Application of an Automatic Data Acquisition System in Mass Transit

by

Srinath Raju

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
Master of Science
in
Civil Engineering

APPROVED:

__________________________  
Antoine G. Hobeika

__________________________  _____________________________
Charles E. Nunnally          Siamak A. Ardekani

October 14, 1987
Blacksburg, Virginia
Application of an Automatic Data Acquisition System in Mass Transit

by

Srinath Raju
Antoine G. Hobeika
Civil Engineering

(ABSTRACT)

A robust algorithm has been developed to do data processing accurately, removing the need for a radio signal to be imbedded for locational accuracy. This, consequently eliminates the signposts installation and maintenance costs and worries. A sensitivity analysis of the algorithm using a real life data file revealed that the matching process does not change with changes in system parameters, adding credibility to the technique used for matching in the algorithm. Next, several new programs have been added to bring together a software package yielding management reports and plots. These reports and plots are tremendous decision aiding tools and since the programs are interactive, the package is easy to use. A user’s manual has also been developed.

Finally, an implementation of the Automated Data Acquisition System at TRT, Norfolk is discussed. A systematic approach to the software development to meet the needs of the transit property has been conceptualized and specific software developed. A discussion of the details of this software development has been addressed, too.

In essence, Automatic Data Acquisition systems research at Virginia Tech has now evolved to such a stage that with a little ‘tuning’ of hardware & the associated software, a very powerful and versatile automated data collection and management aiding tool will be available for economical widespread implementation.
Acknowledgements

The author would like to thank his committee chairman, Dr. Antoine G. Hobeika for his invaluable guidance and support during the course of this study. The author would also like to acknowledge the assistance and cooperation of Dr. Charles Nunnally in the production of this thesis. The author is extremely grateful to Dr. Siamak Ardekani and Dr. Bahram Jamei for providing support and encouragement throughout the course of study at Virginia Tech.

A warm and sincere thanks is also extended to the author’s friends and a host of others for their help.

Most importantly, the author is grateful to his parents, and , for their constant love and support. The encouragement and patience of his brothers, sisters-in-law and aunt are gratefully acknowledged.
Table of Contents

1.0 Introduction .............................................................................................................. 1
  1.1 Background ............................................................................................................ 1
  1.2 Description ............................................................................................................ 2
  1.3 Evolution of ADAS Research and Development at Virginia Tech: ..................... 3
    1.3.1 Standardization of the Software Package: .................................................. 4

2.0 LITERATURE REVIEW ........................................................................................... 6
  2.1 Background ............................................................................................................ 6
  2.2 ADAS General Information .................................................................................. 7
  2.3 ADAS Equipment: ................................................................................................ 10
    2.3.1 Passenger Sensing Equipment: .................................................................... 10
    2.3.2 Location Establishing Equipment [Signposts] ............................................. 11
    2.3.3 Data Gathering Equipment .......................................................................... 12
    2.3.4 Data Transferring Equipment ...................................................................... 12
    2.3.5 Data Processing Equipment ......................................................................... 13
  2.4 Accuracy of ADAS: ............................................................................................... 13
    2.4.1 Total Passenger Counts Summary ............................................................... 14
A.1 INTRODUCTION ................................................... 115
A.1.1 WHAT IS AN ADA SYSTEM? ...................................... 115
A.1.2 APPLICATION AND USAGE OF ADA SYSTEMS ................. 117
A.1.3 HOW TO USE THIS MANUAL ...................................... 118
A.2 FEATURES OF THE SYSTEM (HARDWARE): .............................. 120
A.2.1 HARDWARE COMPONENTS ....................................... 120
A.2.2 INTERFACING HARDWARE COMPONENTS. ........................... 121
A.2.3 OFFICE AUTOMATION SYSTEM COMPONENTS ..................... 121
A.2.4 LAY OUT OF THE COMPONENTS: .................................. 121
A.2.5 INSTALLATION AND TESTING OF THE SYSTEM: ....................... 123
A.2.6 MAINTENANCE, HANDLING AND CARE OF THE SYSTEM: .......... 125
A.2.7 PROBLEMS - PRECAUTIONARY MEASURES. .......................... 126
A.3 FEATURES OF THE SYSTEM (SOFTWARE): .............................. 126
A.3.1 OPERATIONS SOFTWARE [COMMUNICATIONS SOFTWARE]: ............. 126
A.3.2 MANAGEMENT SOFTWARE: ...................................... 128
A.3.3 APPLICATIONS: ................................................ 129
A.3.4 USAGE : HOW TO USE THE SOFTWARE: ............................. 132
A.3.5 USEFUL HINTS ON EFFICIENT STORAGE: ............................ 156
A.3.6 PROBLEMS : THEIR ALLEVIATION ............................... 158
A.4 FEATURES OF THE INTERFACING SYSTEM: .............................. 159
A.4.1 COMPONENTS USED: ........................................... 159
A.4.2 PROCESS OF 'DATA DUMPING': .................................. 160
A.4.3 PROCESS OF 'DATA TRANSFER': ................................. 160

