,10,255,0,0,0,0,75 (or 5404319552861309441 in Int64) or 0,0,0,0,0,0,0,0
2 bytes	**Index or counter** of the vehicle record
1 byte	**Lane number**. This number starts from 0, so the lane number is obtained by adding 1. Example) 2 indicates lane #3.
7 bytes	**Unknown**. It appears the 6th of this 7 bytes looks like a status.
4 bytes	**Timestamp.** It is a 32bit integer representing, number of seconds since Jan 1st 1970, 00:00:00 GMT. Hence, direct translation of this timestamp is GMT, and the local time must be obtained by subtracting a proper number of hours. In general, you **must subtract 8 hours** to get the local time because the DOS timestamp routine was originally written in California.
1 byte	**Error number**. This is a guess and not clearly identified.
3 bytes	**Unknown**
1 byte	**Record type.** 10 = classification only, contains only axle spaces. Data to read=(AxleNum -1)*2 11 = normal axle weight record, i.e., space 1, space2, weight1, weight2, weight3, … order 12 = split weights record, i.e., space1, space2, weight1_left, weight1_right, …

	weight2_left, weight2_right, … weight3_left, weight3_right, … 30-34=calibration mode 41-44=diagnostic mode
1 byte	**Speed.** Km/hour
2 bytes	**Vehicle length in cm.** This is not useful data since the length is not measured but estimated based on axle spacing.
4 bytes	**Unknown**
1 byte	**Number of axles.** This number determines the subsequent number of data items.
Array of 2 bytes (int16)	**Axle Spacing in Cm**. The size of this field is (# of axle -1) * 2 bytes.
Array of 2 bytes (int16)	**Weights in Kg** It depends on type 11 or 12. If the record type is 11, each axle weight is recorded using one 2-byte word. If the record type is 12, it is a split axle record and recorded using two 2-byte words.
25 bytes	**Unknown** It appears as an External data tag and information field. The size is computed by, 13 + (#_of_axles * 6) bytes
2 bytes	**End of record mark** Usually, it is 0x0000.

3.2.3 PVR format

Another type of WIM system used by Mn/DOT is provided by PEEK, and the binary data is formatted using PVR (Peek Vehicle Record) format. This format is completely different from the IRD 1068 or iSinc and much more complex. Each vehicle record includes timestamp, number of axles, speed, vehicle class, axle spacing, axle weights, and error code. Due to the complexity, this format is shown in Appendix-A.

3.3 VC Binary Data

This format is designed to handle several different configurations. It can be configured for a single loop ([]), two piezoelectric sensors (| |), two loops ([] []), or two piezoelectric sensors with a loop (| [] |). The data type can be volume, speed, classification, or a combination of them. In the header, it also defines a flow assignment which is basically a lane assignment. The data collection methods are defined by three studies (Study 1, Study 2, and Study 3). Each study consists of three headings, and the heading can be empty. For example, if Study 1 is to collect volume and speed data, the heading 1 defines the volume definition; heading 2 defines the speed definition; and heading 3 is left empty (i.e. 0's). The data is recorded not as vehicle records but as hourly counts in a bin. For example, classification data is recorded as 15 bins that contain counts of 15 FHWA vehicle types in that hour period. The bins are sometimes allocated for each lane or sometimes combined for all lanes. Again this format is fairly complicated, and thus the details are attached in Appendix-A.

3.4 Conversion Software

For converting binary data to the standard csv formats, two software packages were developed in this project. The first one is called "Bull Converter" which was developed as a converter for all of Mn/DOT's WIM binary data. A screen capture of this software utility is shown in Figure 11. This software converts IRD 1068 and iSinc binary data files into the standard csv WIM format defined in Section 2.1. The software identifies the format differences of 1068 and iSinc from the header file and automatically translates them into the standard WIM csv format. It also recognizes PVR binary format and converts them into the standard WIM csv format. The conversion is done file-by-file and it has an option that only new files are converted. The conversion can be done for multiple years, from a start year to current, or month by month. As each file is converted, the Bull Converter organizes them into the directory structure described in Section 2.2. This organization can be defined through the "Site Name Edit" tab. It also provides a utility to read the log file that contains conversion status information of all converted files, such as conversion time, errors during the conversion, and the original binary file information. Another useful utility included is that the user can look at the header and other details of a binary file. If a conversion error was noted from the log information, this utility can be used to track down the cause of the error. During the conversion process, the Bull Converter also computes vehicle type using a class definition file. This utility was added in order to apply the same classification scheme to all WIM data, regardless of the differences in WIM devices (i.e. manufactured by different companies).

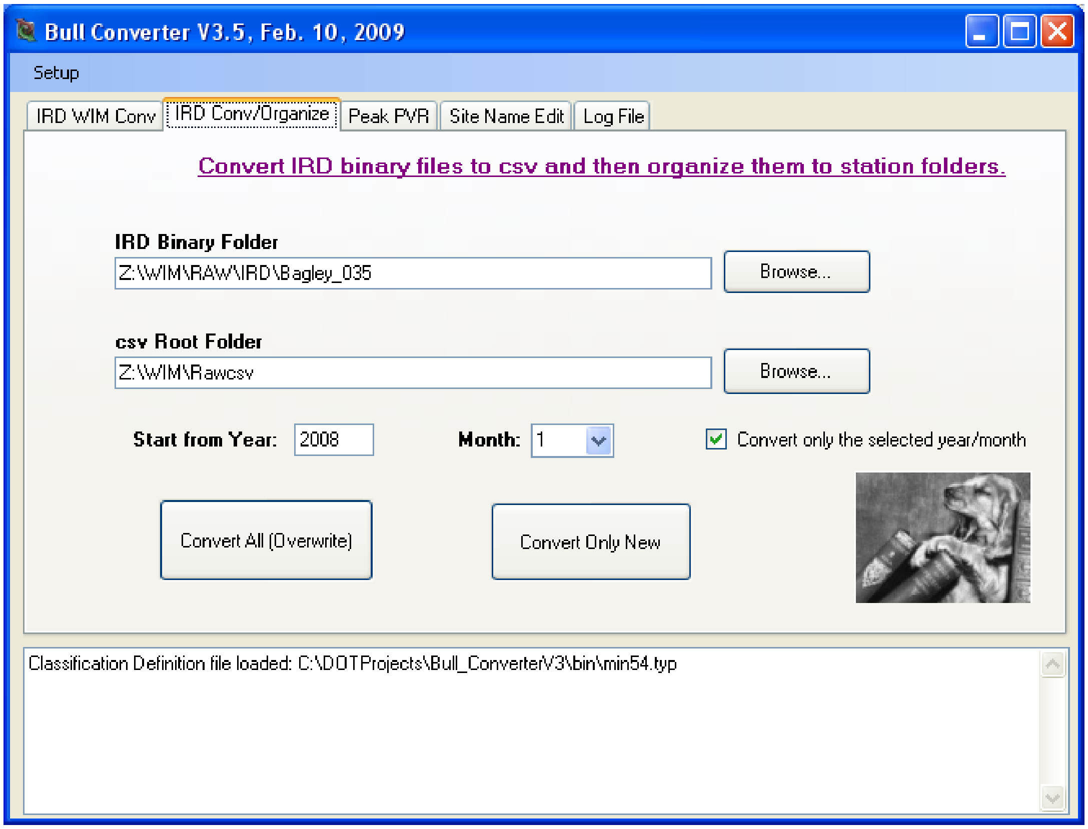


Figure 11: A screen capture of Bull Converter, which converts all of Mn/DOT binary WIM data to a csv WIM format.

The second software package developed is called "Peak Binary Converter" and can convert all of the present and past Mn/DOT's VC data into the standard VC csv format defined in Section 2.2. A screen capture of this software is shown in Figure 12. The conversion utilities are similar to the Bull Converter. The binary data can be converted site-by-site or all of the sites at once. Also, the data can be converted from a starting year to current, specific year, or specific month. The converted files are automatically organized into the directory structure described in Section 2.2.

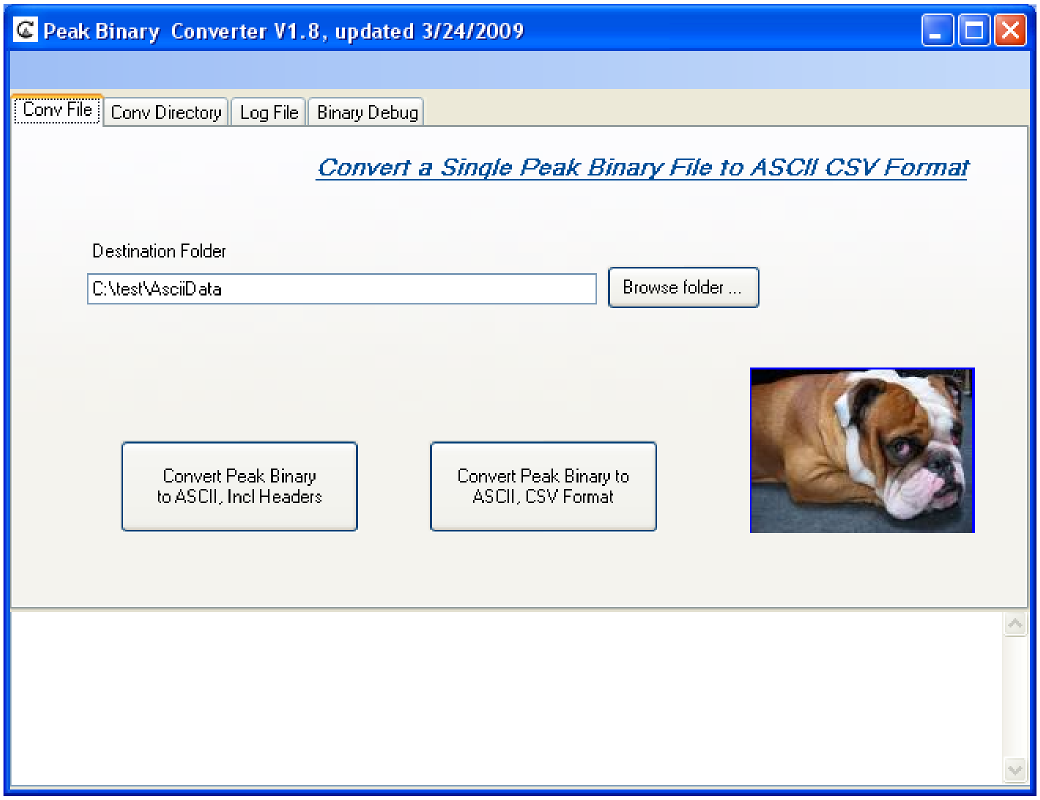


Figure 12: A Screen capture of Peak Binary Converter

CHAPTER 4: DEVELOPMENT OF DATA REPORTING APPLICATIONS

4.1 Application Development

After creating the data warehouse as a part of the network storage at Mn/DOT, several applications that utilize the data in the data warehouse were developed for Mn/DOT. The applications introduced in this project were devised based on the present and future needs of Mn/DOT. The accuracy of the final outputs of the applications was extensively tested by comparing them to the outputs of exiting proprietary software as well as using the past records. Each application was developed as a complete software package that can be installed on any PC using a standard Windows Installer. Only requirement is that the PC must have an access to the data warehouse structure described in Chapter 2. The three software packages developed in this project are BullReport, BullPiezo, and BullGuide. Because all of the binary data was already converted into the standard csv formats, the main advantage of using these software packages is that all of them seamlessly work on the data produced by the data collection devices that are manufactured by different companies or different models within the same company. Much effort has been made to ensure that the quality, performance, and reliability of the software are equivalent or better than the commercial grade software packages supplied by the device manufacturer. Some of the functionalities that are not available from commercial software but needed for the operation of MN/DOT TDA were carefully developed and tested for accuracy. Special attention was paid to the software design, to ensure the software was easy to use and that minimum training is required for using the software. The user interface was mostly implemented using visual components, and the software can run mostly using mouse clicks, i.e., no command lines are used. The capability and functionalities of each package are described in the subsequent sections.

4.2 BullReport

4.2.1 Introduction

BullReport (a short name of Bulldog Report) was developed as an analysis tool for the WIM data, as well as for a data summary reporting tool. Its functions include load spectra analysis of single, tandem, tridem, quadem, steer, and GVW (Gross Vehicle Weight), TMG reporting and analysis tools, and data summary and analysis reporting tools. It consists of four tabs: Station Def, Load Spectrum, TMG Reporting, and Summary Reports. The first tab of this software is shown in Figure 13.

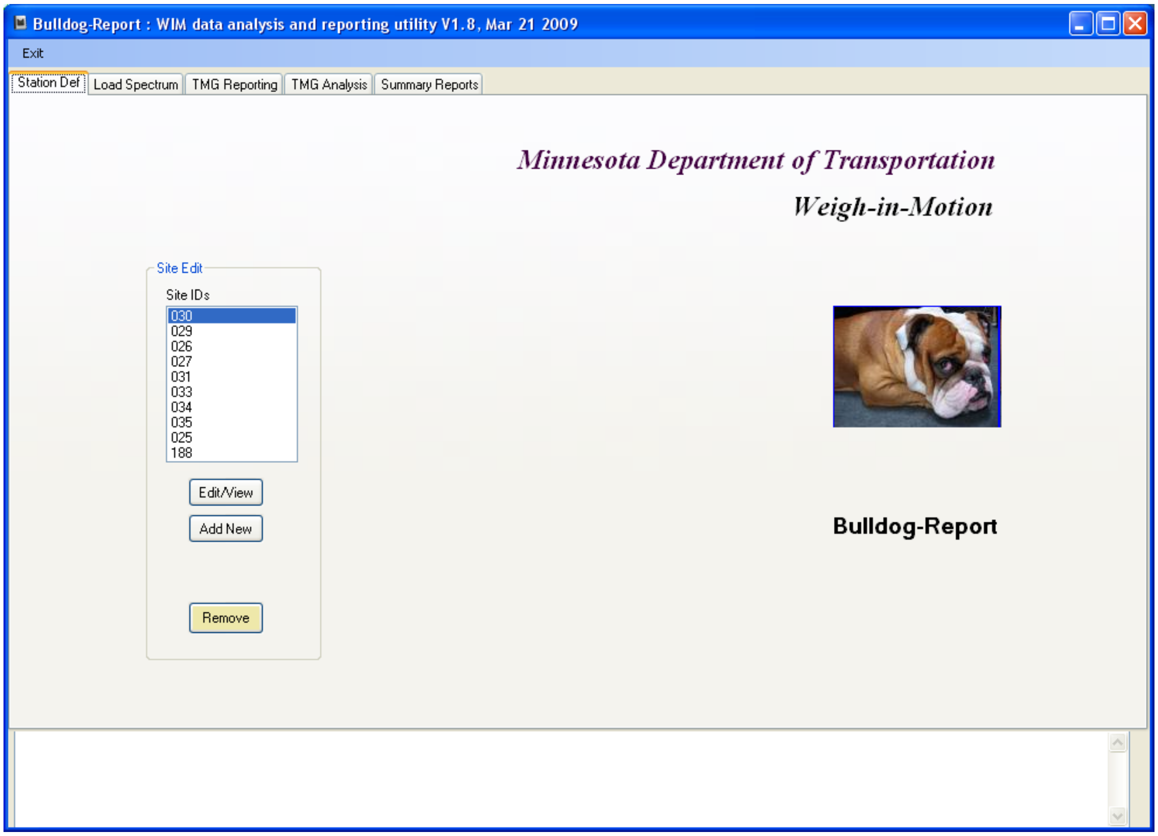


Figure 13: BullReport user interface

4.2.2 Station definition

As the first step of using this software, stations must be defined using the Station Def Tab. Station information include site ID, site name, and lane descriptions and can be added, deleted, or edited. The site information is managed using the buttons and lists shown in the Site Edit group. Doubling clicking the site ID in the Site IDs list box or the Add New button pops up a Lane Edit window at which the user can add or edit the site information. Lanes can be defined up to eight and must specify the directions. The standard TMG-2001 direction code is used, i.e., 1=North, 2=Northeast, 3=East, 4=Southeast, 5=South, 6=Southwest, 7=West, and 8=Northwest. Clicking the Save button saves the all entries. The lane numbers should be check marked, starting from 1 and then consecutively marking without skipping the numbers. The site information needs to be defined only once.

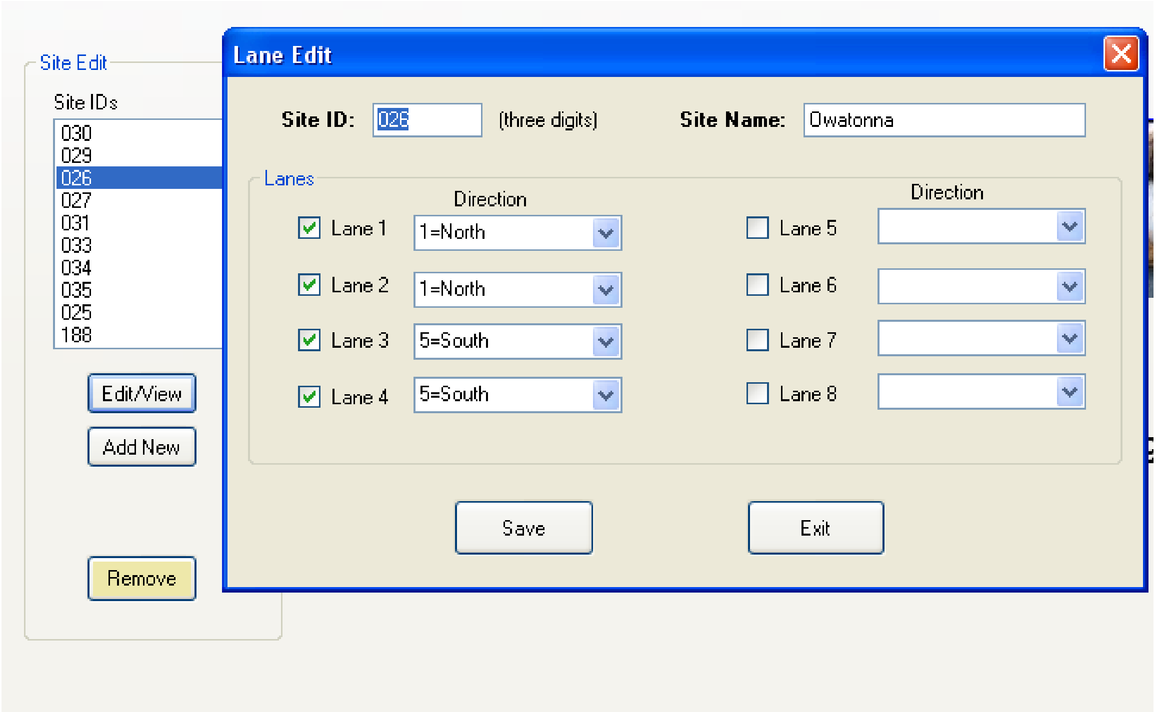


Figure 14: Site Edit group and Lane Edit window

4.2.3 Load Spectra

Load spectrum is a histogram of axle group loads and has become increasingly an important tool for the analysis of pavement maintenance and design [5]. It is also a useful tool for a diagnostic analysis of the data. For example, Figure 15 shows a screen capture of a tandem axle load spectra of types 9 and 16 (9 in FHWA scheme). It shows an overlap of two Gaussian curves with different means, one at 10 Kips and the other at 30 Kips. It is known that tandem axles of type-9 vehicles have a stable distribution as shown in Figure 15 [8]. This distribution property can be used for calibration or diagnostics of a WIM system.

For using this functionality, the user can select the period, vehicle classes, axle groups, bin sizes, and bin ranges. In addition, different types of classification schemes can be chosen. The load spectra data can also be exported as csv text file and can be read using a text editor, such as a Notepad. In addition to the normal axle groups, the load spectra of steer and GVW can also be plotted.

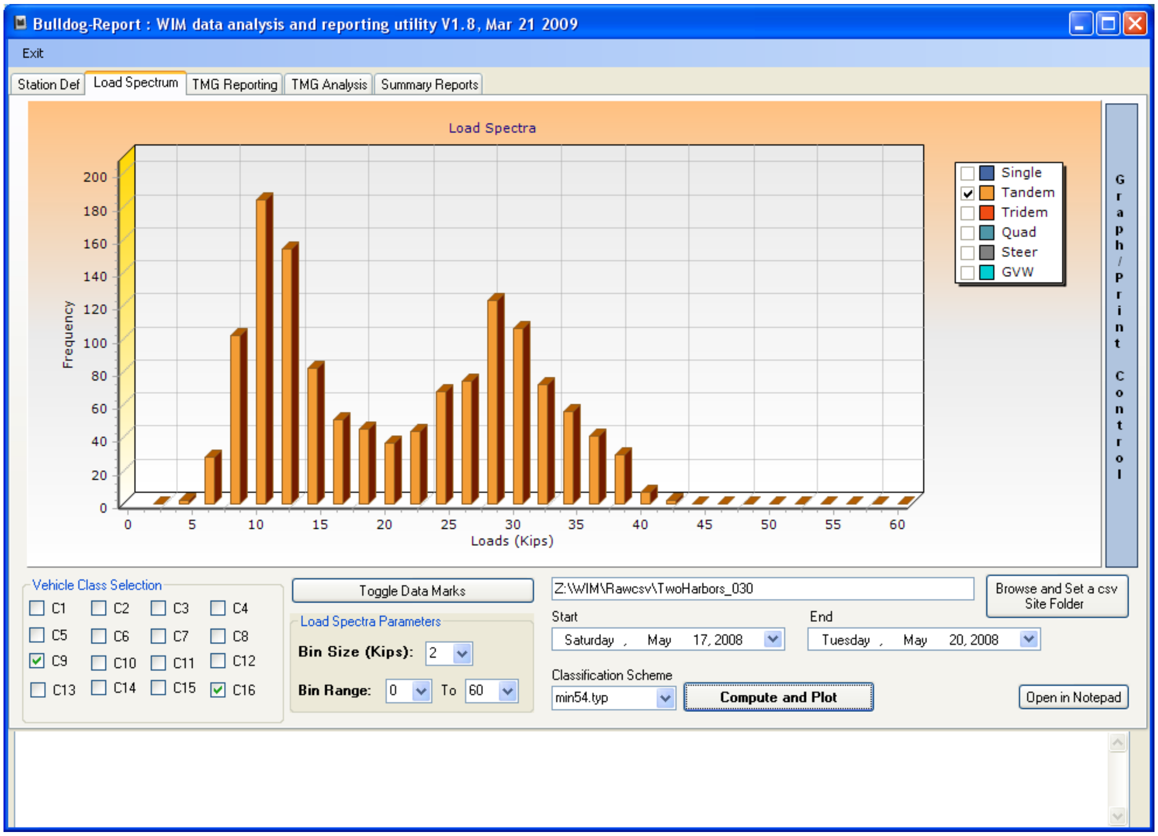


Figure 15: A screen capture of the BullReport's Load Spectrum tab

4.2.4 TMG reporting

FHWA collects a nation-wide WIM data using the data format defined in Section 6 (Traffic Monitoring Data Formats) of TMG-2001 [4]. Mn/DOT participates in this program and submits the data monthly for all WIM sites. The TMG-2001 format defines four different types of data formats, which are station description data, traffic volume data, vehicle classification data, and truck weight data [4]. The TMG Reporting tab in the BullReport provides utilities for producing these four types of data using the csv WIM data. The screens of this utility are shown in Figure 16-18.