Appendix B. EXTERNAL FILES ............................................ 161
B.1 SYSTINFO.EXT FILE ................................................ 161
B.2 SECT15.EXT FILE .................................................. 163

Table of Contents viii
List of Illustrations

Figure  1. Basic functions of an Automatic Data Acquisition System. ...................... 9
Figure  2. ADAS Implementation Strategy (from 6). ......................................... 20
Figure  3. Beams Lay Out at the Stairwells. .................................................. 27
Figure  4. Log Structure in Roanoke Valley Metro System ................................ 29
Figure  5. General Flow Diagram of the Data Processing Algorithm. ...................... 41
Figure  6. Flow Diagram of Data Integration Phase. ......................................... 43
Figure  7. Flow Diagram of Data Matching Phase. .......................................... 46
Figure  8. Flow Diagram of the Output File Storage Phase. .............................. 48
Figure  9. Passenger Load Plot. ............................................................... 80
Figure 10. Route Demand Plot. ................................................................. 81
Figure 11. Route Evaluation Plot. ............................................................... 83
Figure 12. Schedule Evaluation Plot. ......................................................... 85
Figure 13. Systems View of ADAS. ............................................................. 92
Figure 14. Lay Out of the Brown Sheet Route File (TRT, Norfolk). .................... 94
Figure 15. Sample Page of Brown Sheet Route File (TRT, Norfolk). .................... 95
Figure 16. Format of the UMTA TRS 80 Data File (TRT, Norfolk) ...................... 104
List of Tables

Table 1. Total Passenger Counts Summary (from 5) .................................... 15
Table 2. Stop by Stop Accuracy Comparison (from 5) ................................... 17
Table 3. Cost Comparison of ADAS (from 5) .................................................. 18
Table 4. Sample Test Data File Creation. ....................................................... 51
Table 5. Results from different test scenarios. ............................................... 53
Table 6. Variation of the Matching Process with DeltaT .................................. 56
Table 7. Variation of the Matching Process with DeltaL .................................. 57
Table 8. Variation of the matching process with DeltaS1 (DeltaT = 36) ............... 58
Table 9. Variation of the matching process with DeltaS1 (DeltaT = 40) ............... 59
Table 10. Variation of the matching process with DeltaS2 ................................ 61
Table 11. Sample Sheet of Bus Stop Listing. .................................................. 65
Table 12. Sample Sheet of Timepoint Profile Report. ...................................... 69
Table 13. Reporting Levels allowed for Transit Properties. ............................... 71
Table 14. Sample Sheet of Section 15 Daily Record File. ................................. 72
Table 15. Sample Sheet of Section 15 Annual Report File. ............................... 76
Table 16. Sample Sheet of Summary Report File. ........................................... 78
1.0 Introduction

1.1 Background

In recent years, transit systems across America have faced increasing pressures to use their resources more efficiently. The growing awareness to the federal government's increasing cutbacks in the operating subsidies it grants to mass transit has forced a shift in emphasis from capital intensive transit improvements to short range transit efficiency actions. The need to improve transit efficiency is more than ever, now.