The volume, classification, and truck weight reports can be produced for a single day or for single or multiple months. Also, all three types of data reports can be produced at once by using the "All Three" button. The station description data is produced by the "Build a TMG Station Description Report" button, which opens a window shown in Figure 17 and 18. It consists of two tabs with mostly user selectable items. Pre-existing station description files can be loaded, modified, and saved as a new file, or a new file can be easily created. The sample TMG-2001 data files produced by the BullReport are attached in Appendix-B.

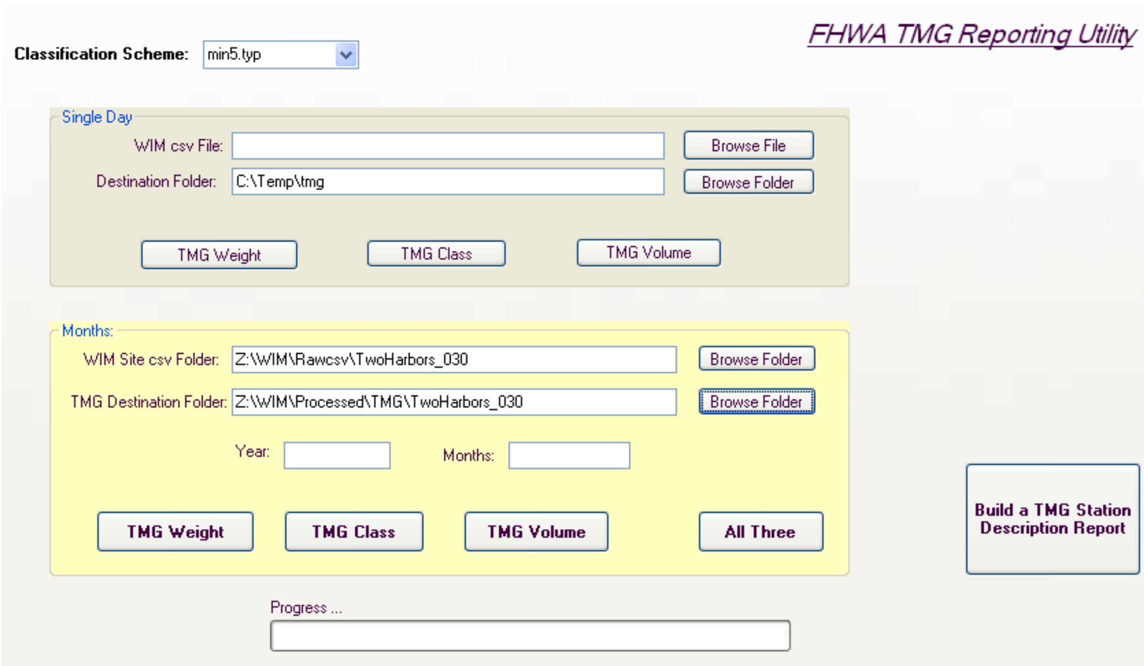


Figure 16: Screen capture of the BullReport TMG reporting utility

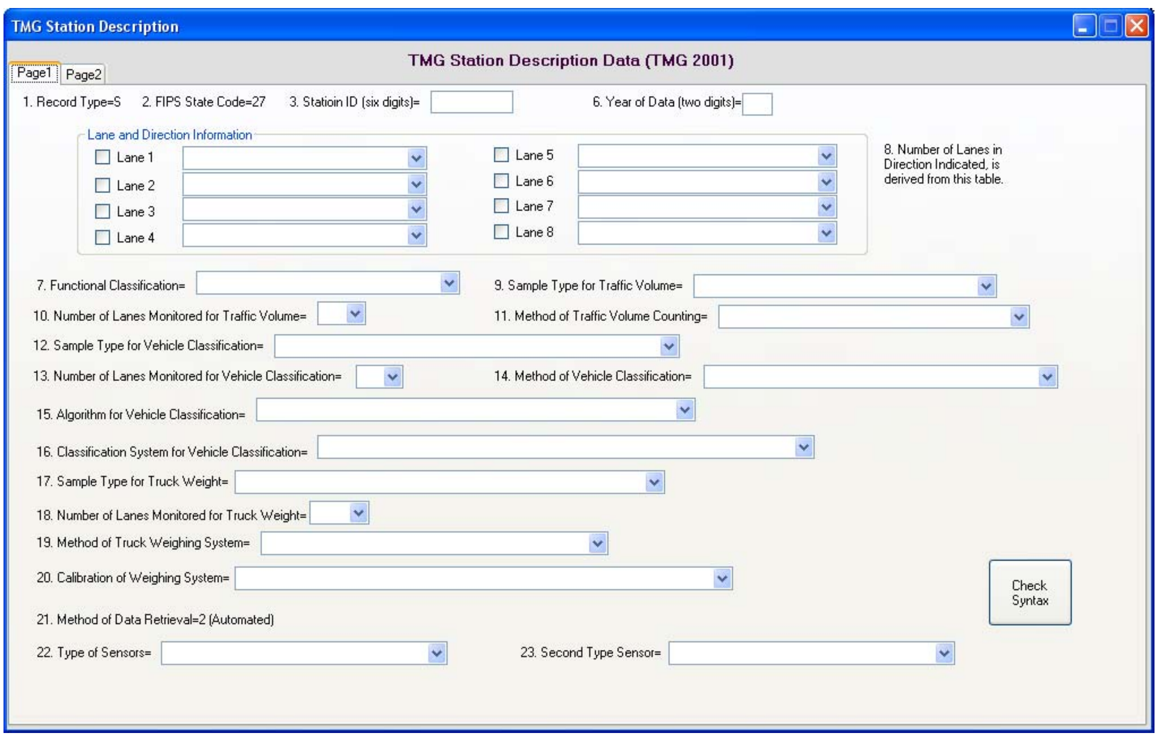


Figure 17: TMG station description utility page 1

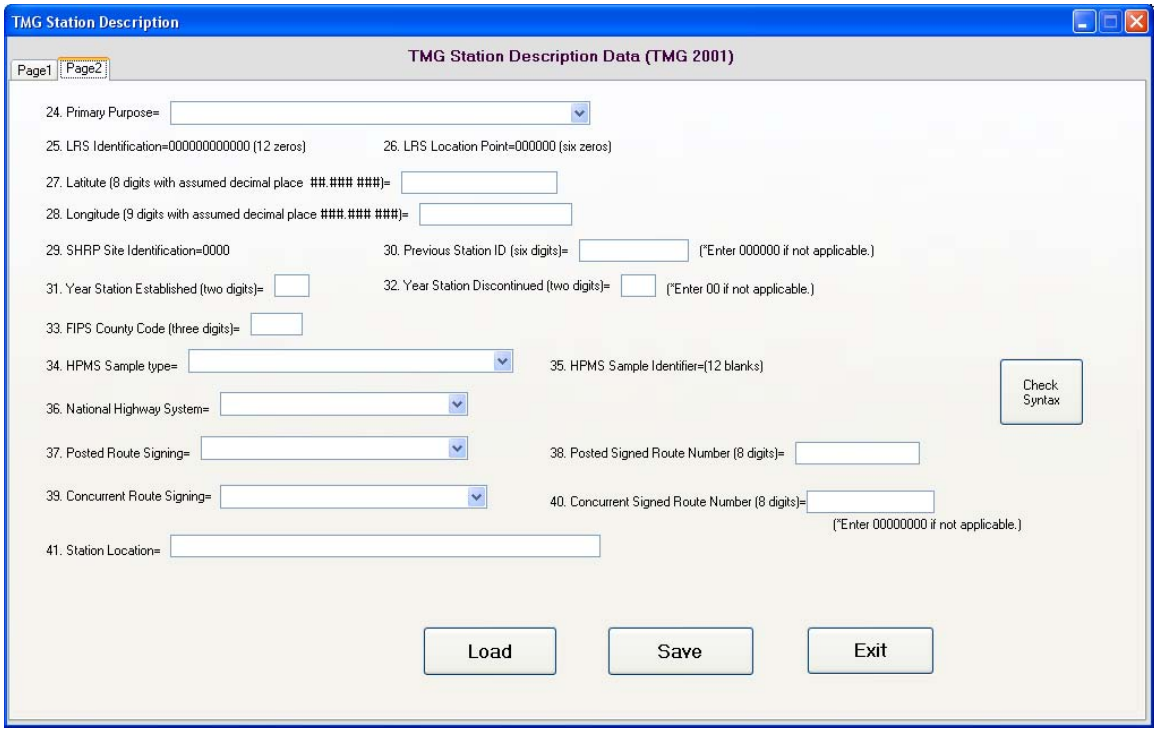


Figure 18: TMG station description utility page 2

4.2.5 TMG-2001 data analysis

Once TMG-2001 formatted data is produced, it can be used as inputs to other analysis tools and software packages. The TMG Analysis tab shown in Figure 19 can load and visualize TMG-2001 volume and classification files through several types of plots. This tool is useful when the format needs to be verified. If the format of the data is correct, it loads successfully, but it produces error messages if the data format has an error. Due to inter-operability, TMG-2001 formatted data produced by other software can be loaded and used for analysis.

The plot utility included can plot any TMG-2001 classification data by each lane, all lanes, or selected lanes. In additions, it can plot any selected date or the whole month. For TMG-2001 volume data, the daily volume can be plotted by hour or the whole month. Figure 20 shows an example plot of TMG volume data. In addition to plots, this utility computes and displays daily volume or monthly total volume, depending on the data selected. The users can modify much of the display options in the “Graph/Print Control” button and that it works similar to the graphing options in Excel.

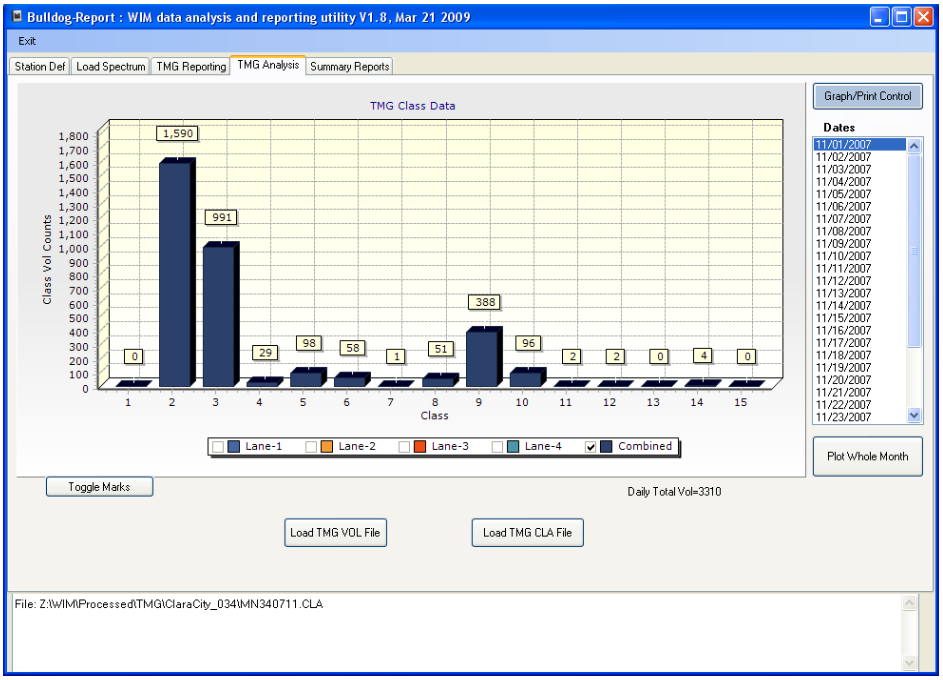


Figure 19: TMG analysis tab

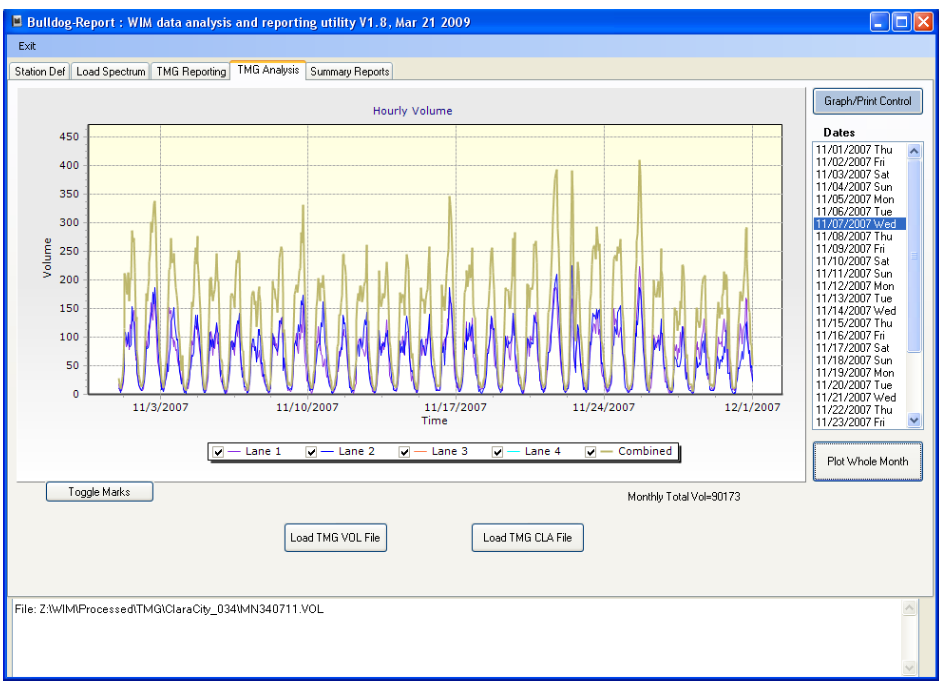


Figure 20: An example plot of TMG volume data

4.2.6 Summary reports

Utilizing the csv WIM data, the BullReport can generate many different types of data reports. It presently can generate twenty one different types of reports, and each report can be formatted in four different formats, which are text, Excel, pdf, and XPS (web format). The reporting tool was implemented in the Summary Reports tab and a sample screen is shown in Figure 21. As a report type is selected, different types of user selectable parameters are displayed in the "Select Reporting Parameters" group. In Figure 21, "Load Spectrum" was selected from the "Report Type" combo-box, which brings up the user selectable parameters: the period defined by the "Start" and "End" calendar box, "Classification Scheme" combo-box, "Lanes" check boxes, bin size, and vehicle types. The report is created when the user presses the "Create Report" button located at the lower right corner of the window.

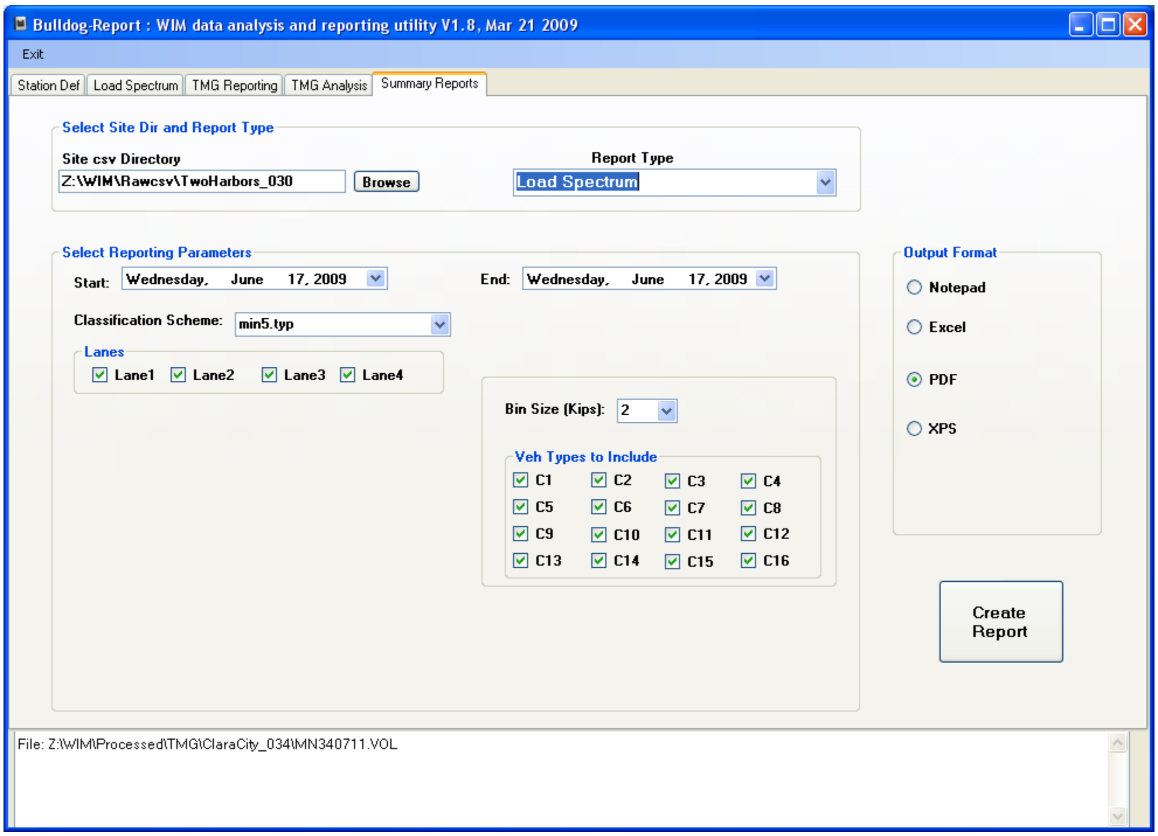


Figure 21: Summary report tab

Each report type is briefly described next. A sample PDF file for each type of report is attached in Appendix-B.

Class by Hour Report

This report shows volume distribution of each type of vehicle per each hour of the selected period. The data table consists of 16 columns representing vehicle Types 1-16 (type-16 is used at Mn/DOT) and 24 rows for 24 hours. The table also includes a total and percent for each column. The percent indicates the percent distribution of the vehicle type during the period and the lane(s) selected. The user has options to select a single or multiple lanes, start and end dates of the reporting period, and the classification scheme.

Speed by Hour Report

This report shows the number of vehicles at a certain range of speeds per each hour. The speed bins can be customized using the number of bins, the first bin range, and bin size. The table consists of 24 rows for each hour of day and the number of columns for the speed bins. Using this report, one can quickly identify how many vehicles exceeded speed limits at what hour. The user can select a particular lane or lanes and the period of the data.

Lane by Hour Report

This report summarizes lane-by-lane volumes by hour. It consists of number of columns for the selected lanes and 24 rows for each hour. The table also includes the total volume and percent of each lane and hourly total volumes for all lanes. Using this table, utilization rate of each lane can be observed.

Lane by Class Report

This report shows lane volume distribution by class. Large trucks more frequently use the driving lanes, which can be observed from this table. This table consists of rows corresponding to the lanes and columns corresponding to the number of vehicle types used in the classification.

Lane by Error Report

This report summarizes error frequencies by lane and error type. Error codes used are 101-113 and 14-39, totaling 39 codes. Code 0 denotes no error case. The table consists of rows corresponding to the lanes and columns corresponding to error codes. The meaning of each error code is provided under the table.

Error Vehicle by Hour Report

This report summarizes frequencies of errors by hour of the day and error type. The error codes used are same as the Lane by Error Report. This table can be used to identify which hour had most errors.

Class by Day of Month Report

This report summarizes each class volume by day-of-week. It includes total, percent, and average volumes in additional rows. The columns correspond to the number of vehicle types defined in the classification scheme, and the rows correspond to day of month.

Truck Count by Day of Month Report

This report is identical to the Class by Day of Month report, except that it only includes trucks. Vehicle types 4 and above are considered as trucks.

Class by Front Axle Weight Report

Front axle weights provide useful diagnostic information. This report consists of weight bin rows and vehicle type columns, by which it shows the front axle weight distribution for each vehicle type. The user can select the weight bin size, classification scheme, and the lanes to be included in the report.

Class by Gross Vehicle Weight Report

This report is identical to the "Class by Front Axle Weight Report," except that the weight distribution is computed using gross vehicle weights (GVW).

Weight Violation by Class Report

This report summarizes over-weight violations by class, axle group (single, tandem, tridem, and quadem), GVW, and bridge weight. The user can specify the weight limit of each violation type, using the parameters shown in Figure 22. Bridge formula weights follow the FHWA guideline published in Aug 2006, ref [9].

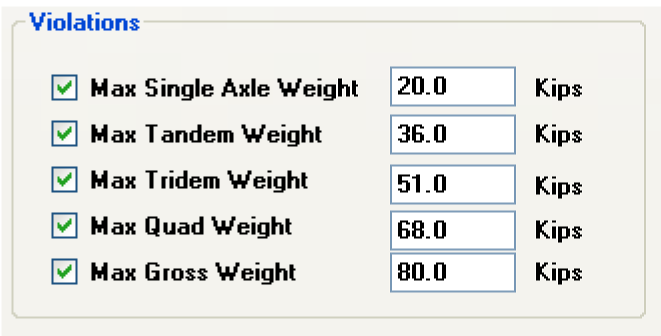


Figure 22: User entry screen of weight limit

Weight Violation by Hour Report

This report is similar to the Weight Violation by Class Report. The difference is that weight violations are shown by hour, instead of class.

18Kip ESALs by Hour Report

This report summarizes equivalent single axle loads (ESALs) by hour and class. ESALs are computed according to the user selection of pavement type (flexible or rigid). For flexible pavements, SN (Structure Number) = 5 and Pt = 2.5 are used as the default values. For rigid pavements, D (pavement thickness) = 10" and Pt = 2.5 are used as the default values. This report only includes trucks.

Axle Count by Axle Weight Report

This report summarizes axle counts, total weights, and ESALs for each weight range in kips. In addition, totals of axle counts, weights, and ESALS are shown at the last row. The user can select the bin size by defining the weight ranges in kips. The default bin size is 2 kips.

Single Axle Load Spectra by Class Report

This report summarizes load distribution of single axles for each vehicle type.

Tandem Axle Load Spectra by Class Report

This report summarizes load distribution of tandem axles for each vehicle type.

Tridem Axle Load Spectra by Class Report

This report summarizes load distribution of tridem axles for each vehicle type.

Quadem Axle Load Spectra by Class Report

This report summarizes load distribution of quadem axles for each vehicle type.

Load Spectra Report

This report summarizes load spectra by single, tandem, tridem, and quadem axles, GVW, and steer axles. The user selectable parameters include the reporting period, lanes, classification scheme, bin size, and vehicle types to include in the load spectra.

Speed by Class Report

This report summarizes the speed distribution of each vehicle type. The user selectable parameters include the speed bin size, number of bins, and the first bin range. Using this report, the user may check the vehicle distribution of average and/or violated speeds for each vehicle type.

Site Summary Report

This report provides a quick summary of several key data for the selected WIM site and period. The types of data summary include vehicle counts and percent, error counts and percent, valid weights, total GVW, average GVW, total ESAL, overweight vehicle counts and percent, and average front axle weight, all of which are tabulated for each vehicle type.

In summary, BullReport can generate a total, 21 different types of formal reports in pdf, text, excel, and web formats. Each report type has its own user selectable parameters such as bin size and can be generated for any time period. It also includes a utility that can generate TMG 2001 formatted station description, traffic volume data, vehicle classification data, and truck weight data. Several graphic utilities are provided for load spectra and TMG 2001 formatted data analysis. These versatile and flexible tools are developed to meet the present and future needs of WIM data analysis, summary, and reports.

4.3 BullGuide

4.3.1 Introduction

In the late 1950s, a series of road tests called the American Association for State Highway Officials (AASHO) Road Test was carried out by the American Association of State Highway and Transportation Officials (AASHTO). This study is frequently used as a primary source of experimental data when vehicle damage to highways is considered for the purposes of road design, vehicle taxation, and costing. One of the significant findings of this study was the concept of load equivalency factors. This concept is used to simplify a complex stream of traffic information into a single design parameter --- the number of equivalent single axle loads (ESALs).

ESALs are still used in many design procedures today but the validity of using a single parameter has been questioned over time. NCHRP Project 1-37A, which was completed in 2004, eventually established the mechanistic-empirical pavement design guide (MEPDG) which utilizes both mechanistic principles and empirical performance observations [5]. The final product was a software package that is known as the MEPDG software [6]. The number of inputs is significantly greater than the traditional ESAL based design, which makes preparation for MEPDG more complicated. NCHRP Project 1-39 was later established and produced a software package called the TrafLoad for collecting and processing traffic data for the MEPDG software [7].

The BullGuide software package was developed to generate MEPDG input data for Mn/DOT's WIM data. It generates the same export output files as those produced by the TrafLoad, but the procedure is much simpler and easier to use, due in part to the customization to Mn/DOT data and a simplified user interface. The BullGuide directly reads in Mn/DOT's csv WIM data and generates MEPDG inputs with just few mouse clicks. TrafLoad, on the other hand, uses TMG formatted data and requires manual entries of pre-processed data. The BullGuide typically takes only a fraction of time to produce MEPDG inputs, in comparison to TrafLoad.

4.3.2 MEPDG input computation

The inputs required for the MEPDG software are AADT for vehicle classes 4-13, Monthly Distribution Factors (MDF) for classes 4-13 for each month, Hourly Distribution Factors (HDF) for classes 4-13 for each hour, Axle Group per Vehicle (AGPV) for each class, and load spectra for single, tandem, tridem, and quadem axle groups. Among them, load spectra computation is the most complicated, and the algorithm used is described.
From each daily WIM data, a load spectra (denoted LS_1day() and a three dimensional array of bins) is first constructed as:

$$LS\_1day(i, j, k) \tag{1}$$

where i is the vehicle classes 1 through 13, j is the axle groups 1 through 4, and k is the load range index 1 through 39. The load range index is determined according to Table 8. For each month, a daily array is organized as day-of-week (DOW), i.e.,

$$LS\_DOW(w, d, i, j, k) \tag{2}$$

where w is the week index for the month (i.e., wth week of the month) and d is an index of DOW (0=Sunday, 1=Monday, …, 6=Saturday). Next, average DOW load spectra are computed for each month m, i.e.,

$$LS\_DOW\_Avg(m, d, i, j, k) \tag{3}$$

By averaging the seven days of *LS_DOW*_Avg, monthly load spectra for class i, axle group j, load range k, and for the month m are computed as,

$$LS\_Month(m, i, j, k) \tag{4}$$

The array *LS_Month()* is used to construct the load spectra input file for the MEPDG software. Similarly, volume data is used to construct AADT for each class. That is, volume data is organized as average DOW for each month and class. Monthly AADT is then computed by averaging the average DOW volume for the month for each class. The final AADT is the average of each month for the year for each class.

If multiple years of data are used for data preparation, an exponential average is computed to put more weights on the most recent year, i.e., $\bar{x}_y = 0.5x_y + 0.5\bar{x}_{y-1}$ where x_y denotes the current year data and $\bar{x}_{y-1}$ denotes an exponential average up to one year past data.

Table 8: Load Range Index for Load Spectra

Load Range	Upper Limit of Load Range (kips) by Type of Axle group			
	Single	Tandem	Tridem	Quadem
1	3	6	12	12
2	4	8	15	15
3	5	10	18	18
4	6	12	21	21
5	7	14	24	24
6	8	16	27	27
7	9	18	30	30
8	10	20	33	33
9	11	22	36	36
10	12	24	39	39
11	13	26	42	42
12	14	28	45	45
13	15	30	48	48
14	16	32	51	51
15	17	34	54	54
16	18	36	57	57
17	19	38	60	60
18	20	40	63	63
19	21	42	66	66
20	22	44	69	69
21	23	46	72	72
22	24	48	75	75
23	25	50	78	78
24	26	52	81	81
25	27	54	84	84
26	28	56	87	87
27	29	58	90	90
28	30	60	93	93
29	31	62	96	96
30	32	64	99	99
31	33	66	102	102
32	34	68		
33	35	70		
34	36	72		
35	37	74		
36	38	76		
37	39	78		
38	40	80		
39	41	82		

4.3.3 BullGuide output files

BullGuide produces nine different files in a directory named using the site ID and the ending year of the data used. Suppose that the site ID is 25 and the ending year of the data is 2008. The directory name is then assigned as 025-2008. The road orientation is used as a part of file names. This inclusion is to maintain the compatibility to the TrafLoad. Suppose that the road orientation of this site is 8 (Northwest). The files produced by BullGuide in the directory "025-2008" are then:

- 025-8.trfl
- AGPV.025-8.txt
- SingleAxle.025-8.txt
- TandemAxle.025-8.txt
- TridemAxle.025-8.txt
- QuadAxle.025-8.txt
- MAF.025-8.txt
- Summary.txt

The first six files in the list are compatible to the TrafLoad export files. These files can be directly imported using the "Import TrafLoad" utility available under the "Tools" menu of the MEPDG software (see Figure 23). When these files are imported to the MEPDG software, the respective fields are populated. Unfortunately, the present Version 1.003 MEPDG has a normalization error when it populates inputs for the monthly adjustment factors (MAF) using TrafLoad export files. The file "MAF.025-8.txt" in the list is the corrected version of the MAF table, which can be loaded to the MEPDG software using the "Load MAF From File" button in the "Traffic Volume Adjustment Factors" window (see Figure 24).

The "summary.txt" file in the above list includes summary information about the site and data. It contains start and end date of the data used, the site name, the site ID, lane direction, pavement type, total number of days, total number of week days, total volumes of type 4-13 vehicles processed, truck AADT, and AADT. This data is useful in building a new MEPDG Project, as well as for other applications.

Figure 23: Import TrafLoad utility of the MEPDG tools menu

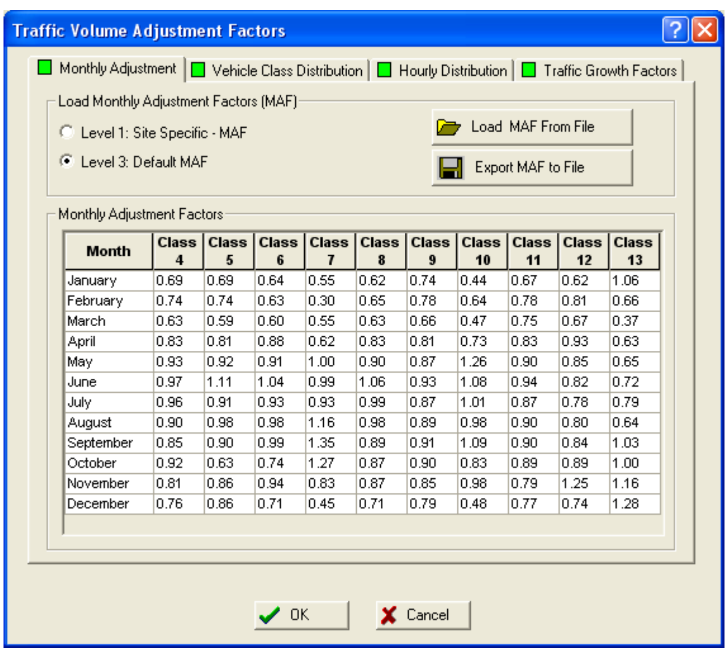


Figure 24: The “Load MAF From File” button is used to load the corrected MAF.

4.3.4 BullGuide user instruction

BullGuide consists of two tab pages as shown in Figure 25. The main functions are (1) to produce the import files for the MEPDG software and (2) to visualize and analyze the data. It first tests data availability and provides choices of data filtering for quality control. The steps of producing the MEPDG input files are:

- Set the root folder of the WIM csv data
- Select the site
- Select the design lane and pavement type
- Select the start and end dates of the data to be used
- Select the data filtering choices
- Set the folder for the output files in the Export/Graphs tab
- Click the "Run MEPDG Export" button in the Export/Graphs tab

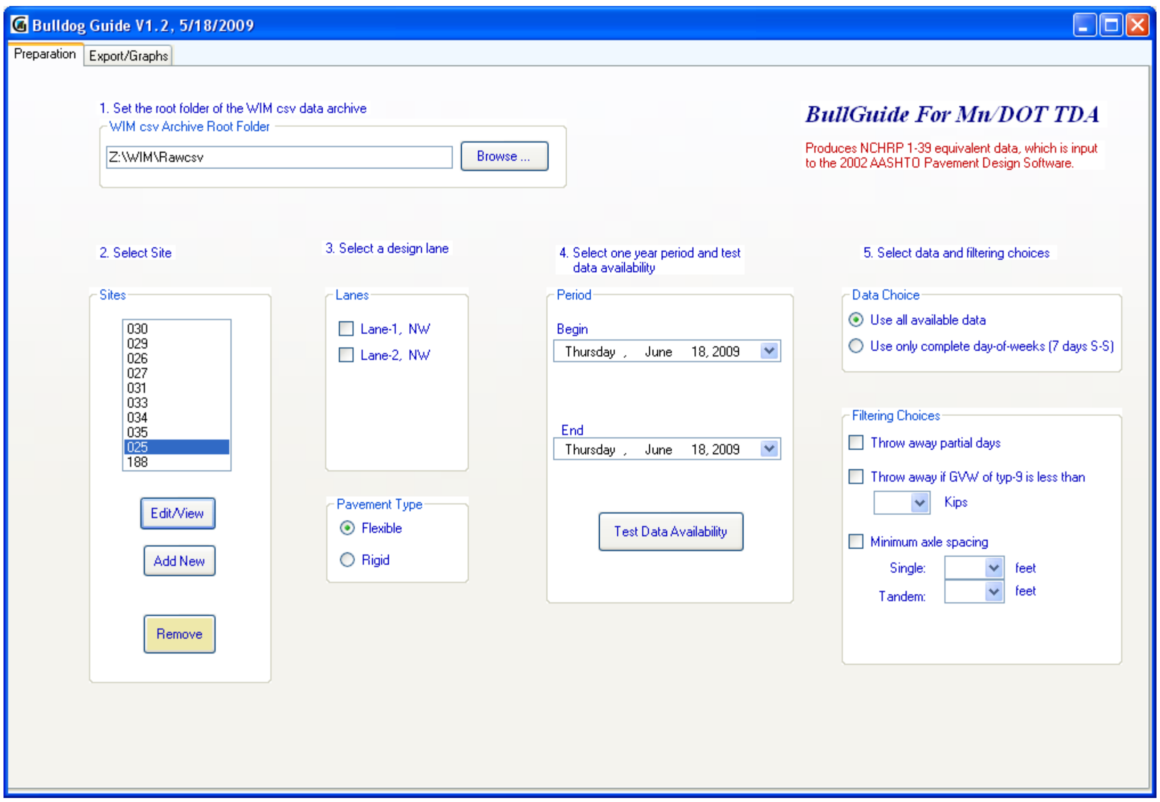


Figure 25: BullGuide software

The BullGuide includes a data visualization tool for analyzing the export data. The graph utility screen is shown in Figure 26. The data plotting utilities implemented include AGPV data, AADT, HDF, MDF, and load spectra. The load spectra are plotted by class and axle groups. The plot examples of the data from site 25 are shown in Figures 27-31. These plots can be used for checking the quality or validity of the data. For example, the load spectra of tandem axles of type-9 vehicles must have two peaks in a certain range. This pattern is shown in Figure 27. If this pattern is not observed from the graph, the user must suspect the data quality is poor and may select a different data period.

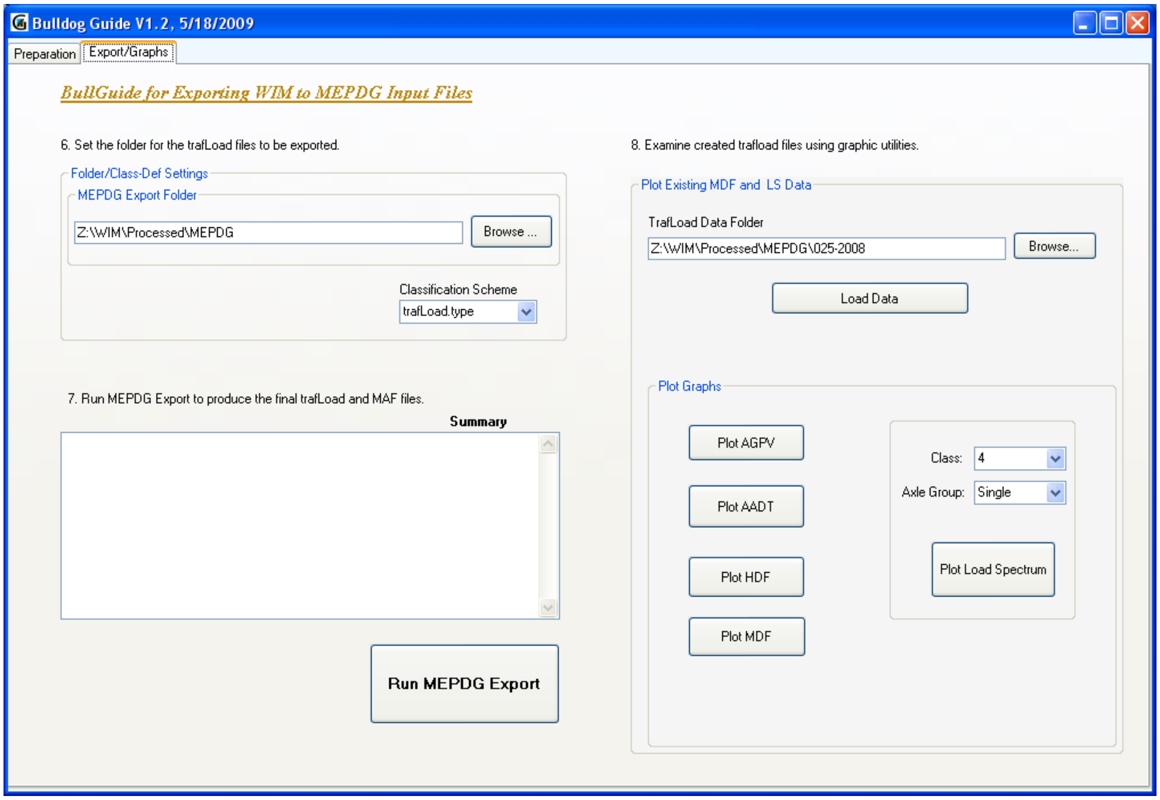


Figure 26: Utility functions in Export/Graphs tab

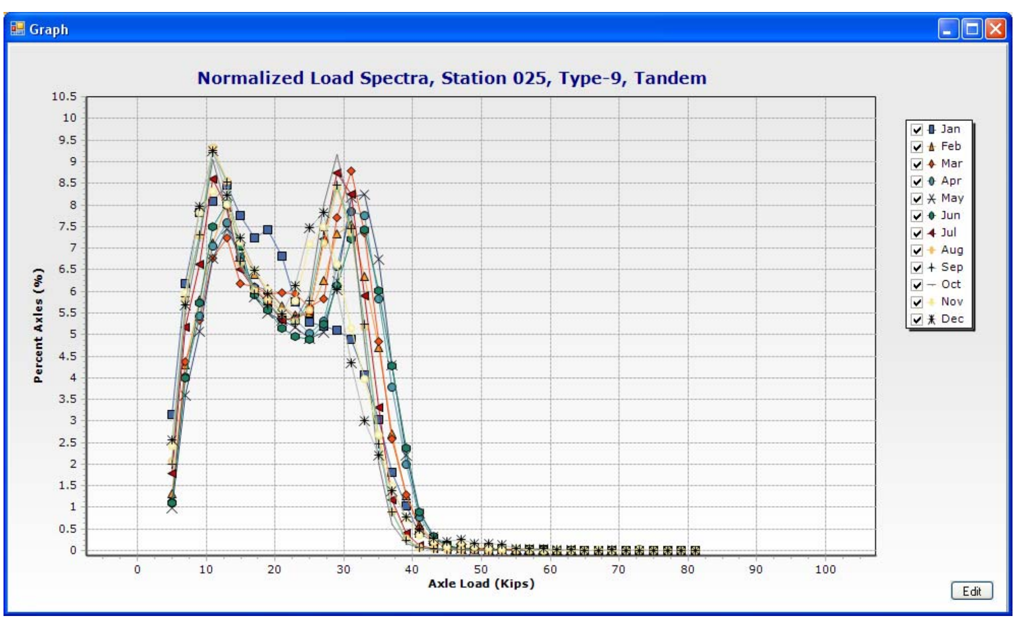


Figure 27: Normalized load spectra of tandem axles of the type-9 vehicles at Site 25

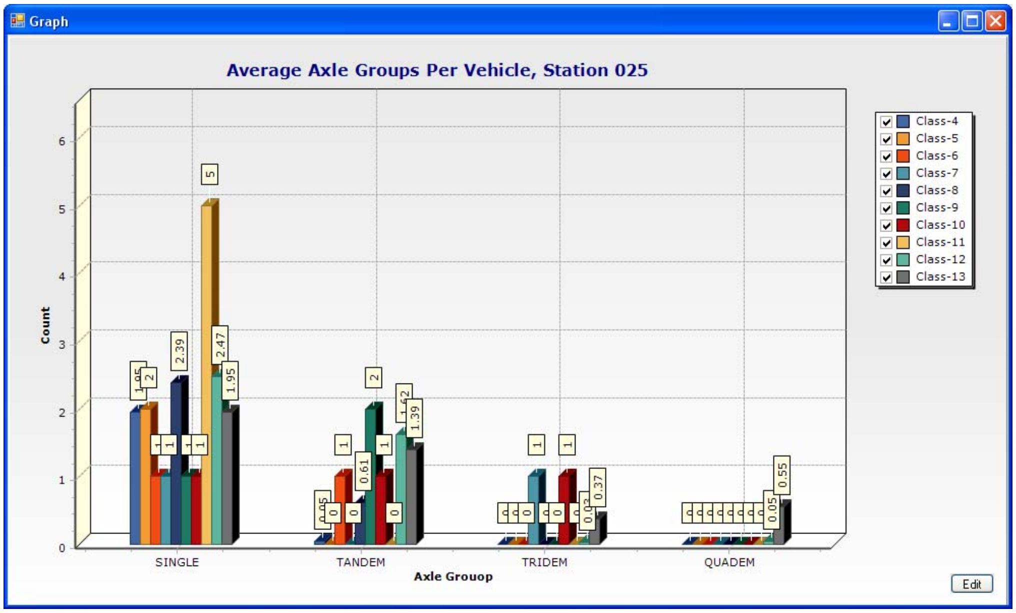


Figure 28: Plot of AGPV data for site 25

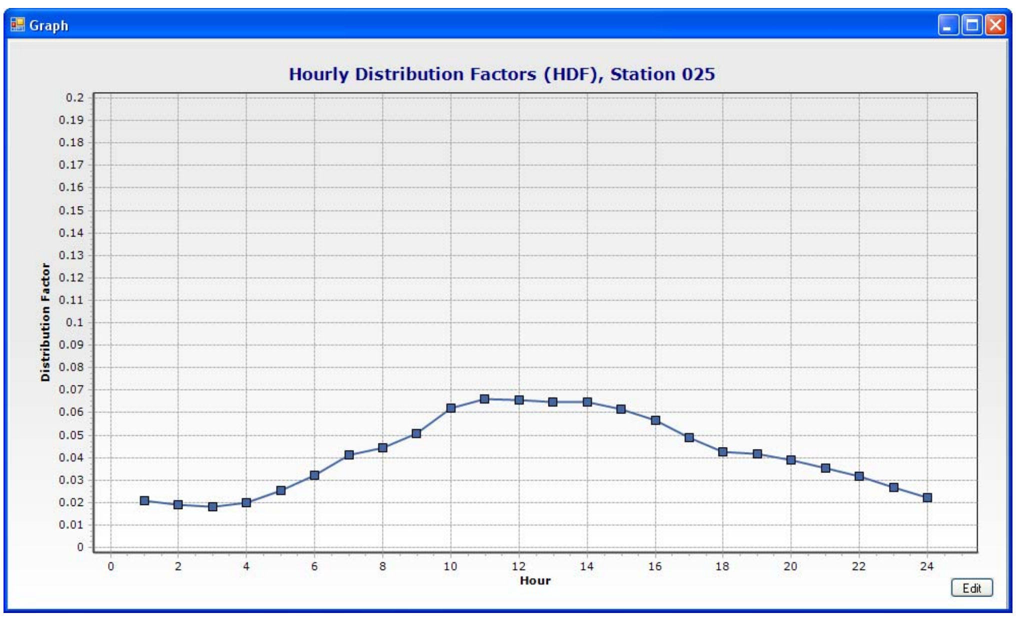


Figure 29: A plot example of normalized hourly distribution factors (HDF)