In order to improve efficiency, transit managers will need to make intelligent and informed decisions. But such decisions require an extensive, accurate and continuous flow of data on ridership, fare revenue, schedule adherence and the like. Establishing this kind of a comprehensive and statistically valid database through labor intensive manual data collection method may be expensive and may offset the savings that would otherwise be incurred through wise decisions. It is to be noted that the largest cost item in the manual data collection budgets is the manpower needed to collect data on board buses and on the street [2]. Hence, transit properties are beginning to adopt automated data acquisition techniques.
The federal government is also promoting research investigating various aspects related to future widespread implementation of Automated Data Acquisition Systems (ADAS) [6]. Automated monitoring programs lend the capability of detecting detailed changes in running and loading characteristics as well as in various level of service standards [2]. Also, they lend the capability of automating the fulfilment of UMTA requirements. Recent research has considerably advanced the state-of-the-art of transit assessment methods. A number of large and small transit agencies have implemented ADAS [1]. They have now adopted sets of service performance measures and set up systematic evaluation programs. One such example is the Valley Metro at Roanoke, VA.

1.2 Description

An Automated Data Acquisition System is a composition of hardware and associated computer software which work together to yield certain basic reports, crucial in making intelligent and informed decisions concerning routes, schedules and operations. The data collected using the hardware gives information about the number of passengers boarding and alighting at various stops at various times. There are number of advantages to knowing the location and time of passenger activities rather than just the magnitude. For instance, such information may show that certain parts of a route have very little activity. Route revision in such a case may result in improved service at no additional cost. Similarly, the data might indicate a very sharp rise or increase in demand at certain times of the day. This will allow operations managers or planners to make allowances to account for these, like having larger buses, or perhaps reduced headways. Such trends would never become apparent without the time and location data which ADAS provide. Also, the data collected using the ADAS hardware can be used to evaluate the transit system's schedule adherence. The data might show that certain stops are consistently being reached well before or after the scheduled arrival time.
Adjusting the schedule would not only benefit the riders, but would also relieve the extra pressure which an unrealistic schedule places on the drivers. Finally, the data collected can be used to generate demand models for individual routes so that better operational planning for efficient management of resources is possible.

1.3 Evolution of ADAS Research and Development at Virginia Tech:

The Automatic Data Acquisition System (ADAS) or Automatic Passenger Counter System (APC) project has gone through several stages in it's development. In the first stage, the APC System was installed at Roanoke Valley Metro in Roanoke, Virginia. In this stage, certain software to decode, edit, manipulate and create a database from raw data was also addressed. In the next stage, some of the software development potential from the processed data file was tapped and certain management reports and plots which would be useful for decision making as well as to satisfy certain mandatory requirements from UMTA for obtaining operating subsidies were generated. This work was addressed by Avadhani [9]. In the next phase of the project, Anderson [3] addressed the matching process to make it more credible. He also worked on building some more management reports and plots. Here, the programs were written in Basica, and were made interactive. The design of the software was made with modularity in mind. Since then, the concept of a modular and structured system has been introduced. The author has continued the work from that point onwards to refine the matching algorithm through a better understanding of the process. Some more crucial reports and plots have also been developed. The user-friendliness of the package has been improved. The
programs have been documented and compiled into executable forms making the package quicker and more efficient.

1.3.1 Standardization of the Software Package:

The software package put together by Virginia Tech has been standardised with an option to create processed output file in the format desired by any transit agency. This can be identified by the fact that the Automatic Data Acquisition System developed by Virginia Tech has four levels namely

- Level 0 - Installation of the hardware and the establishment of communications to dump the raw data file on to a processing computer.

- Level 1 - Data Processing and Report and Plot Generation using APC software programs developed by Virginia Tech.

- Level 2 - Development of interfacing input / output software to convert existing reference files (if any) from the particular transit agency’s system to the format of Level 1 and also to convert output file to the format of Level 3 depending upon the transit property’s needs, if any.