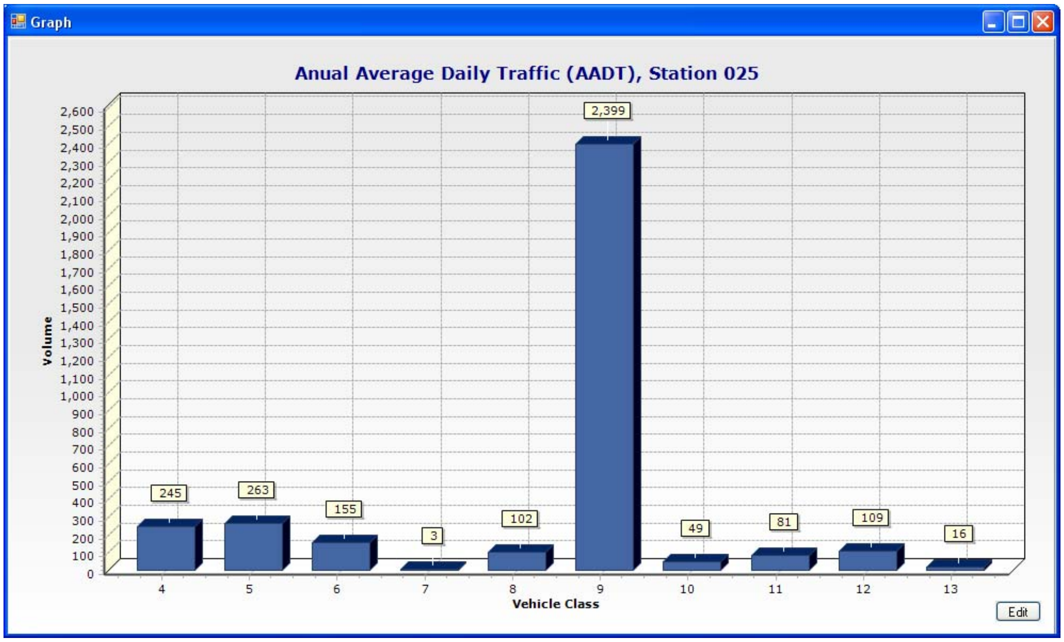


Figure 30: Plot of AADT for site 25

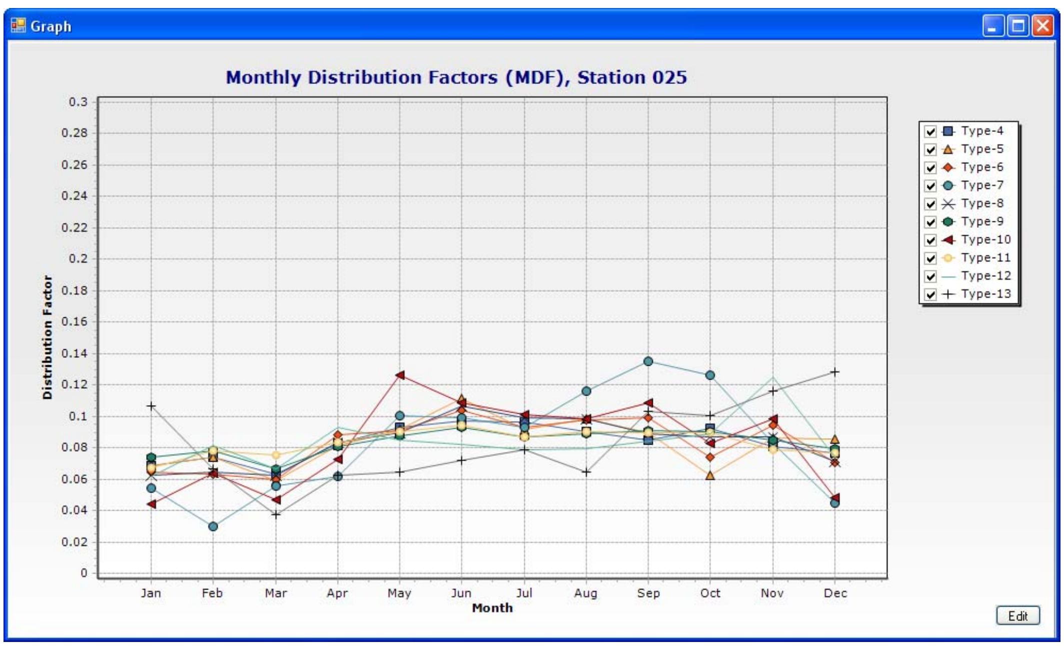


Figure 31: Plot of monthly distribution factors (MDF) of the site 25

4.4 BullPiezo

4.4.1 Introduction

The BullPiezo software package was developed as a VC data application for this project. The software consists of three tabs, as shown in Figure 32. It initially started by writing codes to compute seasonal adjustment factors for VC sites, and then more functions were gradually added. The present version provides four main functions:

- data computation of seasonal adjustment factors (SAF), AADT, and monthly average daily traffic (MADT) using VC data
- plot of AADT, average daily truck volume, SAF of each vehicle type
- data computation of SAF, AADT, and MADT using WIM data
- generation of TMG reporting data

The computational procedures of SAF, AADT, and MADT data were implemented by following the guidelines provided in FHWA TMG 2001 [1].

In the past, SAF were only generated using VC data at Mn/DOT. This software demonstrates that SAF can also be generated using WIM data. This was easily accomplished, due to the network access of the data warehouse created and the unified standard format.

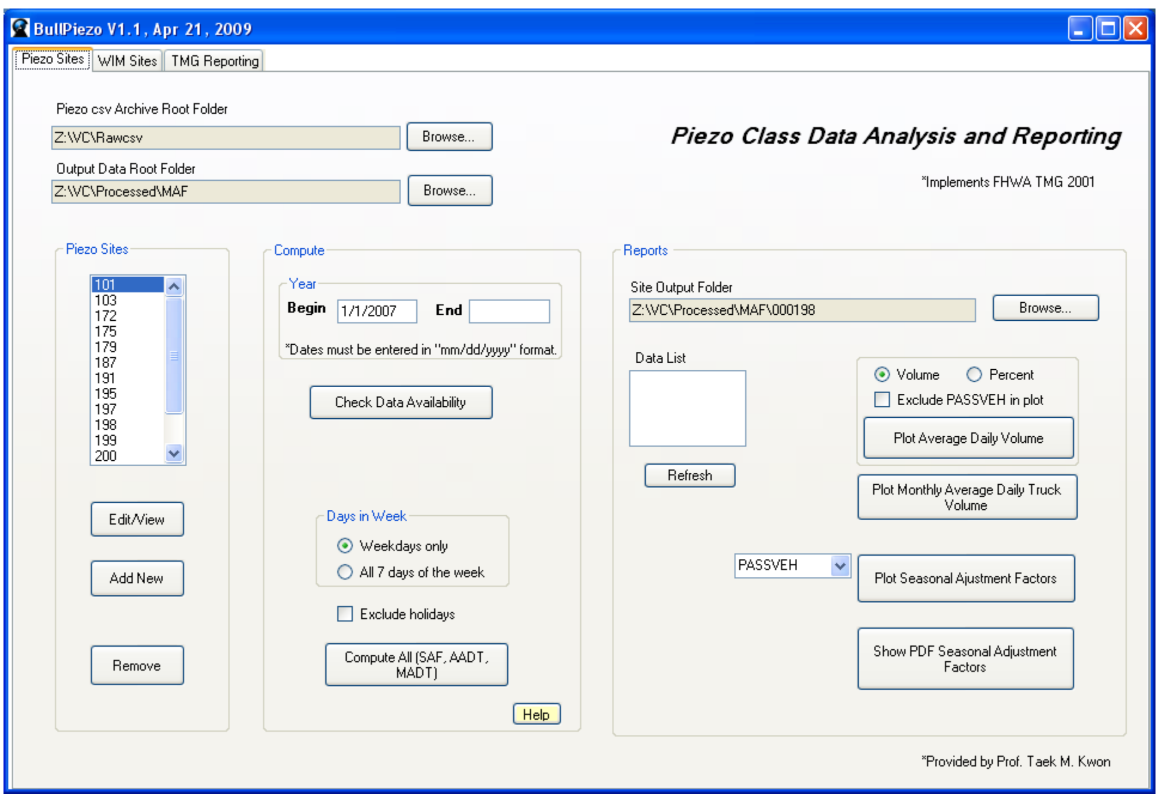


Figure 32: BullPiezo screen

4.4.2 Computation

Average annual daily traffic (AADT) can be computed in two ways, i.e., by a simple average or an average of an average. AASHTO recommends the average of an average approach, i.e.,

$$AADT = \frac{1}{7}\sum_{i=1}^{7}\left[\frac{1}{12}\sum_{j=1}^{12}\left(\frac{1}{n}\sum_{k=1}^{n}Vol_{ijk}\right)\right] \qquad (5)$$

where:

Vol = daily traffic for day k, of day-of-week i, and month j
i = day of week
j = month of year
k = week of the month
n = the number of days of that day of the week during that month

This computation starts by computing the average day of week traffic volume for each month. Each day of week traffic volume is averaged over 12 months, which results in seven values. An average of the seven values becomes AADT.

Let $MADT_j$ denote the average daily traffic for month *j*. Seasonal adjustment factor for month *j* (SAF_j) is then computed using:

$$SAF_j = \frac{AADT}{MADT_j}$$

In BullPiezo an option for selecting a week as weekdays only or all seven days was implemented. If the “week days only” option is selected, MADT and SAF are computed using only weekdays. However, AADT is always computed using seven days of week. The “week days only” option should be used if the short count data are collected only during the weekdays (Monday through Friday).

4.4.3 User instruction

BullPiezo runs on one VC site at a time and produces four output files. Suppose that the site ID is 101 and the ending year of the data computed is 2008, then the output files are:

- AADT-101-2008.txt
- MADT-101-2008.txt
- MAF-101-2008.txt
- availDates-101-2008.txt

All of the output files are written in ASCII text, so that they can be read using any text editor. The AADT-101-2008.txt file contains AADT of the site 101 for each vehicle type. Similarly, MADT-101-2008.txt contains monthly average daily traffic, and MAF-101-2008.txt contains monthly adjustment factors, which Mn/DOT refers as seasonal adjustment factors

(SAF). In all files, the first line shows the file information which consists of data type, year, site ID, and 5 or 7 days of week used. The second line is the header for the actual data. The Type column specifies vehicle class groups which are defined in Table 9. This grouping is used at Mn/DOT for VC data. Group types 2-8 are considered as Trucks.

Table 9: Vehicle Class Grouping

Group Type	Type Name	Grouping by FHWA Class
1	PASSVEH	Class 1 + 2 + 3
2	2AXSU	Class 5
3	3+AXSU	Class 6 + 7
4	3AXSEMI	(Class 8) * 0.35
5	4AXSEMI	(Class 8) * 0.65
6	5+AXSEMI	Class 9 + 10
7	TRKTRLF/BUS	Class 4
8	TWINS	Class 11 + 12 + 13

The availDates-101-2008.txt file contains the dates of data used for the computation in a calendar format. If no data is present for a certain day, that day is marked with “0”. A sample file for each listed above is attached in Appendix-C. Next, how to actually use the BullPiezo to produce the output files is described.

<u>How to produce the data</u>

Step 1: Select the “Piezo csv Archive Root Folder” and the “Output Data Root Folder” using the “Browse” buttons. According to the data warehouse organization defined in Section 2.2, the root folders are selected. Figure 33 illustrates which folder should be placed where.

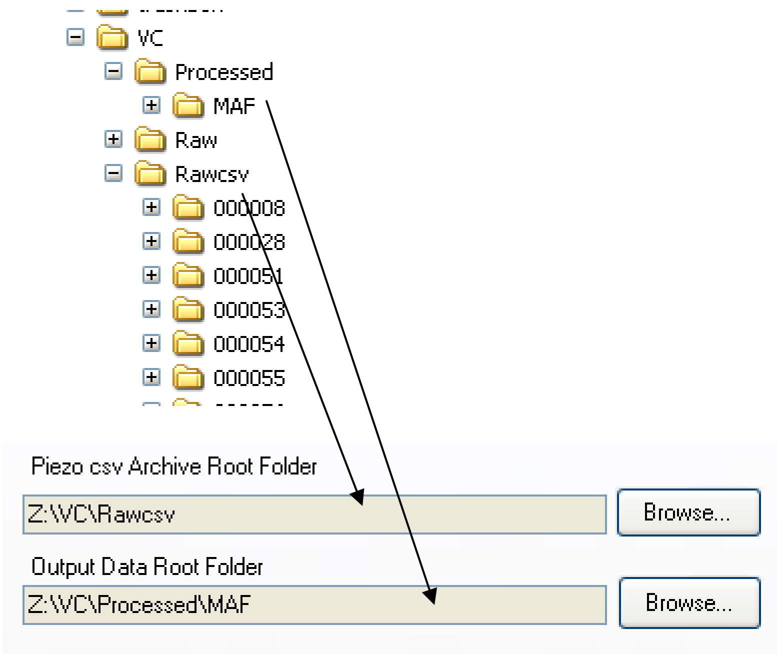


Figure 33: Root folder entry example

Step 2: Select a site from the "Piezo Sites" list box. If the site is not present, define it using the "Add New" button.

Step 3: Enter the "Begin" and "End" dates of the year. Check data availability of the site by pressing the "Check Data Availability" button. (It should be noted that the Year does not have be defined from Jan 1st to Dec 31st. It can be any period, for example, 6/1/2007 to 8/1/2008 or multiple years.)

Step 4: Click the "Compute All" button to produce all necessary files.

Steps 2-4 are illustrated using Figure 34. It should be noted that SAF can be produced using only weekdays, all seven days of week, or excluding holidays. These selections are not affected for the AADT computation. AADT is always computed using all available data. It is recommended that user produce multiple years of output files for each site to make it more convenient for later comparative plotting.

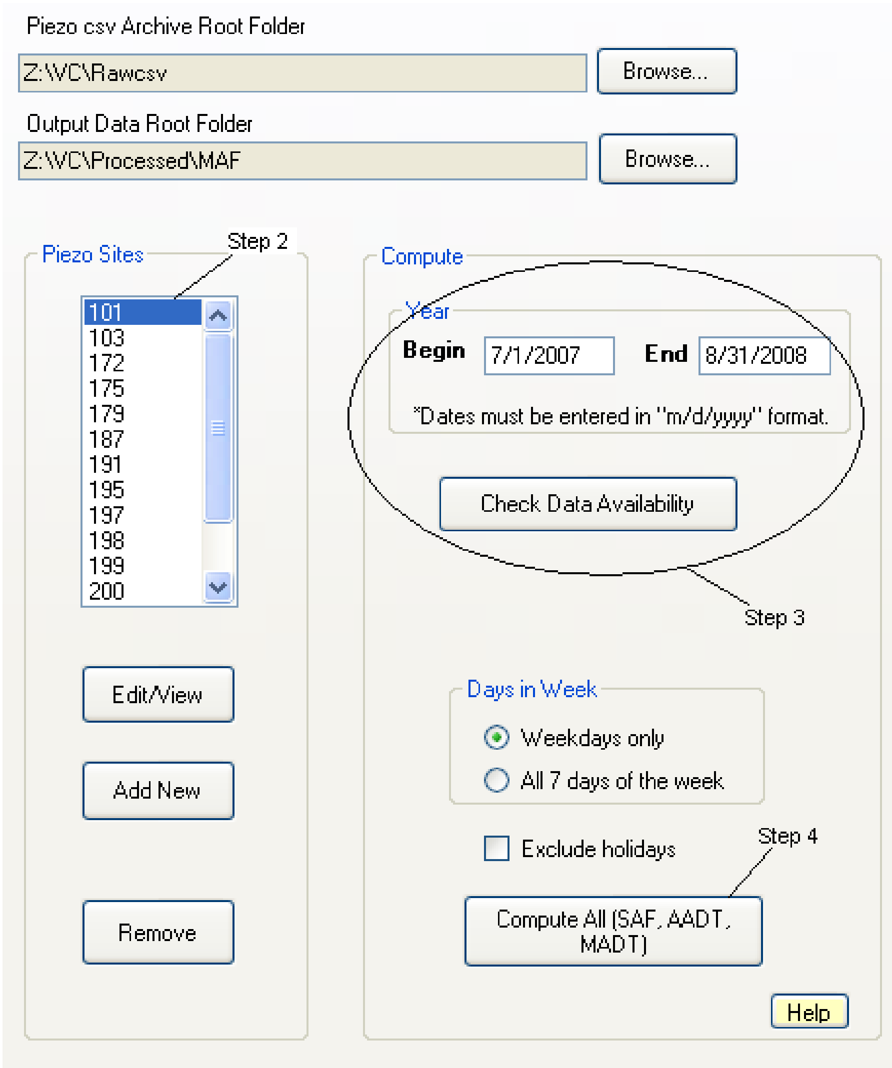


Figure 34: A screen example of Steps 2-4 for producing the output files

Data plotting

Plotting utilities are included as a part of the software package for reporting applications. The plotting routines include plots of average daily volumes, monthly average daily truck volume, and seasonal adjustment factors. These plotting routines were developed according to the suggestions provided by Mn/DOT. Figure 35 shows the final plot utility screen implemented.

As the first step of using the plot utility, the "Site Output Folder" should be specified using the "Browse" button, followed by a click the "Refresh" button. This creates a list of the sites and years of the data available as shown in Figure 35. Selecting multiple items using the check marks allows for a plot of multiple years on the same graph.

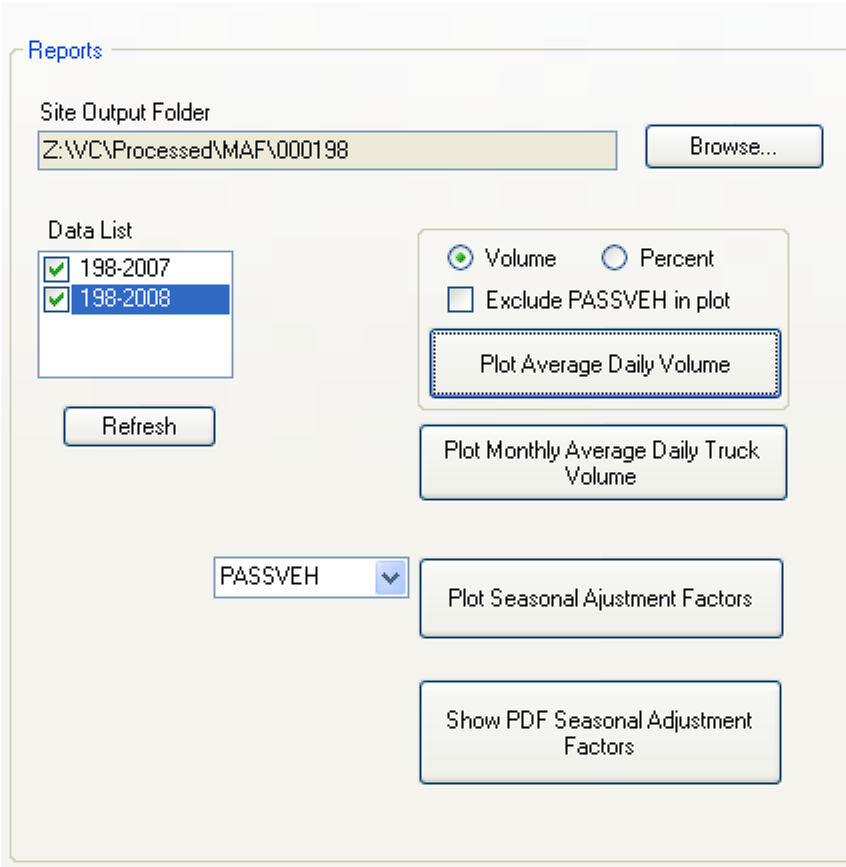


Figure 35: BullPiezo plot utility screen

Below shows examples of the plots implemented.

Plot of Average Daily Volume: Average daily volume per each vehicle group type can be either plotted by volume or percent. Also, this routine has an option that can exclude the PASSVEH group. This exclusion can be useful if the user wishes to include only trucks. Figure 36 shows an example that plots traffic volume distribution for the years 2007 and 2008 of site 198. The AADT information is shown under the title of the graph. This example plot shows the distribution changes from year to year for different vehicle groups. In this example, it can be noted that the volumes of passenger vehicles and five axle semi-trucks were increased while the other types are decreased. The edit button is used to modify the appearance of the graph, such as change of the title, 3-D bar to 2-D bar, export of image files, etc.

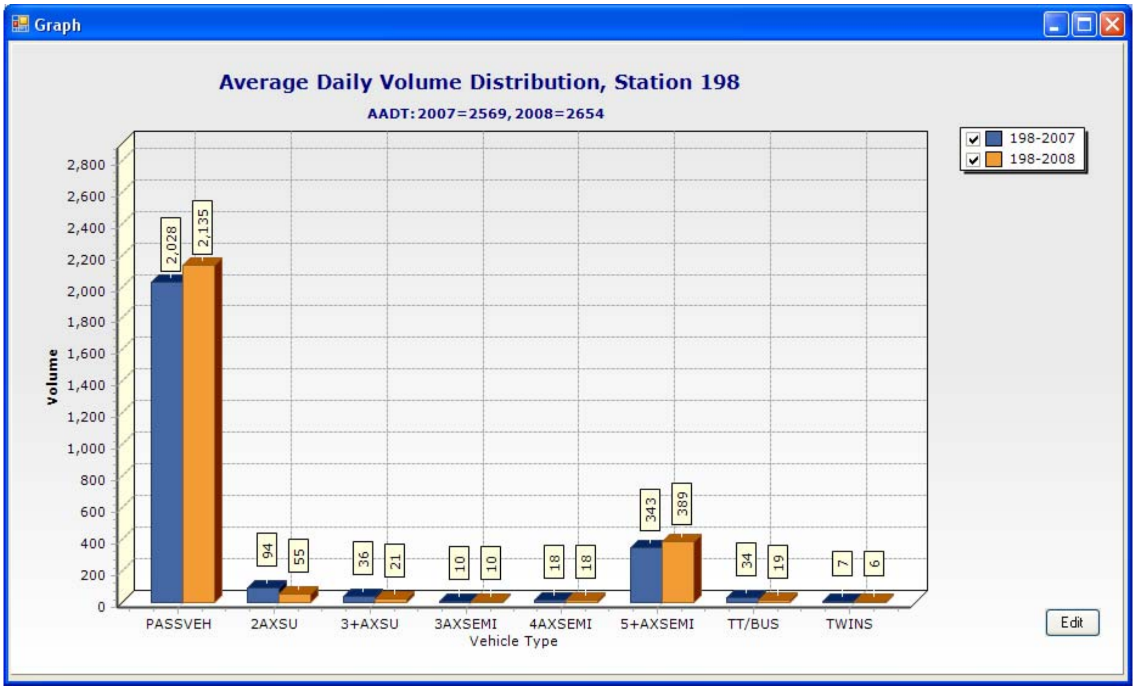


Figure 36: An example plot of average daily volume distribution

Plot of Monthly Average Daily Truck Volume: This plot shows monthly change of average daily truck volume. An example plot is shown in Figure 37.

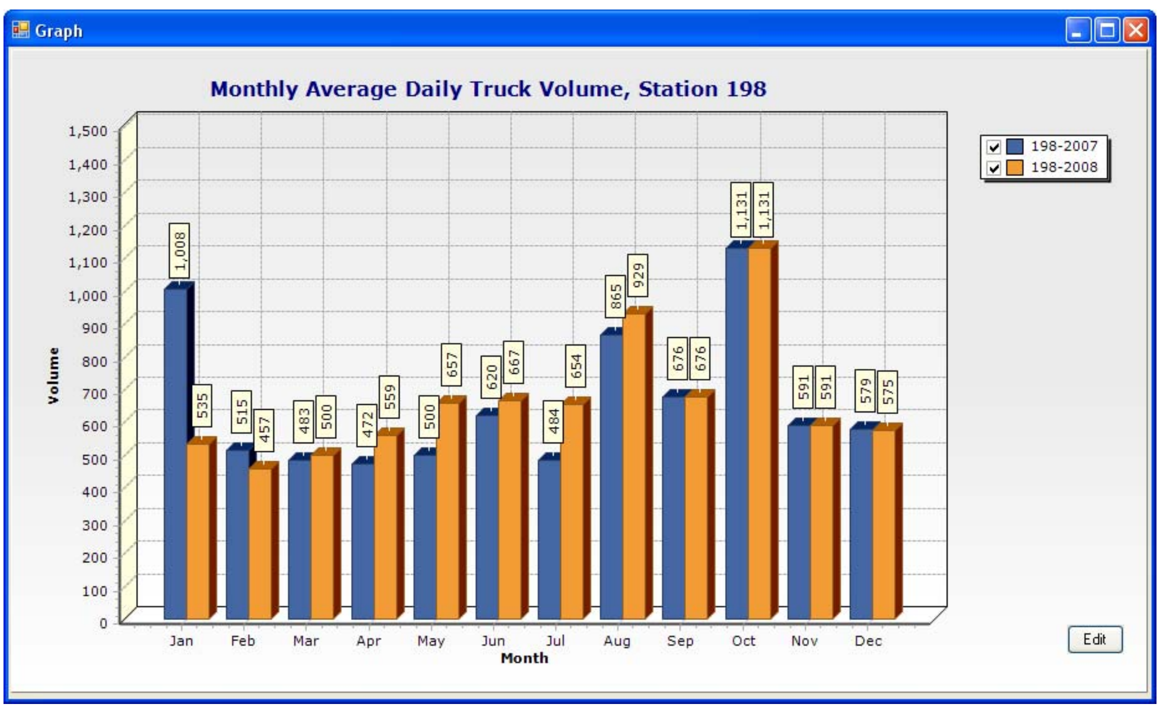


Figure 37: An example plot of monthly change of average daily truck volume

Plot of Seasonal Adjustment Factors: Seasonal adjustment factors (SAF) are plotted for multiple years on the same graph for the selected vehicle group (combo box selection). The months are always plotted from Jan first to Dec last, even if the actual order is different. For example, if the data period was from June 2007 to August 2008, Jan 2008 is plotted first and Dec 2007 last. If the selected period contains more than one year, an exponential average is used to give more weights on the current year. Figure 38 shows an example plot of SAF. SAF can also be displayed in a table form for reporting applications. This table is outputted as a pdf file and an example screen is shown in Figure 39.

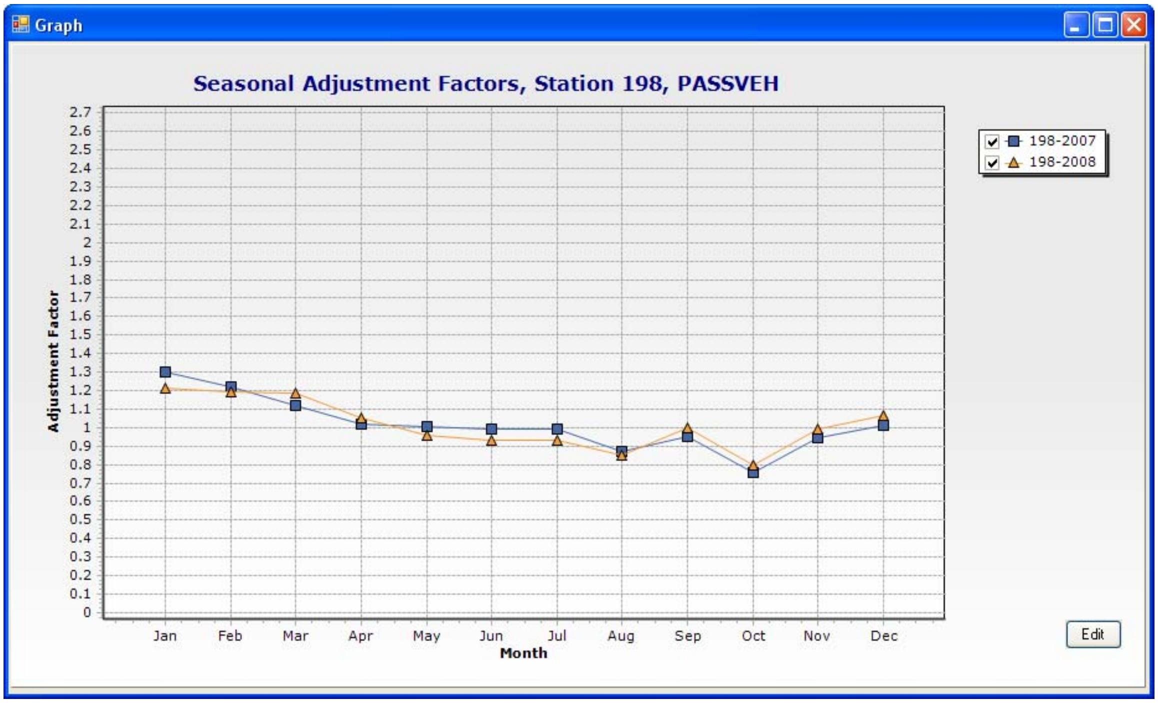


Figure 38: A plot example of seasonal adjustment factors

Seasonal Adjustment Factors

Station: 198 Lanes: #1 #2
Year: 2007

Minnesota Department of Transportation
Office of Transportation Data & Analysis
Generated by BullPiezo

Month

Body Type	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
PASSVEH	1.299	1.221	1.117	1.016	1.008	0.990	0.990	0.875	0.950	0.759	0.945	1.010
2AXSU	0.208	0.715	1.860	1.675	1.660	1.661	1.770	0.952	1.391	0.760	1.625	1.646
3+AXSU	0.252	0.909	1.591	1.506	1.204	0.464	1.360	0.820	1.094	0.732	1.313	1.102
3AXSEMI	1.024	1.376	1.652	1.396	1.583	1.503	1.617	0.438	0.956	0.288	1.144	1.212
4AXSEMI	1.024	1.376	1.652	1.396	1.583	1.503	1.617	0.438	0.956	0.288	1.144	1.212
5+AXSEMI	2.318	1.192	0.904	0.958	0.918	0.771	0.940	0.585	0.663	0.457	0.752	0.763
TT/BUS	0.224	1.065	3.905	3.972	2.129	1.902	2.106	0.511	1.544	0.325	1.780	2.987
TWINS	0.084	1.414	1.381	1.300	1.097	1.590	1.146	0.932	0.948	0.846	1.213	1.183

Figure 39: An example of SAF table in a pdf format

It was shown above that SAF, MADT, and AADT data are computed from the csv VC data. Considering that WIM data is more detailed than VC data, it is not too difficult to produce the SAF, MADT, and AADT data. Therefore, BullPiezo implemented utilities that produce the same SAF, MADT, and AADT data using the WIM csv files. The screen for this utility is shown in Figure 40. Notice that the screen is nearly identical to the VC data portion. Only differences are the data source and the sites. This utility demonstrates that WIM data can be used for VC data applications.

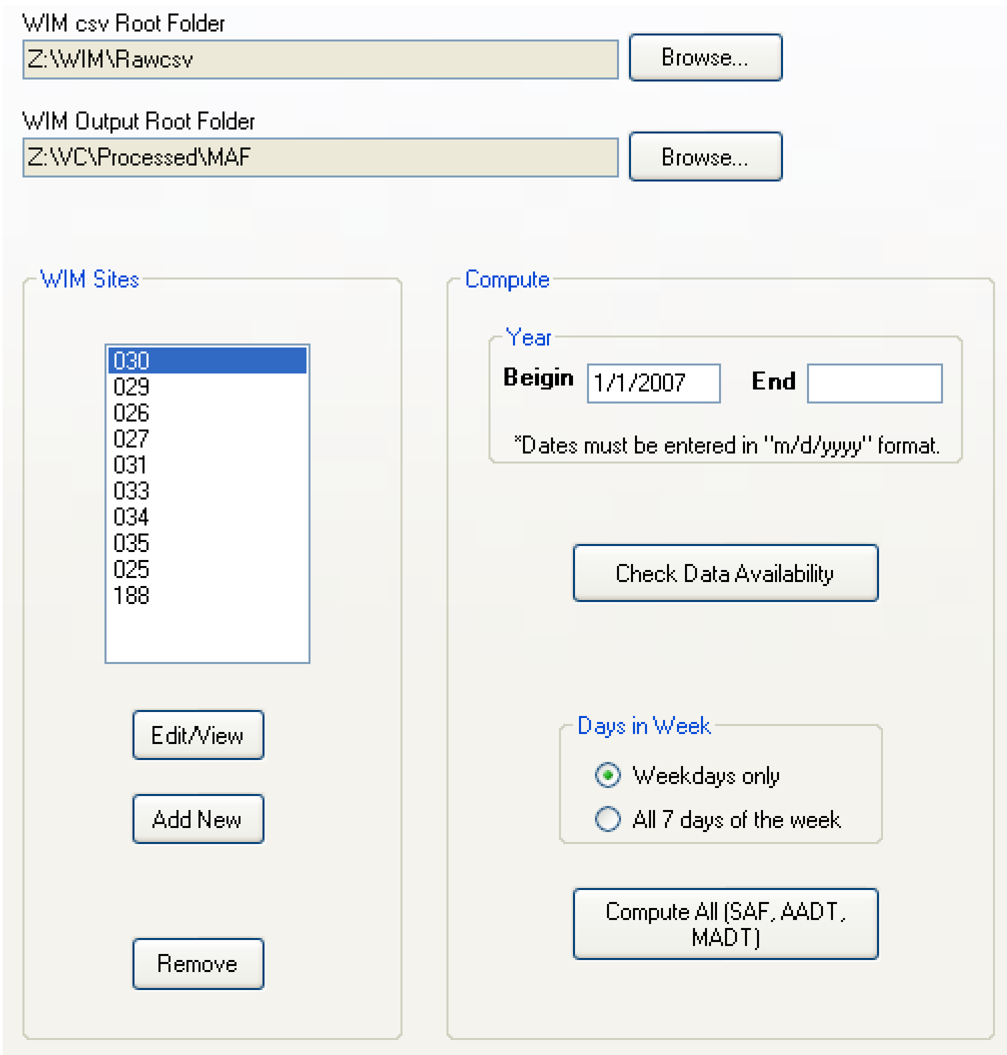


Figure 40: SAF, AADT, MADT data computation utilities for WIM csv files

Another useful application of BullPiezo is to produce TMG 2001 formatted data for sharing and submittals to FHWA using the VC data. The VC data can be converted to TMG 2001 Traffic Volume and Vehicle Classification formats. However, the VC data cannot be converted to the TMG 2001 Truck Weight format, since no individual vehicle weight records are available from the VC data. The screen of this utility is shown in Figure 41. At the time of this writing, only the TMG Station Description Report was implemented. The rest of functions will be added in the future versions.

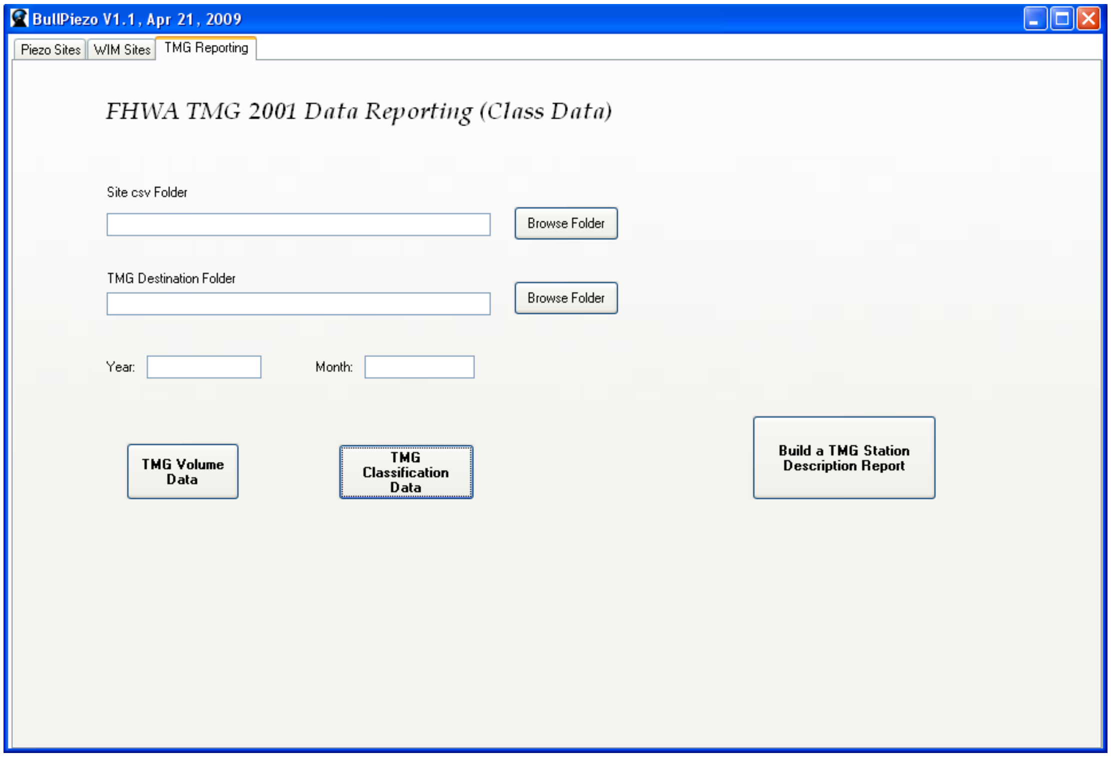


Figure 41: Data conversion utility --- from VC data to TMG 2001 formats

CHAPTER 5: CONCLUSION

This project was formulated based on the experience of a past project that successfully implemented a networked data automation system for ATR and short-count data based on a traffic data warehouse [1]. The key to the success of this past project was the traffic data warehouse that housed large scale loop data and the development of software that can freely generate ATR and short-count data without installing data collection devices. The present project is to expand the data warehouse functionality to include Mn/DOT's VC and WIM data.

Mn/DOT presently manages twenty-nine VC sites and twelve WIM sites installed on various Minnesota roadways. The number of VC and WIM sites is expected to continuously grow. One of the main challenges is management of the large amount of data and extraction of information from the data pool. Building a well organized data warehouse using inexpensive network storage devices is a solution that this project pursued. However, unlike the loop data, the VC and WIM raw data did not have a uniform format because of a mix of devices from different manufacturers and models. Moreover, all of them were proprietary binary formats that can only be understood by the manufacturer provided software. Therefore, the first task was to develop data translation software that translates binary data to standardized unified formats. The research team successfully decoded all of the binary formats of VC and WIM data and implemented the decoding logic into two software packages: one for WIM data and the other for VC data. The unified formats were developed as a table that consists of ASCII encoded text strings separated by commas. These formats are simple and can be easily read by common text editors, spreadsheet programs, and databases. The increased size by using ASCII codes was not a concern for data warehousing since network storages have a large capacity. For the organization of the data, a hierarchical tree structure was used for a fast search. The data organization function was directly implemented in the decoding software, so that the translated data is immediately organized into the data hierarchy as the raw binary data is translated into the unified formats. The software was delivered to Mn/DOT and a VC/WIM data warehouse was successfully constructed.

Three application software packages were developed as a part of the project. These software packages can be installed on any Mn/DOT PC that has the network access to the VC/WIM data warehouse. They are BullReport, BullGuide, and BullPiezo. The BullReport provides functionalities of load spectra analysis, TMG formats, and 21 data summary reporting utilities in pdf, text, web, and spread sheet formats. The BullGuide directly generates MEPDG traffic input data for the WIM sites. The BullPiezo generates seasonal adjustment factors for short-count stations, AADT (Annual Average Daily Traffic), and monthly average truck traffic using VC data. All of the functionalities of these software packages were suggested by Mn/DOT.

This project was successful in several aspects. The first is that the data warehouse solution provided a better solution to existing data problems. The second is that the tools developed are actually used for the present Mn/DOT tasks and improved efficiency. The third is that the new system produces the outputs much faster than the proprietary software used in the past. The fourth is that Mn/DOT can install the software in unlimited number of PCs without licensing concerns.

For future work, implementation of security functions is recommended. For example, several data access security levels can be developed for different user levels. Also, further automation in the data warehousing is recommended. This includes automated data polling, decoding, and organization.

The lessons learned from this project are: (1) a large amount of VC/WIM data can be easily managed and shared by creating a simple networked data warehouse; (2) unified format translation can solve diverse nature of data collection devices; (3) a single point network access provides simplicity in data application developments; (4) VC applications such as seasonal adjustment factors can be derived from the WIM data, providing additional VC sites without installing the VC data collection devices.

REFERENCES

[1] T.M. Kwon, *TMC Traffic Data Automation for Mn/DOT's Traffic Monitoring Program*, Minnesota Department of Transportation, Report No. MN-RC-02004-29, July, 2004, St. Paul, MN.

[2] W.H. Inmon, *Building the Data Warehouse*, New York: John Wiley & Sons, 1993. ISBN 0-471-56960-7.

[3] T.M. Kwon, *Transportation Data Research Laboratory: Data Acquisition and Archiving of Large Scaled Transportation Data, Analysis Tool Developments, and On-Line Data Support*, Report #: CTS 09-07, ITS Institute, Center for Transportation Studies, Minneapolis, MN, March 9, 2009.

[4] Federal Highway Administration (FHWA), *Traffic Monitoring Guide*, U.S. Department of Transportation, Federal Highway Administration, and Office of Highway Policy Information, Washington D.C., May 1, 2001.

[5] NCHRP 1-37A, "Using Mechanistic Principles to Implement Pavement Design," 2002, FHWA, Washington D.C.

[6] MEPDG software, Version 1.003, May 24, 2007, Applied Research Associates, Inc., Washington, D.C. www.trb.org/mepdg/software.htm. Accessed October 30, 2008.