- Level 3 - Providing the output file in the required format of the particular transit property, if any for further processing and report and plot generation.

This modular system design has now allowed the ADA system to be implemented in any transit agency with modifications to be made in Level 2 depending on Level 3. Also, it
allows for a comparison between the outputs of the tailor-made software of transit property, if any, with the software package put together as part of the ADA system. These are all some of the ideas put forth in this thesis. An implementation of this ‘System’s Approach’ has also been addressed [Ch. 7] where another system has been installed at Tidewater Rapid Transit at Norfolk, VA.

In summary, the following constitute the major work towards this thesis:

- Refinement, Debugging and Modification of the matching algorithm. This algorithm will eliminate the need for signposts.

- Testing and Validation of the algorithm by creating test data files simulating test scenarios.

- Justification of the algorithm by a sensitivity analysis on real life data file obtained from Roanoke Valley Metro at Roanoke, VA.

- Development of additional software for the management namely Summary Reports File, Route Evaluation Plot and Schedule Evaluation Plot.

- The documentation of processing programs and improvement of user-friendliness of the package.

- Conceptualization of the ‘System’s Approach’ to the development of software for new ADAS installations, and implementation of the same for an experimental project at TRT at Norfolk, VA. This involved development of interfacing software based on TRT’s needs and requirements.
2.0 LITERATURE REVIEW

2.1 Background

Transit is a collection and distribution process for people with common travel needs. Although passengers need not start at the same origin and end at the same destination, their travel patterns must concentrate them in sufficient numbers to justify specialized equipment and common routing.

Route design concerns itself with the assembly and dispersal of people so that they can be accumulated on a fixed route in quantities that will justify transit service and yet serve the many diverse trips that take place in an urban area. The ADAS equipment put together at Virginia Tech can be used for data collection only on fixed routes. Fixed routes are those, whose service is fixed in both time and space [4]. They generally exist in systems where high capacity and general service are required.

Collection of data on ridership and schedule adherence on these fixed routes is fundamental to transit system productivity: that is, optimum use of labor and capital expenditures, operational efficiency and the provision of effective bus service. The information derived from such data is used by transit managers for the following:
• Create, evaluate and adjust schedules and run times.
• Plan and justify route changes.
• Assign particular bus types (ex: articulated).
• Evaluate marketing strategies (ex: fare incentives).
• Estimate expected revenue.
• Determine fleet needs.

2.2 ADAS General Information

The general economic climate, funding cuts and reduced operating subsidies are now forcing transit properties to adjust service and fares and/or otherwise modify operating practices. To do these changes in an informed and systematic fashion so that system productivity is increased, without letting the service and operations to deteriorate, and to observe the effects of various service or fare changes on ridership and revenue, a thorough and reliable database is needed. To set up this kind of a database through manual data collection methods would be expensive. Hence automated data collection methods are in vogue, today.

Automatic Data Acquisition Systems are different from Automatic Vehicle Monitoring (AVM) systems in that ADAS do not provide real time vehicle monitoring and control. Automatic Data Acquisition Systems offer an opportunity to reduce the costs and improve the reliability and ultimate accuracy of ridership and service information. Also, the continuous flow of data will stimulate creative analysis and tuning of resources and services.

When ADAS is used, typically, data are stored on-board the vehicle, retrieved periodically via computer-readable devices, and processed by a computer into specific information reports, as needed. The basic steps showing the functioning of an ADAS are given in Figure 1.

Referring to Fig 1, the following are the basic steps
1. Data Acquisition.

2. Data Recording and Storage.

3. Data Transfer.

4. Reporting and Analysis.

Data Acquisition: This stage involves the following steps: The door activity is sensed by an ‘ON/OFF’ contact signal, passenger activity is sensed by sensing devices and distance is sensed by odometer pulses with the aid of a transmission pick up.

Data Recording and Storage: This stage involves the following steps: Recording the door activity, Recording Passenger Ons and Offs, Recording Distances and Recording Time. The Passenger Ons and Offs are recorded on the passenger Count Modules, Distances are recorded on the Odometer power module and all these together with the time of their occurrence are stamped on the Data Storage Module in the order of their creation or occurrence.