[7] Cambridge Systematics, Inc., Washington State Transportation Center, and Chaparral Systems, Inc., *NCHRP Report 538: Traffic Data Collection, Analysis, and Forecasting for Mechanistic Pavement Design*, Transportation Research Board of the National Academies, Washington, D.C., 2005 (NCHRP 1-39).

[8] C. Dahlin, "Proposed method for calibrating Weigh-in-Motion systems and for monitoring that calibration over time," *Transportation Research Record*, 1364, pp. 161-168.

[9] FHWA, "Bridge formula weights," FHWA-HOP-06-105, Office of Freight Management and Operations, U.S. Department of Transportation, Washington D.C., Aug. 2006.

APPENDIX-A: PVR AND VC FORMAT

PEEK PVR Binary Format (WIM data)

The binary file consists of heading information followed by repetition of several record types.

Heading information

Index	Bytes	Example value(s)	Description
1	1	255	Start of file mark
2	1	2	Version Number (could be a mark)
3	6		Sec, min, hr, day, mo, yr, **Start time**, one byte per each item
4	1	4	Could be a mark
5	6		Sec, min, hr, day, mo, yr**, End time**, one byte per each item
6	1	5	Could be a mark
7	8	02F4CC96 02610011	Serial number, It should be printed as Hex
8	2	$6, $6	Firmware version no: 6.6, must printed as Hex
9	1	1	Comm port number or ID
10	1	0	Separation mark
11	12	"000188000000"	Station ID (a 12-digit numeric number). In some file it is used for Site ID (non numeric). ASCII characters.
12	1	0	Separation mark
13	12	"188-4L-CLASS"	Station ID, 12 ASCII characters
14	1	0	Separation mark
15	**1**	**4**	**Number of arrays. Equivalent to number of lanes**
16	1	8	Number of sensors, two per array
17	1	4	Count of subsequent sensor definitions
18	4	17	Sensor type for each array: 11=loops [], 7=axle sensors | |, 17=WIM | [] |
19	1	2	Version Number. It appears define header version
20	**1**	**C7**	**Type 1, Info mask**
21	1	BD	Type 2, Info mask
22	1	OC	Type 3, Info mask
23	8	"MN022008"	Name of classification table in ASCII characters
24	1	0	Separation mark
25	2	41, 4	? Expect 4 bytes.
26	2	176,166	Max Veh Length in mm (166*256 + 176)/304.8=140ft

27	2	148,53	Max inter-axle distance in mm (53*256 + 148)/304.8=45ft
28	2	41,8	? Expect 8 records
29	8*(1+4)	 (0) 242,3,232,3 (1) 242,3,232,3 (2) 212,3,232,3 (3) 212,3,232,3 (4) 212,3,232,3 (5) 212,2,232,3 (6) 147,3,232,3 (7) 147,3,232,3	Initial Sensor Calibration 1.010000=(3*256 + 242)/(3*256+232) 1.010000 0.980000 0.980000 0.980000 0.980000 0.915000 0.915000
30	2	21, 4	Expect initial Autocal for 4 records
31	8*(1+4)	 (0) 242,3,232,3 (1) 212,3,232,3 (2) 212,3,232,3 (3) 147,3,232,3	Initial Autocal for 4 records 1.010000, for array1 0.980000, for array 2 0.980000, for array 3 0.915000, for array 4
32	4*(4)	 248,255,255,255 248,255,255,255 248,255,255,255 248,255,255,255	Class masks F8FFFFFF, array 1 of 4 F8FFFFFF, array 1 of 4 F8FFFFFF, array 1 of 4 F8FFFFFF, array 1 of 4

There are several different types of records that are repeated, followed by the header information. Four record types are summarized below.

Record Type=199

Index	Bytes	Example value(s)	Description
1	1	199	Record type. Two types are used 199: Normal record 135: When status is 236 or 172 which are error conditions, the record starts with 135.
2	2	49,1	Timestamp, min:sec Min=(1*256 + 49)\1024=0 Sec = [(1*256 + 49) and 0x3ff]/10=30
3	1	128	Timestamp, hr, only five LSB are used Hr=128 and 0x1f=0
4	1	14 or 242	???, unknown.
5	1	6	Lane # = (divided by 2) + 1 0 → Lane 1 2 → Lane 2 4 → Lane 3 6 → Lane 4

6	1	5	Number of axles, 5 axles in this case
7	2	173,104	Speed 60mph=(104*256+173)/447
8	1	8	Vehicle class Number, need to add one. This example is class 9.
9	1	194	?? Don't know. Need to figure out
10	1	61	?? This byte become the record-length byte when the record type is 135
11	1	60	Error code 60=normal 61=indicates that the status byte is next followed by the length byte. It leads to two bytes for GVW for length=13 236 (EC)=critical error, no axle spacing, no axle weights. 3 more bytes after the length byte 172 (AC)=critical error, No axle distance. The end of record should be searched by 199 or 135. The length byte does not work in this case. 20=no axle weight data
12	1	25	Length (25 – 2) bytes ahead until the end of this vehicle record
13	1 or 0	0xCE	Status byte, it only exists if err-61
14	2	B1, B2	Vehicle #=(B2*256 + B1)
15	2	192,171	GVW in gram, 48.4 Kips, need further investigation (171*256 + 192)*2.205/2=48.4
16	1	117	??? Unknown, need to figure out This byte exists only in the following cases: - if length byte > 13 - if (length byte=13) and (error code=60) This byte does not exist - if length byte =12 - if length byte = odd and error=61
17	4*(2)	 150,17 6,5 85,34 209,4	Two bytes for each axle spacing in mm. Since this is a 5 axle vehicle, four axle spaces are in the record. Axle space 1, 14.8 ft = (17*256 + 150)/304.8
18	5*(2)	 75,37 120,34 44,32	Axle weights. five records in this case (37*256+75)*2.205/2=10.5 Kips

		253,33 137,33	

Record type-135, Vehicle Records

Index	Bytes	Example value	Description
1	1	135	Record type 135: error conditions When the status is 236 or 172, the vehicle record starts with 135.
2	2	49,1	Timestamp, min:sec Min=(1*256 + 49)\1024=0 Sec = [(1*256 + 49) and 0x3ff]/10=30
3	1	128	Timestamp, hr, only five LSB are used Hr=128 and 0x1f=0
4	1	14	?, 14 classes
5	1	6	Lane # = (divided by 2) + 1 0 → Lane 1 2 → Lane 2 4 → Lane 3 6 → Lane 4
6	1	1	Number of axles. One axle in this case.
7	2	173,104	Speed 60mph=(104*256+173)/447
8	1	14	Vehicle class, need to add one. This example is class 15.
9	1	13	?? Don't know. It appears err. 60,61
10	1	7	Length or count that indicates number of bytes remaining minus 2 for this record. Call it n.
11	2	1,0	Vehicle number, 1 in this case
11	1	236	Error code 236 (EC)=critical error, no axle spacing, no axle weights. 3 more bytes after the length byte 172 (AC)=critical error, No axle distance. The end of record should be searched by 199 or 135. The length byte does not work in this case.
12	n-3		The remaining bytes of the record. These bytes could be axle spacing or weights, but it should be interpreted as raw bytes.

Record type-4, Auto-calibration Records (8 bytes)

Index	Bytes	Example value	Description
1	1	4	Record type 4: auto-calibration value
2	1	8	Count, count-2 bytes are ahead
3	1	1	Start mark
4	5	255,142,3,127,3	1.016760

Record type-128, Auto-calibration Records (8 bytes)

Index	Bytes	Example value	Description
1	1	128	Record type 128: calibration of a form of maintenance record
2	2	Val1,val2	Timestamp, min:sec Min=(val2*256 + val1)\1024 Sec = [(val2*256 + val1) and 0x3ff]/10
3	1	81	Timestamp, hr, only five LSB are used Hr=(81 and 0x1f)=5
4	1	242	mark
5	4	6,128,2,4	Four bytes, not understood yet
6	1	8	Count, (count-2) bytes are ahead
7	Count-2	1,255,117,3,232,3	(count-2) data bytes

PEEK Binary Volume/Speed/Classification Format

Index	Bytes	Typical value	Description
1	1	255	Start of file mark
2	1	2	Version Number (could be a separation mark)
3	6		Sec, min, hr, day, mo, yr, **Start time**, one byte per each item
4	1	2 or 3	Could be a mark
5	6		Sec, min, hr, day, mo, yr**,** **End time**, one byte per each item
6	1	3 or 4	Could be a mark
7	8	02F4CC96 02610011	Serial number, It should be printed as Hex
8	2	$29, $4	Firmware version no: 4.29, must printed as Hex
9	1	1	Comm port number or ID
10	1	0	Separation mark
11	12	"000103000000"	Station ID (a 12-digit numeric number in ASCII). Some file it is used for Site ID (non numeric)
12	1	0	Separation mark
13	12	"103-4LCLSSPD"	Site ID, 12 ASCII characters. It is some times switched with Station ID.
14	1	0	Separation mark
15	**1**	**4**	**Number of arrays. Equivalent to number of lanes**
16	1	8	Number of sensors, two per array
17	1	4	Count of subsequent sensor definitions
18	4	11, 12 or 7	Sensor type for each array: 11=loop [], 7=axle sensor | |, 12=loops [] [] 17=WIM | [] |
19	1	2	Version Number. It appears define header version
20	**1**	**60**	**Data interval in minutes, usually 60 min** **This value could be used to determine how many rows of data in the data file.**
21	1	0	Speed units, 0=mph
22	1	0	Length units, 0=feet
23	1	0	Weight units, 0=Kips
24	1	0	Separation mark
25	**1**	**1-3**	**Number of active studies**
26	8	MN03F___	Name of classification table in ASCII The default is "FHWA-USA" It is ignored for volume data.
27	1	0	Separation mark
28	1	14 or 15	Number of classes
29	1	0	Separation mark
30	1	4 or 8	Number of active flows, 4=volume, 8=classification
31	2	1,0	Three bytes of numeric number. It reads as 401.

32	1	4	Number of subsequent bytes for maximum vehicle Length and inter-axle distance
33	2	40, 107	Maximum vehicle length in millimeter. Example, 107 * 256 + 40 = 27,432mm (27,432mm)/304.8=90feet
34	2	160, 47	Maximum inter-axle distance. Example, 47*256 + 160=12,192mm 12,192/304.8=40feet
35	1	0	Separation Mark

Subsequent data defines active studies, which depends on the Item 25, the number of active studies. For example, if the number of active studies is 3, then it has the following structure for defining the data format.

Study 1 of 3

Flow assignment
Heading 1:
Heading 2:
Heading 3:

Study 2 of 3

Flow assignment
Heading 1:
Heading 2:
Heading 3:

Study 3 of 3

Flow assignment
Heading 1:
Heading 2:
Heading 3:

*Note: A separation mark "0" is present between the Study definitions.

Flow assignment: The number of bytes used for defining the flow assignment appears to depend on the **number of active flows**. The format may depend on the sensor type.

1,2 two active flows, two lanes, vol
1,2,3,4 four active flows, four lanes, vol

0,1,1,2 four active flows, two lanes, piezo, speed
01,1,2,2,3,3,4 eight active flows, four lanes, piezo, speed

255, 1, 255, 2, 255, 3, 255, 4 eight active flows, four lanes, piezo, cls

Heading Formats

*Note: No separation mark is present between Headings, nor before or after.

Volume=1
1,4, 0,0,0,1,0,2,0,254,255
volume, 4 bins (for 4 lanes).
When 254 appears, it should read the following byte 255

Speed=3
3,13, 0,218, 69,148, 78,79, 87,11, 96,198, 104,129, 113,60, 122,247, 130,178, 139,109, 148,159, 174,21, 192,254, 255
Speed, 13 bins
Each pair denotes speed, i.e., (A,B)=(A*256 + B)/447 mph
Example, (69,148) = (69*256 + 148)/147 = 40mph
Above example is translated into mph as
0, 40, 45, 50, 55, 60, 65, 70, 75, 80, 100, 110 mph bins, which represents
0-40, 40-45, 45-50, 50-55, 55-60, 60-65, 65-70, 70-75, 75-80, 80-85, 85-100, 100-110, 110-above, in which 13 bins are formed.

Classification=2
2,15 0,0, 0,1, 0,2, 0,3, 0,4, 0,5, 0,6, 0,7, 0,8, 0,9, 0,10, 0,11, 0,12, 0,13, 0,254,255
Classification, 15 bins

None=0
0,0,0
Heading None is represented by three zeros.

Each study always consists of three headings. For example

1,4, 0,0,0,1,0,2,0,254,255,	Heading 1
0,0,0,	Heading 2
0,0,0	Heading 3

In each study, Heading 1 seems the only one actually recorded.

Data:

- After study definitions and a separation mark, bin data follows.
- The first byte is always the counter for each record that corresponds to a time slot. If the data interval is 60min, the counter sequence goes, 1,2,3,…,23,0. If the start time is not midnight, the counter starts with the corresponding hour.
- In the bin data, if the byte data has MSB=1000 0000 then it needs to read one more byte and should be read. Suppose that val1 and val2 are the read bytes, then the magnitude is computed as: num = (val1 And &H3F) * 256 + val2. Only six bits of the high order byte is used. If the value exceeds six bit, three byte format is used as described in the next bullet.
- Large values are in three byte sequence. Let val1, val2, val3 are data sequence. If (val1 and &Hcf)=1100 0000, then num=val2 * 256 + val3. Example, 192,128,64. In this case, since 192=&HC0, num=128*256 + 64.

- At the end of a set bin data, a separation mark 0 is inserted.

Data format

The key for data decoding is determining how may each types of data saved in the bins. There are several cases. In some cases, bins are combined for all lanes, which is referred to as "combined." Here are several cases observed.

Vol: 2 * numArray
Spd: numLanes * spdBins
Cls: numLanes * clsBins

Vol: numArray
Cls: numLanes * clsBins

Vol: 2 * numArray
Spd: spdBins (combined)

Vol: 2 * numArray
Spd: spdBins (combined)
Cls: numLanes * clsBins

The number of data points can only be determined by checking different possibilities of combinations by confirming through the hour counter.

Index	Bytes	Typical value	Description
41	1	0,1,2,…,23	A counter equivalent to ending hour of the data. The counter for 24th hour is 0.
42	variable	(Total number of bins) * 2	Bin data. The total number of bins is determined from the number of studies and type.
43	1	0	Separation mark
			Repeat of bin data for each time slot

Exceptions:

- When all three study types are volume, only the first two studies are in the data.

APPENDIX-B: BULLREPORT REPORT SAMPLES

Class By Hour Report

Site: TwoHarbors ID: 030 Lanes: #1 #2 #3 #4
Classification Scheme: min5.typ
From: Monday, May 26, 2008 To: Wednesday, May 28, 2008

Minnesota Department of Transportation
Office of Transportation Data & Analysis
Generated by BullReport

Classification (Number of Vehicles)

Hour	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	Total
0->1	0	58	28	0	0	0	0	0	5	0	0	0	0	0	0	0	91
1->2	0	40	13	0	0	0	0	0	13	0	0	0	0	0	0	0	66
2->3	0	34	9	0	0	0	0	0	9	0	0	0	0	0	0	0	52
3->4	0	43	17	0	3	0	0	0	5	0	0	0	0	1	2	0	71
4->5	0	91	45	1	3	0	0	1	12	2	0	0	0	0	14	0	169
5->6	0	303	75	0	8	2	0	7	19	5	0	0	0	2	4	0	425
6->7	0	645	218	0	29	4	2	5	29	7	0	0	0	2	7	0	948
7->8	0	874	266	2	32	8	2	2	35	9	0	0	0	7	24	0	1261
8->9	0	647	313	1	30	7	1	6	41	5	0	2	0	4	10	0	1067
9->10	0	791	357	3	33	10	1	11	51	5	0	0	0	5	11	0	1278
10->11	1	842	367	2	29	12	0	15	44	7	0	1	0	2	22	0	1344
11->12	2	1069	461	2	37	8	1	12	54	7	0	0	0	9	19	0	1681
12->13	0	1162	513	2	29	10	0	10	70	5	1	0	0	9	22	0	1833
13->14	1	1154	516	2	35	9	0	17	58	7	0	0	0	9	41	0	1849
14->15	0	1176	513	2	25	6	2	17	50	3	1	0	0	5	39	0	1839
15->16	1	1284	526	2	38	6	1	16	45	4	1	0	0	9	47	0	1980
16->17	1	1302	483	1	30	11	1	6	38	3	1	0	0	2	66	0	1945
17->18	0	1280	416	1	16	5	0	9	33	3	1	0	1	6	35	0	1806
18->19	1	853	330	1	20	5	0	7	30	1	0	0	0	3	52	0	1303
19->20	1	560	190	1	11	1	0	5	30	1	0	0	0	2	9	0	811
20->21	2	517	153	0	8	0	1	4	18	0	0	0	0	1	14	0	718
21->22	1	377	115	0	5	0	0	0	11	2	0	0	0	1	8	0	520
22->23	0	273	93	0	1	2	0	1	10	0	0	0	0	0	0	0	380
23->24	0	155	39	0	3	0	0	1	5	1	0	0	0	0	0	0	204
Total	11	15530	6056	23	425	106	12	152	715	77	5	3	1	79	446	0	23641
%	0.0	65.7	25.6	0.1	1.8	0.4	0.1	0.6	3.0	0.3	0.0	0.0	0.0	0.3	1.9	0.0	100.0

Speed by Hour Report

Site: TwoHarbors ID: 030 Lanes: #1 #2 #3 #4
From: Monday, May 26, 2008 To: Wednesday, May 28, 2008

Minnesota Department of Transportation
Office of Transportation Data & Analysis
Generated by BullReport

MPH Range (Number of Vehicles)

Hour	0->30	30->35	35->40	40->45	45->50	50->55	55->60	60->65	65->70	70->75	75->80	80+	Total
0->1	0	0	0	0	1	1	6	50	23	9	1	0	91
1->2	0	0	0	0	1	0	5	44	8	6	2	0	66
2->3	0	0	0	0	1	0	3	36	5	6	1	0	52
3->4	2	0	0	0	0	2	5	44	12	6	0	0	71
4->5	14	0	0	0	0	0	4	83	36	22	7	3	169
5->6	4	0	0	0	0	3	11	236	85	66	18	2	425
6->7	7	0	0	0	0	2	38	453	218	192	32	6	948
7->8	26	0	0	1	0	5	29	515	344	292	44	5	1261
8->9	11	2	1	0	2	7	25	519	286	184	27	3	1067
9->10	11	2	0	0	1	10	36	571	403	213	29	2	1278
10->11	25	1	1	0	1	6	57	546	414	253	34	6	1344
11->12	23	3	0	0	0	8	72	702	507	316	46	4	1681
12->13	21	0	1	0	1	12	68	730	554	379	57	10	1833
13->14	44	3	1	0	4	11	59	718	551	395	53	10	1849
14->15	39	1	1	0	0	6	58	762	549	371	45	7	1839
15->16	48	1	2	0	2	3	57	887	549	380	43	8	1980
16->17	67	1	0	0	0	9	40	908	518	361	36	5	1945
17->18	36	0	1	0	1	8	34	847	473	354	48	4	1806
18->19	53	0	0	0	1	5	24	611	325	241	40	3	1303
19->20	9	0	0	0	0	4	29	449	195	105	16	4	811
20->21	15	0	0	0	0	5	23	412	143	96	23	1	718
21->22	9	0	0	1	1	8	17	309	106	59	8	2	520
22->23	0	2	1	0	0	5	31	237	65	34	4	1	380
23->24	0	0	0	0	0	4	11	138	29	18	2	2	204
Total	464	16	9	2	17	124	742	10807	6398	4358	616	88	23641
%	2.0	0.1	0.0	0.0	0.1	0.5	3.1	45.7	27.1	18.4	2.6	0.4	100.0

Note: Bin 40->45 means that the range is from 40.000 to 44.999...mph.

Lane by Hour Report

Site: TwoHarbors ID: 030 Lanes: #1 #2 #3 #4
From: Monday, May 26, 2008 To: Wednesday, May 28, 2008

Minnesota Department of Transportation
Office of Transportation Data & Analysis
Generated by BullReport

Number of Vehicles

Hour	Lane-1 (N)	Lane-2 (N)	Lane-3 (S)	Lane-4 (S)	Total
0->1	47	5	2	37	91
1->2	34	5	0	27	66
2->3	32	2	1	17	52
3->4	33	4	4	30	71
4->5	67	9	16	77	169
5->6	209	29	13	174	425
6->7	389	73	60	426	948
7->8	440	85	133	603	1261
8->9	438	63	76	490	1067
9->10	448	64	113	653	1278
10->11	394	53	141	756	1344
11->12	463	84	268	866	1681
12->13	498	75	289	971	1833
13->14	527	88	302	932	1849
14->15	571	115	245	908	1839
15->16	676	123	273	908	1980
16->17	738	147	182	878	1945
17->18	727	171	144	764	1806
18->19	506	102	106	589	1303
19->20	377	43	60	331	811
20->21	365	54	27	272	718
21->22	273	37	19	191	520
22->23	203	21	15	141	380
23->24	117	6	6	75	204
Total	8572	1458	2495	11116	23641
%	36.3	6.2	10.6	47.0	100.0

Lane By Class Report

Site: 20080528 ID: 030 Lanes: #1 #2 #3 #4
Classification Scheme: min5.typ
From: Monday, May 26, 2008 To: Wednesday, May 28, 2008

Minnesota Department of Transportation
Office of Transportation Data & Analysis
Generated by BullReport

Classification (Number of Vehicles)

Lane	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	Total
Lane-1, N	5	6007	1694	15	207	49	5	71	305	41	1	3	1	26	142	0	8572
Lane-2, N	1	971	399	0	22	3	0	3	29	4	0	0	0	1	25	0	1458
Lane-3, S	0	1675	674	2	17	6	2	7	29	2	0	0	0	11	70	0	2495
Lane-4, S	5	6877	3289	6	179	48	5	71	352	30	4	0	0	41	209	0	11116
Total	11	15530	6056	23	425	106	12	152	715	77	5	3	1	79	446	0	23641
%	0.0	65.7	25.6	0.1	1.8	0.4	0.1	0.6	3.0	0.3	0.0	0.0	0.0	0.3	1.9	0.0	100.0

Lane By Error Report

Site: 20080528 ID: 030 Lanes: #1 #2 #3 #4
From: Monday, May 26, 2008 To: Wednesday, May 28, 2008

Minnesota Department of Transportation
Office of Transportation Data & Analysis
Generated by BullReport

Error Code (Number of Vehicles)

Lane	0	101	102	103	104	105	106	107	108	109	110	111	112	113	14	15	16	17	18	19	20	31	32	33	34	35	36	37	38	39	Total
Lane-1, N	88	12	6663	1836	12	255	72	3	108	399	45	0	3	3	18	189	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9708
Lane-2, N	2588	3	1125	465	0	33	9	0	0	33	3	0	0	0	3	36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4298
Lane-3, S	5392	0	1191	432	3	15	6	0	3	36	3	0	0	0	3	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7117
Lane-4, S	22415	3	6051	2739	3	222	75	12	54	438	36	0	0	0	51	300	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32399
Total	30483	18	15030	5472	18	525	162	15	165	906	87	0	3	3	75	558	0	0	0	0	0	0	0	0	0	0	0	0	0	0	53620
%	57.0	0.0	28.1	10.2	0.0	1.0	0.3	0.0	0.3	1.7	0.2	0.0	0.0	0.0	0.1	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0

ERR 0: Normal, no error
ERR 101: Upstream loop failure
ERR 102: Downstream loop failure
ERR 103: Upstream & downstream loop failure
ERR 104: Loop in wrong order
ERR 105: High/low idle level
ERR 106: Maximum number of axles exceeded
ERR 107: Zero axle detected
ERR 108: Unequal axle count
ERR 109: Zero axles on upstream WIM sensor
ERR 110: Axle sensors in wrong order
ERR 111: Axle distance too short
ERR 112: Zero axles pm downstream WIM sensor
ERR 113: Vehicle too slow
ERR 14: Memory buffer failure, queue overflow
ERR 15: Not defined
ERR 16: Axle on sensor too long
ERR 17: Upstream loop failure
ERR 18: Loop bounce
ERR 19: One axle detected
ERR 20: Unknown error
ERR 31: Off-scale hit
ERR 32: Over height
ERR 33: Significant speed change
ERR 34: Significant weight difference
ERR 35: Vehicle headway too short
ERR 36: Unequal axle detected
ERR 37: Wrong lane
ERR 38: Tailgating
ERR 39: On-scale missed

Error Vehicle By Hour Report

Site: TwoHarbors ID: 30 Lanes: #1 #2 #3 #4
From: Monday, May 26, 2008 To: Wednesday, May 28, 2008

Minnesota Department of Transportation
Office of Transportation Data & Analysis
Generated by BullReport

Error Code (Number of Vehicles)

Hour	0	101	102	103	104	105	106	107	108	109	110	111	112	113	14	15	16	17	18	19	20	31	32	33	34	35	36	37	38	39	Total
0->1	85	0	0	0	0	0	0	0	0	0	0	0	90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	175
1->2	60	12	6663	1836	12	255	72	3	108	399	45	0	3	3	18	189	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9678
2->3	39	3	1125	465	0	33	9	0	0	33	3	0	0	0	3	36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1749
3->4	70	0	1191	432	3	15	6	0	3	36	3	0	0	0	3	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1795
4->5	148	3	6051	2739	3	222	75	12	54	438	36	0	0	0	51	300	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10132
5->6	362	0	0	0	2	0	0	1	0	2	3	0	348	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	718
6->7	910	0	0	0	4	5	0	0	0	3	1	0	640	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1563
7->8	1330	4	0	0	2	5	0	4	0	14	10	0	692	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2063
8->9	1138	1	2	0	1	0	0	2	0	13	7	0	732	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1896
9->10	1628	2	0	0	2	1	0	0	0	26	9	0	831	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2499
10->11	2035	6	0	0	12	0	0	0	0	24	13	0	735	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2827
11->12	2736	8	3	0	14	0	0	6	0	80	14	0	932	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3796
12->13	2976	11	3	0	15	0	0	7	0	66	11	0	972	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4064
13->14	2988	4	6	0	36	7	0	12	0	54	22	0	1016	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4145
14->15	2753	11	0	0	26	3	0	5	0	42	21	0	1116	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3978
15->16	2721	7	0	0	19	6	0	11	0	43	19	0	1309	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4138
16->17	2296	7	0	0	15	0	0	1	0	36	50	0	1381	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3787
17->18	2076	0	0	0	5	4	0	1	0	25	24	0	1349	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3485
18->19	1502	1	0	0	6	1	0	1	0	16	103	0	984	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2614
19->20	926	1	0	0	8	1	0	0	0	7	4	0	750	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1697
20->21	687	1	0	0	3	5	0	0	0	4	7	0	661	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1368
21->22	492	0	0	0	0	0	0	0	0	0	6	0	513	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1011
22->23	356	0	0	0	0	0	0	0	0	1	0	0	378	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	735
23->24	169	0	0	0	0	0	0	0	0	0	0	0	220	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	389
Total	30483	82	15044	5472	188	563	162	66	165	1362	411	0	15652	19	75	558	0	0	0	0	0	0	0	0	0	0	0	0	0	0	70302

Hour	0	101	102	103	104	105	106	107	108	109	110	111	112	113	14	15	16	17	18	19	20	31	32	33	34	35	36	37	38	39	Total
%	43.4	0.1	21.4	7.8	0.3	0.8	0.2	0.1	0.2	1.9	0.6	0.0	22.3	0.0	0.1	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0

ERR 0: Normal, no error
ERR 101: Upstream loop failure
ERR 102: Downstream loop failure
ERR 103: Upstream & downstream loop failure
ERR 104: Loop in wrong order
ERR 105: High/low idle level
ERR 106: Maximum number of axles exceeded
ERR 107: Zero axle detected
ERR 108: Unequal axle count
ERR 109: Zero axles on upstream WIM sensor
ERR 110: Axle sensors in wrong order
ERR 111: Axle distance too short
ERR 112: Zero axles pm downstream WIM sensor
ERR 113: Vehicle too slow
ERR 14: Memory buffer failure, queue overflow
ERR 15: Not defined
ERR 16: Axle on sensor too long
ERR 17: Upstream loop failure
ERR 18: Loop bounce
ERR 19: One axle detected
ERR 20: Unknown error
ERR 31: Off-scale hit
ERR 32: Over height
ERR 33: Significant speed change
ERR 34: Significant weight difference
ERR 35: Vehicle headway too short
ERR 36: Unequal axle detected
ERR 37: Wrong lane
ERR 38: Tailgating
ERR 39: On-scale missed

Class By Day Report

Site: TwoHarbors ID: 030 Lanes: #1 #2 #3 #4
Classification Scheme: min5.typ
From: Monday, May 26, 2008 To: Wednesday, May 28, 2008

Minnesota Department of Transportation
Office of Transportation Data & Analysis
Generated by BullReport

Classification(Number of Vehicles)

Day	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	Total
26 Mon	3	5332	2279	5	78	7	3	46	97	8	4	0	0	28	139	0	8029
27 Tue	2	5188	1953	12	172	45	4	51	316	40	1	2	0	26	121	0	7933
28 Wed	6	5010	1824	6	175	54	5	55	302	29	0	1	1	25	186	0	7679
Total	11	15530	6056	23	425	106	12	152	715	77	5	3	1	79	446	0	23641
%	0.0	65.7	25.6	0.1	1.8	0.4	0.1	0.6	3.0	0.3	0.0	0.0	0.0	0.3	1.9	0.0	100.0
Avg	4	5177	2019	8	142	35	4	51	238	26	2	1	0	26	149	0	7880

Truck Count By Day Report

Site: TwoHarbors ID: 030 Lanes: #1 #2 #3 #4
Classification Scheme: min5.typ
From: Monday, May 26, 2008 To: Wednesday, May 28, 2008

Minnesota Department of Transportation
Office of Transportation Data & Analysis
Generated by BullReport

Classification (Number of Vehicles)

Day	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C16	Total
26 Mon	5	78	7	3	46	97	8	4	0	0	0	248
27 Tue	12	172	45	4	51	316	40	1	2	0	0	643
28 Wed	6	175	54	5	55	302	29	0	1	1	0	628
Total	23	425	106	12	152	715	77	5	3	1	0	1519
%	1.5	28.0	7.0	0.8	10.0	47.1	5.1	0.3	0.2	0.1	0.0	100.0
Avg	8	142	35	4	51	238	26	2	1	0	0	506

If Type-16 is present, it was added to Type-9.

Class By Front Axle Weight Report

Site: TwoHarbors ID: 030 Lanes: #1 #2 #3 #4
Classification Scheme: min5.typ
From: Monday, May 26, 2008 To: Wednesday, May 28, 2008

Minnesota Department of Transportation
Office of Transportation Data & Analysis
Generated by BullReport

Classification (Number of Steer Axles)

Kips	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	Total
0->2	11	11961	838	0	0	0	0	0	0	0	0	0	0	27	2	0	12839
2->4	0	3569	5069	0	125	0	0	41	3	1	3	0	0	42	0	0	8853
4->6	0	0	149	2	160	3	3	31	9	5	2	0	0	2	0	0	366
6->8	0	0	0	5	97	18	2	46	120	25	0	0	0	0	0	0	313
8->10	0	0	0	2	34	30	0	21	331	22	0	0	0	0	0	0	440
10->12	0	0	0	2	8	26	1	6	239	20	0	0	0	3	0	0	305
12->14	0	0	0	5	1	18	3	3	13	1	0	0	0	2	0	0	46
14->16	0	0	0	3	0	8	2	3	0	1	0	2	1	0	0	0	20
16->18	0	0	0	1	0	1	1	1	0	2	0	1	0	1	0	0	8
18->20	0	0	0	3	0	0	0	0	0	0	0	0	0	1	0	0	4
20->22	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	2
22->24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24->26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26->28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28->30	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
30->32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32->34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34->36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36->38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38->40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40 +	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	11	15530	6056	23	425	106	12	152	715	77	5	3	1	79	2	0	23197
%	0.0	66.9	26.1	0.1	1.8	0.5	0.1	0.7	3.1	0.3	0.0	0.0	0.0	0.3	0.0	0.0	100.0

Note: Vehicle fragments are excluded in this report.

Class By Gross Vehicle Weight Report

Site: TwoHarbors ID: 030 Lanes: #1 #2 #3 #4
Classification Scheme: min5.typ
From: Monday, May 26, 2008 To: Wednesday, May 28, 2008

Minnesota Department of Transportation
Office of Transportation Data & Analysis
Generated by BullReport

Classification (Number of Vehicles)

Kips	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	Total
0->2	11	992	23	0	0	0	0	0	0	0	0	0	0	13	0	0	1039
2->4	0	12526	1397	0	1	0	0	0	0	0	0	0	0	8	0	0	13932
4->6	0	1894	3190	0	26	0	0	0	0	0	0	0	0	18	0	0	5128
6->8	0	95	969	0	78	0	0	10	0	0	0	0	0	14	0	0	1166
8->10	0	19	269	0	59	0	0	15	0	0	0	0	0	7	0	0	369
10->12	0	3	106	1	51	0	0	11	0	0	0	0	0	3	0	0	175
12->14	0	1	47	3	54	1	0	9	2	0	0	0	0	3	0	0	120
14->16	0	0	24	1	59	2	0	12	2	0	4	0	0	2	0	0	106
16->18	0	0	17	1	26	8	1	5	1	1	0	0	0	0	0	0	60
18->20	0	0	7	2	30	9	1	10	6	1	1	0	0	0	0	0	67
20->22	0	0	5	0	20	4	1	9	11	0	0	0	0	0	0	0	50
22->24	0	0	2	0	7	8	0	7	18	1	0	0	0	0	0	0	43
24->26	0	0	0	1	9	6	1	12	51	2	0	0	0	1	0	0	83
26->28	0	0	0	0	3	7	1	10	47	2	0	0	0	1	0	0	71
28->30	0	0	0	0	0	7	0	9	49	6	0	0	0	0	0	0	71
30->32	0	0	0	0	2	8	0	6	51	6	0	0	0	0	0	0	73
32->34	0	0	0	0	0	9	0	4	32	3	0	0	0	0	0	0	48
34->36	0	0	0	0	0	9	0	6	17	2	0	0	0	1	0	0	35
36->38	0	0	0	0	0	8	0	2	26	1	0	0	0	0	0	0	37
38->40	0	0	0	1	0	3	0	3	13	2	0	0	0	1	0	0	23
40->42	0	0	0	0	0	5	0	4	15	2	0	0	0	0	0	0	26
42->44	0	0	0	2	0	2	0	2	8	1	0	0	0	0	0	0	15
44->46	0	0	0	2	0	4	1	1	6	2	0	0	0	0	0	0	16
46->48	0	0	0	2	0	2	0	1	6	1	0	0	0	1	0	0	13
48->50	0	0	0	1	0	0	1	1	9	0	0	0	0	0	0	0	12
50->52	0	0	0	0	0	1	0	0	7	0	0	0	0	1	0	0	9
52->54	0	0	0	3	0	3	2	0	12	0	0	1	0	0	0	0	21
54->56	0	0	0	1	0	0	0	0	12	1	0	0	0	0	0	0	14
56->58	0	0	0	0	0	0	1	1	19	1	0	0	0	1	0	0	23

Kips	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	Total
58->60	0	0	0	1	0	0	1	0	19	0	0	0	0	0	0	0	21
60->62	0	0	0	0	0	0	1	0	17	0	0	0	0	0	0	0	18
62->64	0	0	0	0	0	0	0	0	19	0	0	1	0	0	0	0	20
64->66	0	0	0	0	0	0	0	1	43	6	0	0	0	0	0	0	50
66->68	0	0	0	0	0	0	0	0	45	2	0	0	0	2	0	0	49
68->70	0	0	0	0	0	0	0	1	39	2	0	0	0	1	0	0	43
70->72	0	0	0	0	0	0	0	0	26	4	0	0	0	0	0	0	30
72->74	0	0	0	1	0	0	0	0	19	3	0	1	0	0	0	0	24
74->76	0	0	0	0	0	0	0	0	21	4	0	0	0	0	0	0	25
76->78	0	0	0	0	0	0	0	0	11	8	0	0	0	0	0	0	19
78->80	0	0	0	0	0	0	0	0	8	2	0	0	0	0	0	0	10
80->82	0	0	0	0	0	0	0	0	7	1	0	0	0	0	0	0	8
82->84	0	0	0	0	0	0	0	0	6	3	0	0	0	0	0	0	9
84->86	0	0	0	0	0	0	0	0	4	2	0	0	0	0	0	0	6
86->88	0	0	0	0	0	0	0	0	8	1	0	0	0	0	0	0	9
88->90	0	0	0	0	0	0	0	0	3	0	0	0	0	1	0	0	4
90->92	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
92->94	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
94->96	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
96->98	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
98->100	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
100 +	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
Total	11	15530	6056	23	425	106	12	152	715	77	5	3	1	79	0	0	23195
%	0.0	67.0	26.1	0.1	1.8	0.5	0.1	0.7	3.1	0.3	0.0	0.0	0.0	0.3	0.0	0.0	100.0

Note: Vehicle fragments are excluded in this report.

Weight Violations By Class Report

Site: TwoHarbors ID: 030 Lanes: #1 #2 #3 #4
Classification Scheme: min5.typ
From: Monday, May 26, 2008 To: Wednesday, May 28, 2008

Minnesota Department of Transportation
Office of Transportation Data & Analysis
Generated by BullReport

Weight Violations

Class	Valid	Viol	% Viol	Single	Tandem	Tridem	Quadem	GVW	Bridge
C1	11	0	0	0	0	0	0	0	0
C2	15530	0	0	0	0	0	0	0	0
C3	6056	0	0	0	0	0	0	0	0
C4	20	3	15	0	1	0	0	0	3
C5	424	1	0	1	0	0	0	0	0
C6	99	7	7	2	5	0	0	0	5
C7	11	1	9	1	0	0	0	0	0
C8	150	2	1	0	2	0	0	0	2
C9	666	49	7	3	39	0	0	28	46
C10	66	11	17	0	2	0	0	11	7
C11	5	0	0	0	0	0	0	0	0
C12	3	0	0	0	0	0	0	0	0
C13	0	1	0	0	0	0	0	1	1
C14	76	3	4	3	0	0	0	1	0
C15	444	2	0	0	0	0	0	0	2
C16	0	0	0	0	0	0	0	0	0
Total	23561	80	0	10	49	0	0	41	66

Note 1: Valid, Viol, GVW, and Bridge coulumns are valid or violated vehicle counts.
Note 2: Single, Tandem, Tridem, and Quadem coulumns are violated axle group counts.
Note 3: Violation weights: Single>20, Tandem>36, Tridem>51, Quad>68, GVW>80 kips

Weight Violations By Hour Report

Site: TwoHarbors ID: 030 Lanes: #1 #2 #3 #4
From: Monday, May 26, 2008 To: Wednesday, May 28, 2008

Minnesota Department of Transportation
Office of Transportation Data & Analysis
Generated by BullReport

Weight Violations

Hour	Valid	Viol	% Viol	Single	Tandem	Tridem	Quadem	GVW	Bridge
0->1	91	0	0	0	0	0	0	0	0
1->2	66	0	0	0	0	0	0	0	0
2->3	48	4	8	0	4	0	0	3	4
3->4	71	0	0	0	0	0	0	0	0
4->5	168	1	1	0	1	0	0	1	1
5->6	423	2	0	0	2	0	0	1	2
6->7	946	2	0	0	0	0	0	1	1
7->8	1260	1	0	0	1	0	0	1	1
8->9	1061	6	1	2	4	0	0	2	4
9->10	1273	5	0	1	2	0	0	1	4
10->11	1338	6	0	1	0	0	0	2	4
11->12	1674	7	0	1	4	0	0	5	5
12->13	1828	5	0	3	2	0	0	0	3
13->14	1840	9	0	0	8	0	0	7	8
14->15	1837	2	0	0	0	0	0	1	1
15->16	1972	8	0	1	7	0	0	5	7
16->17	1945	0	0	0	0	0	0	0	0
17->18	1799	7	0	0	4	0	0	3	7
18->19	1299	4	0	0	3	0	0	2	4
19->20	807	4	0	1	4	0	0	2	3
20->21	715	3	0	0	1	0	0	1	3
21->22	517	3	1	0	1	0	0	2	3
22->23	379	1	0	0	1	0	0	1	1
23->24	204	0	0	0	0	0	0	0	0
Total	23561	80	0	10	49	0	0	41	66

Note 1: Valid, Viol, GVW, and Bridge coulumns are valid or violated vehicle counts.
Note 2: Single, Tandem, Tridem, and Quadem coulumns are violated axle group counts.
Note 3: Violation weights: Single>20, Tandem>36, Tridem>51, Quad>68, GVW>80 kips

ESAL By Hour and Class Report

Site: TwoHarbors ID: 030 Lanes: #1 #2 #3 #4
Classification Scheme: min5.typ
From: Monday, May 26, 2008 To: Wednesday, May 28, 2008

Minnesota Department of Transportation
Office of Transportation Data & Analysis
Generated by BullReport

Classification(ESAL)

Hour	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	Total
0->1	0.00	0.00	0.00	0.00	0.00	2.97	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.97
1->2	0.00	0.00	0.00	0.00	0.00	5.21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.21
2->3	0.00	0.00	0.00	0.00	0.00	16.46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	16.46
3->4	0.00	0.09	0.00	0.00	0.00	3.43	0.00	0.00	0.00	0.00	0.13	0.00	0.00	3.64
4->5	0.62	0.62	0.00	0.00	0.01	5.51	5.05	0.00	0.00	0.00	0.00	0.00	0.00	11.82
5->6	0.00	0.65	0.52	0.00	3.98	13.24	7.72	0.00	0.00	0.00	1.11	0.00	0.00	27.23
6->7	0.00	2.47	0.87	0.97	2.07	18.75	12.85	0.00	0.00	0.00	0.00	0.00	0.00	37.99
7->8	0.67	1.24	3.00	2.81	1.37	21.84	2.41	0.00	0.00	0.00	1.36	0.00	0.00	34.71
8->9	0.80	3.85	4.19	1.96	2.67	22.48	5.17	0.00	5.47	0.00	0.00	0.00	0.00	46.59
9->10	0.92	7.09	3.74	0.80	1.42	39.14	12.15	0.00	0.00	0.00	0.01	0.00	0.00	65.27
10->11	0.78	2.84	4.60	0.00	3.13	38.18	5.33	0.00	0.53	0.00	0.01	0.00	0.00	55.39
11->12	3.40	2.47	2.64	0.60	5.92	31.89	14.43	0.00	0.00	0.00	10.53	0.00	0.00	71.87
12->13	1.21	4.34	7.30	0.00	2.48	48.96	1.79	0.01	0.00	0.00	1.84	0.00	0.00	67.94
13->14	2.49	2.46	2.93	0.00	7.02	44.69	19.16	0.00	0.00	0.00	0.04	0.00	0.00	78.81
14->15	1.72	1.84	0.85	0.49	1.38	23.48	2.15	0.02	0.00	0.00	0.03	0.00	0.00	31.96
15->16	1.95	2.38	4.89	0.03	2.39	40.49	3.53	0.05	0.00	0.00	20.54	0.00	0.00	76.23
16->17	0.75	2.28	3.77	0.17	0.74	25.86	2.42	0.01	0.00	0.00	0.00	0.00	0.00	36.01
17->18	1.25	0.85	5.28	0.00	0.45	32.73	2.52	0.01	0.00	5.02	1.81	0.00	0.00	49.92
18->19	0.01	0.90	0.90	0.00	1.46	31.21	2.27	0.00	0.00	0.00	11.97	0.00	0.00	48.71
19->20	0.20	0.76	0.42	0.00	0.31	26.47	0.13	0.00	0.00	0.00	15.06	0.00	0.00	43.36
20->21	0.00	0.59	0.00	0.05	0.88	16.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	17.78
21->22	0.00	0.15	0.00	0.00	0.00	12.64	0.07	0.00	0.00	0.00	0.00	0.00	0.00	12.86
22->23	0.00	0.61	0.13	0.00	0.10	9.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.10
23->24	0.00	0.01	0.00	0.00	0.00	3.42	2.16	0.00	0.00	0.00	0.00	0.00	0.00	5.59
Total	16.78	38.50	46.03	7.89	37.77	534.55	101.31	0.10	6.00	5.02	64.44	0.00	0.00	858.40
%	1.95	4.48	5.36	0.92	4.40	62.27	11.80	0.01	0.70	0.59	7.51	0.00	0.00	100.00

Pavement: Flexible, SN=5, Pt=2.5

Axle Count By Axle Weight Report

Site: TwoHarbors ID: 030 Lanes: #1 #2 #3 #4
Classification Scheme: min5.typ
From: Monday, May 26, 2008 To: Wednesday, May 28, 2008

Minnesota Department of Transportation
Office of Transportation Data & Analysis
Generated by BullReport

Single Axles (Count, Tot Weight, ESAL)

Wt Range (Kips)	Count	Tot Wt (Kips)	ESAL
0->2	30383.00	42476.75	2.28
2->4	14888.00	37981.47	7.19
4->6	1546.00	7469.91	7.39
6->8	937.00	6523.90	18.86
8->10	805.00	7210.10	45.63
10->12	668.00	7309.87	87.32
12->14	665.00	8672.15	180.44
14->16	572.00	8519.00	267.97
16->18	276.00	4664.13	215.82
18->20	109.00	2053.13	131.06
20->22	27.00	553.91	45.21
22->24	4.00	88.72	9.00
24->26	0.00	0.00	0.00
26->28	1.00	27.18	4.82
30+	2.00	61.46	15.38
Total	50883.00	133611.68	1038.37

ESAL was computed for single axles.
Pavement: Flexible, SN=5, Pt=2.5

Single Axle Loads by Class Report

Site: TwoHarbors ID: 030 Lanes: #1 #2 #3 #4
Classification Scheme: min5.typ
From: Monday, May 26, 2008 To: Wednesday, May 28, 2008

Minnesota Department of Transportation
Office of Transportation Data & Analysis
Generated by BullReport

Classification (Number of Single Axles)

Kips	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	Total
0->2	0	21102	3890	1	4	0	2	62	3	0	3	0	0	53	2	0	25122
2->4	0	4937	8903	0	226	0	6	103	109	5	17	0	0	76	0	0	14382
4->6	0	7	439	4	231	3	6	76	60	6	4	0	0	9	0	0	845
6->8	0	0	44	8	175	18	2	80	144	28	1	0	0	5	0	0	505
8->10	0	0	1	3	114	30	1	52	341	26	0	2	0	2	0	0	572
10->12	0	0	0	5	50	26	2	25	254	33	0	0	0	2	0	0	397
12->14	0	0	0	7	27	18	4	14	67	9	0	2	0	2	0	0	150
14->16	0	0	0	6	16	8	3	19	77	4	0	3	2	3	0	0	141
16->18	0	0	0	1	5	1	1	9	34	6	0	1	1	1	0	0	60
18->20	0	0	0	3	1	0	1	1	15	3	0	0	1	4	0	0	29
20->22	0	0	0	0	1	2	1	0	3	0	0	0	0	1	0	0	8
22->24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24->26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26->28	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
28->30	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
30->32	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
32->34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34->36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36->38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38->40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40 +	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	0	26046	13277	38	850	106	29	441	1107	120	25	8	4	161	2	0	42214
%	0.0	61.7	31.5	0.1	2.0	0.3	0.1	1.0	2.6	0.3	0.1	0.0	0.0	0.4	0.0	0.0	100.0

Note: Vehicle fragments are excluded in this report.

Tandem Axle Loads by Class Report

Site: TwoHarbors ID: 030 Lanes: #1 #2 #3 #4
Classification Scheme: min5.typ
From: Monday, May 26, 2008 To: Wednesday, May 28, 2008

Minnesota Department of Transportation
Office of Transportation Data & Analysis
Generated by BullReport

Classification (Number of Tandem Axles)

Kips	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	Total
0->2	11	298	0	0	0	0	0	0	0	0	0	0	0	0	0	0	309
2->4	0	2310	3	0	0	0	0	2	2	1	0	0	0	5	0	0	2323
4->6	0	62	2	0	0	1	0	0	21	0	0	0	0	3	0	0	89
6->8	0	1	1	0	0	6	0	3	80	3	0	0	0	1	0	0	95
8->10	0	0	1	0	0	8	0	11	139	6	0	0	0	2	0	0	167
10->12	0	0	2	0	0	7	0	5	160	7	0	0	0	0	0	0	181
12->14	0	0	1	0	0	7	1	5	97	8	0	0	0	0	0	0	119
14->16	0	0	0	1	0	8	1	4	62	7	0	0	0	0	0	0	83
16->18	0	0	0	1	0	6	0	4	36	3	0	0	0	1	0	0	51
18->20	0	0	0	1	0	9	0	4	34	4	0	0	0	1	0	0	53
20->22	0	0	0	0	0	13	0	2	28	6	0	0	0	3	0	0	52
22->24	0	0	0	0	0	15	0	0	62	12	0	0	0	0	0	0	89
24->26	0	0	0	0	0	6	0	0	93	8	0	0	0	2	0	0	109
26->28	0	0	0	2	0	5	0	1	110	16	0	0	0	1	0	0	135
28->30	0	0	0	0	0	3	0	1	121	9	0	0	0	0	0	0	134
30->32	0	0	0	0	0	2	0	1	68	9	0	0	1	0	0	0	81
32->34	0	0	0	1	0	5	0	0	42	5	0	0	0	0	0	0	53
34->36	0	0	0	0	0	0	0	0	40	3	0	0	1	0	0	0	44
36->38	0	0	0	0	0	4	0	0	26	0	0	0	0	0	0	0	30
38->40	0	0	0	1	0	0	0	0	12	2	0	0	0	0	0	0	15
40->42	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
42->44	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	3
44->46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46->48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
48->50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50->52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
52->54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
54->56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
56->58	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Kips	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	Total
58->60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
60->62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
62->64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
64->66	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
66->68	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
68->70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
70->72	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
72->74	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
74->76	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
76->78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78->80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
80 +	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	11	2671	10	7	0	106	2	45	1234	109	0	0	2	19	0	0	4216
%	0.3	63.4	0.2	0.2	0.0	2.5	0.0	1.1	29.3	2.6	0.0	0.0	0.0	0.5	0.0	0.0	100.0

Note: Vehicle fragments are excluded in this report.

Tridem Axle Loads by Class Report

Site: TwoHarbors ID: 030 Lanes: #1 #2 #3 #4
Classification Scheme: min5.typ
From: Monday, May 26, 2008 To: Wednesday, May 28, 2008

Minnesota Department of Transportation
Office of Transportation Data & Analysis
Generated by BullReport

Classification (Number of Tridem Axles)

Kips	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	Total
0->2	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	7
2->4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4->6	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	2
6->8	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	2
8->10	0	0	4	0	0	0	0	0	0	8	0	0	0	0	0	0	12
10->12	0	0	2	0	0	0	0	0	0	9	0	0	0	0	0	0	11
12->14	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	2
14->16	0	0	0	0	0	0	0	0	0	2	0	0	0	1	0	0	3
16->18	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
18->20	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	2
20->22	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
22->24	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
24->26	0	0	0	0	0	0	1	0	0	1	0	1	0	1	0	0	4
26->28	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	3
28->30	0	0	0	1	0	0	2	0	0	3	0	0	0	0	0	0	6
30->32	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	2
32->34	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
34->36	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	3
36->38	0	0	0	0	0	0	1	0	0	3	0	0	0	0	0	0	4
38->40	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	3
40->42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42->44	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	2
44->46	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
46->48	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
48->50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50->52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
52->54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
54->56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
56->58	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Kips	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	Total
58->60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
60->62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
62->64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
64->66	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
66->68	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
68->70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
70->72	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
72->74	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
74->76	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
76->78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78->80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
80->82	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
82->84	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
84->86	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
86->88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
88->90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
90->92	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
92->94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
94->96	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
96->98	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
98->100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100 +	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	0	0	8	7	0	0	7	0	0	39	0	2	0	11	0	0	74
%	0.0	0.0	10.8	9.5	0.0	0.0	9.5	0.0	0.0	52.7	0.0	2.7	0.0	14.9	0.0	0.0	100.0

Note: Vehicle fragments are excluded in this report.

Quadem Axle Loads by Class Report

Site: TwoHarbors ID: 030 Lanes: #1 #2 #3 #4
Classification Scheme: min5.typ
From: Monday, May 26, 2008 To: Wednesday, May 28, 2008

Minnesota Department of Transportation
Office of Transportation Data & Analysis
Generated by BullReport

Classification (Number of Quadem Axles)

Kips	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	Total
0->2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2->4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4->6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6->8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8->10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10->12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12->14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14->16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16->18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18->20	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
20->22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22->24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24->26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26->28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28->30	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
30->32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32->34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34->36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36->38	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
38->40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40->42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42->44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
44->46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46->48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
48->50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50->52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
52->54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
54->56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
56->58	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Kips	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	Total
58->60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
60->62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
62->64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
64->66	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
66->68	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
68->70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
70->72	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
72->74	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
74->76	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
76->78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78->80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
80->82	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
82->84	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
84->86	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
86->88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
88->90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
90->92	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
92->94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
94->96	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
96->98	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
98->100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100->102	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
102->104	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
104->106	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
106->108	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
108->110	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
110 +	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	0	0	0	0	0	0	0	0	0	2	0	1	0	0	0	0	3
%	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	66.7	0.0	33.3	0.0	0.0	0.0	0.0	100.0

Note: Vehicle fragments are excluded in this report.

Load Spectra Report

Site: TwoHarbors ID: 030 Lanes: #1 #2 #3 #4
Classification Scheme: min5.typ
From: Monday, May 26, 2008 To: Wednesday, May 28, 2008
Veh Types: #1 #2 #3 #4 #5 #6 #7 #8 #9 #10 #11 #12 #13 #14 #15 #16

Minnesota Department of Transportation
Office of Transportation Data & Analysis
Generated by BullReport

Axle Group (Number of Axle Groups)

Kips	Single	Tandem	Tridem	Quadem	Steer	GVW	Total
0->2	25126	309	7	0	12839	1039	39320
2->4	14382	2323	0	0	8853	13932	39490
4->6	845	89	2	0	366	5128	6430
6->8	505	95	2	0	313	1166	2081
8->10	572	167	12	0	440	369	1560
10->12	397	181	11	0	305	175	1069
12->14	150	119	2	0	46	120	437
14->16	141	83	3	0	20	106	353
16->18	60	51	1	0	8	60	180
18->20	29	53	2	1	4	67	156
20->22	8	52	1	0	2	50	113
22->24	0	89	1	0	0	43	133
24->26	0	109	4	0	0	83	196
26->28	1	135	3	0	0	71	210
28->30	1	134	6	1	1	71	214
30->32	1	81	2	0	0	73	157
32->34	0	53	1	0	0	48	102
34->36	0	44	3	0	0	35	82
36->38	0	30	4	1	0	37	72
38->40	0	15	3	0	0	23	41
40->42	0	1	0	0	0	26	27
42->44	0	3	2	0	0	15	20
44->46	0	0	1	0	0	16	17
46->48	0	0	1	0	0	13	14
48->50	0	0	0	0	0	12	12
50->52	0	0	0	0	0	9	9
52->54	0	0	0	0	0	21	21
54->56	0	0	0	0	0	14	14

Kips	Single	Tandem	Tridem	Quadem	Steer	GVW	Total
56->58	0	0	0	0	0	23	23
58->60	0	0	0	0	0	21	21
60->62	0	0	0	0	0	18	18
62->64	0	0	0	0	0	20	20
64->66	0	0	0	0	0	50	50
66->68	0	0	0	0	0	49	49
68->70	0	0	0	0	0	43	43
70->72	0	0	0	0	0	30	30
72->74	0	0	0	0	0	24	24
74->76	0	0	0	0	0	25	25
76->78	0	0	0	0	0	19	19
78->80	0	0	0	0	0	10	10
80->82	0	0	0	0	0	8	8
82->84	0	0	0	0	0	9	9
84->86	0	0	0	0	0	6	6
86->88	0	0	0	0	0	9	9
88->90	0	0	0	0	0	4	4
90->92	0	0	0	0	0	0	0
92->94	0	0	0	0	0	1	1
94->96	0	0	0	0	0	1	1
96->98	0	0	0	0	0	1	1
98->100	0	0	0	0	0	1	1
100 +	0	0	0	0	0	1	1
Total	42218	4216	74	3	23197	23195	92903
%	45.4	4.5	0.1	0.0	25.0	25.0	100.0

Speed By Class Report

Site: TwoHarbors ID: 030 Lanes: #1 #2 #3 #4
Classification Scheme: min5.typ
From: Monday, May 26, 2008 To: Wednesday, May 28, 2008

Minnesota Department of Transportation
Office of Transportation Data & Analysis
Generated by BullReport

Classification(Number of Vehicles)

MPH	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	Total
0->30	0	13	6	0	1	0	0	0	0	0	0	0	0	0	444	0	464
30->35	0	12	3	0	1	0	0	0	0	0	0	0	0	0	0	0	16
35->40	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9
40->45	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2
45->50	0	12	5	0	0	0	0	0	0	0	0	0	0	0	0	0	17
50->55	0	81	35	0	2	0	0	2	1	1	0	0	0	2	0	0	124
55->60	1	440	216	0	17	4	0	1	57	2	0	0	0	4	0	0	742
60->65	8	7430	2415	20	273	64	9	101	393	48	3	3	1	38	1	0	10807
65->70	2	4075	1911	2	89	32	2	36	201	20	2	0	0	26	0	0	6398
70->75	0	2960	1269	1	37	6	0	10	63	6	0	0	0	6	0	0	4358
75->80	0	431	178	0	5	0	0	1	0	0	0	0	0	1	0	0	616
80+	0	66	18	0	0	0	1	1	0	0	0	0	0	1	1	0	88
Total	11	15530	6056	23	425	106	12	152	715	77	5	3	1	79	446	0	23641
%	0.0	65.7	25.6	0.1	1.8	0.4	0.1	0.6	3.0	0.3	0.0	0.0	0.0	0.3	1.9	0.0	100.0

Note: Bin 40->45 means that the range is from 40.000 to 44.999...mph.

Site Summary Report

Site: TwoHarbors   ID: 030   Lanes: #1 #2 #3 #4
Classification Scheme: min5.typ
From: Monday, May 26, 2008   To: Wednesday, May 28, 2008

Minnesota Department of Transportation
Office of Transportation Data & Analysis
Generated by BullReport

Classification

Data Type	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	Total
Veh Cnt	11	15530	6056	23	425	106	12	152	715	77	5	3	1	79	446	0	23641
% Cnt	0.0	65.7	25.6	0.1	1.8	0.4	0.1	0.6	3.0	0.3	0.0	0.0	0.0	0.3	1.9	0.0	100.0
Err Cnt	10	6161	1775	16	210	51	6	73	321	43	1	3	1	55	333	0	9059
% Err	0.1	68.0	19.6	0.2	2.3	0.6	0.1	0.8	3.5	0.5	0.0	0.0	0.0	0.6	3.7	0.0	100.0
Valid Wt	11	15530	6056	23	424	104	11	152	686	66	5	3	0	76	446	0	23593
TGW	9	47816	32071	852	5324	3206	488	3377	34714	4391	79	191	133	962	0	0	133612
Avg GVW	0.81	3.08	5.30	37.02	12.53	30.25	40.64	22.22	48.55	57.02	15.78	63.54	132.86	12.17	0.00	0.00	481.77
Tot ESAL	0.00	3.28	7.60	16.78	38.50	46.03	7.89	37.77	534.55	61.00	0.04	3.26	5.02	32.26	0.00	0.00	794.0
Avg ESAL	0.00	0.00	0.00	0.73	0.09	0.43	0.66	0.25	0.75	0.79	0.01	1.09	5.02	0.41	0.00	0.00	10.23
Ovr Wt	0	0	0	0	1	2	1	0	29	11	0	0	1	3	0	0	48
% Ovwt	0.0	0.0	0.0	0.0	0.2	1.9	8.3	0.0	4.1	14.3	0.0	0.0	100.0	3.8	0.0	0.0	132.6
Avg FrAxWt	0.28	1.72	2.58	11.88	5.29	10.49	10.49	6.29	9.32	8.91	3.86	15.70	15.52	3.40	0.00	0.00	105.7

Pavement: Flexible, SN=5, Pt=2.5
All weights (Valid Wt, TGW, Avg GVW, and Avg FrAxWt) are in Kips.

APPENDIX-C: BULLPIEZO SAMPLE OUTPUTS

AADT-101-2008.txt

AADT-101-2008.txt - Notepad

```
AADT,2008,101,dow=7
Class-1,Class-2,Class-3,Class-4,Class-5,Class-6,Class-7,Class-8
26373.7400,427.5345,197.4147,99.1690,184.1710,1039.0600,67.4945,17.5443
```

MADT-101-2008.txt

MADT-101-2008.txt - Notepad

```
MADT,2008,101,dow=5
Month,Class-1,Class-2,Class-3,Class-4,Class-5,Class-6,Class-7,Class-8
1,27215.3300,521.5400,178.6500,82.6070,153.4130,1206.8600,42.4700,21.9800
2,28974.3600,528.2367,158.1800,89.1508,165.5658,1205.5200,43.3500,20.9367
3,29180.1800,512.6666,164.7500,86.5783,160.7883,1263.9170,60.3500,23.4333
4,28976.1300,511.0500,192.0500,103.0155,191.3145,1295.9000,67.9100,25.8700
5,28115.7800,522.6334,254.6000,183.5633,340.9033,1365.5500,86.0833,24.7667
6,19869.7000,447.7000,268.0000,145.6525,270.4975,1076.9000,83.1500,19.9500
7,26954.6500,515.0500,288.8000,172.0600,319.5400,1328.1000,113.6000,20.9500
8,31726.4000,557.0000,362.5000,126.4900,234.9100,1350.6000,102.8000,23.5000
9,32352.0500,585.1500,340.1500,140.2625,260.4875,1442.8500,117.9500,22.7000
10,31699.0900,583.4100,330.2200,137.6690,255.6709,1468.6700,114.2300,23.3100
11,29736.7800,568.6000,284.5100,116.5185,216.3915,1296.6200,101.9300,20.1200
12,27911.6900,535.3200,201.7900,78.3860,145.5740,1210.2300,54.7933,22.9767
```

MAF-101-2008.txt

MAF-101-2008.txt - Notepad

```
MAF,2008,101,dow=5
Type,Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec
1,0.9691,0.9102,0.9038,0.9102,0.9380,1.3273,0.9784,0.8313,0.8152,0.8320,0.8869,0.9449
2,0.8198,0.8094,0.8339,0.8366,0.8180,0.9550,0.8301,0.7676,0.7306,0.7328,0.7519,0.7987
3,1.1050,1.2480,1.1983,1.0279,0.7754,0.7366,0.6836,0.5446,0.5804,0.5978,0.6939,0.9783
4,1.2005,1.1124,1.1454,0.9627,0.5402,0.6809,0.5764,0.7840,0.7070,0.7203,0.8511,1.2651
5,1.2005,1.1124,1.1454,0.9627,0.5402,0.6809,0.5764,0.7840,0.7070,0.7203,0.8511,1.2651
6,0.8610,0.8619,0.8221,0.8018,0.7609,0.9649,0.7824,0.7693,0.7201,0.7075,0.8014,0.8586
7,1.5892,1.5570,1.1184,0.9939,0.7841,0.8117,0.5941,0.6566,0.5722,0.5909,0.6622,1.2318
8,0.7982,0.8380,0.7487,0.6782,0.7084,0.8794,0.8374,0.7466,0.7729,0.7526,0.8720,0.7636
```

availDates-101-2008.txt

availDates-101-2008.txt - Notepad

File Edit Format View Help

```
Data Available for the Period Selected
Site ID=101, Site Name=
Start:9/1/2007,   End: 12/31/2008

             Sun    Mon     Tue     Wed     Thu     Fri     Sat
September/2007
               0      0       0       0       0       0       1
               2      3       4       5       6       7       8
               9     10      11      12      13      14      15
              16     17      18      19      20      21      22
              23     24      25      26      27      28      29
              30      0       0       0       0       0       0
October/2007
               0      1       2       3       4       5       6
               7      8       9      10      11      12      13
              14     15      16      17      18      19      20
              21     22      23      24      25      26      27
              28     29      30      31       0       0       0
November/2007
               0      0       0       0       1       2       3
               0      5       6       7       8       9      10
              11     12      13      14      15      16      17
              18     19      20      21      22      23      24
              25     26      27      28      29      30       0
December/2007
               0      0       0       0       0       0       1
               2      3       4       5       6       7       8
               9     10      11      12      13      14      15
              16     17      18      19      20      21      22
              23     24      25      26      27      28      29
              30     31       0       0       0       0       0
January/2008
               0      0       1       2       3       4       5
               6      7       8       9      10      11      12
              13     14      15      16      17      18      19
              20     21      22      23      24      25      26
              27     28      29      30      31       0       0
February/2008
               0      0       0       0       0       1       2
               3      4       5       6       7       8       9
              10     11      12       0      14      15      16
              17     18      19      20      21      22      23
              24     25      26      27      28      29       0
March/2008
               0      0       0       0       0       0       1
               2      3       0       5       6       7       8
               9      0      11      12      13      14      15
              16     17      18      19      20      21      22
              23     24      25      26      27      28      29
              30     31       0       0       0       0       0
April/2008
               0      0       1       2       3       4       5
               6      7       8       9      10      11      12
              13     14      15      16      17      18      19
```