Data Transfer: This stage involves the following steps namely: Transfer to storage media and then transferring data to CPU. The first step involves a communication software to transfer data from the DSM to the secondary storage device.

Data Processing, Reporting and Analysis: This stage is the key stage and involves several steps. First of all, the reference files are created. The raw data file is retrieved, activity records are created, good loops are located and matched and then the output file is stored. Next, depending on the needs, the reports and plots are generated.
Figure 1. Basic functions of an Automatic Data Acquisition System.
2.3 ADAS Equipment :

Though all ADA systems have a similar function, they may be very different in the equipment they use to accomplish their tasks. One problem is the lack of industry standards. Every company supplying the equipment seems to use a different approach to solve the same problems. Reliability will improve as more systems are put to work and more experience is gained. Following are the major components of ADA systems:

1. Passenger Sensing Equipment.
2. Location Establishing Equipment.
3. Data Gathering Equipment.
4. Data Transferring Equipment.
5. Data Processing Equipment.

2.3.1 Passenger Sensing Equipment :

These consist of mainly the counters or sensors. There are three main types of sensors :

1. Infrared Beam Sensors : This type of sensor transmits two or more beams of infrared light across the doorway of the bus. When one or more of the beams is broken the signal is sent to a logic unit in the on board computer. Depending upon the order in which the beams are restored, the signal is registered as an ‘off’ or an ‘on’. Generally, the more beams there are, the more reliable the count [5]. This is the type of sensor used at Valley Metro as well as at TRT, Norfolk.
2. Reflective Infrared Beam Sensors: These sensors are very similar to the regular infrared sensors except that they use transmitters that also act as receivers of the beam. Thus, only one side of the doorway has a sensor unit; the other side has a reflector mounted opposite the light source.

3. Treadle Sensor Mats: Another way to sense the presence of a passenger is through the use of pressure sensitive mats. These mats are placed on the stairways at the front and rear of the bus and send a signal to the onboard computer when a force exceeding some predetermined threshold amount is applied. Depending on the sequence in which the signal is triggered the logic unit records the activity as either an on or an off. A fourth type of sensor is being developed, but has not been used in actual service. This sensor would be similar to infrared sensors but would use ultrasonic sound waves rather than beams of light.

2.3.2 Location Establishing Equipment [Signposts]

Signposts are optional equipment employed by many APC systems to increase the accuracy of the location determination. Signposts are transmitters located at specified points along a fixed route which periodically emit a coded signal. When a properly equipped bus enters the range of the signpost, it receives the signal and records the time and location at which the event occurred. Later, this information provides a very convenient reference point for determining the location of a particular event (viz passenger boarding).

On systems without signposts, data location referencing is performed by special software which compares the events with detailed schedule and/or distance reference files. This method eliminates the problems and costs associated with the installation and maintenance of radio signposts.
2.3.3 Data Gathering Equipment

These consist of the following essential modules -

1. Odometer/Power Module (OPM).
2. Passenger Count Modules (PCM).
3. Data Storage Module (DSM).
4. Diagnostic Module (DGM).

All these modules are basically microprocessors engineered by Red Pine. The Data Storage Module is the ‘System Controller’. It is the whole ‘brain’ of the system. The DSM is a small microprocessor that accepts, monitors and controls the data collection and data transfer functions of all other modules. It also includes a clock. It accepts data directly from the other modules as well as the signal from the driver door control switch.

The Data Storage Module at Valley Metro, Roanoke is a Motorola 6803P Microprocessor which is capable of interpreting, recording and transferring the data to a secondary device. It has an electronic timer incorporated in it so that the time of occurrence of various events can be recorded. A logic algorithm helps it convert this information into number of boardings and alightings (ONs and OFFs).

2.3.4 Data Transferring Equipment

The Data Transferring Equipment include the secondary storage device and communication links. The secondary storage device used may be a lap held computer. In the case of Valley Metro, it is the DG/ONE. At TRT, Norfolk, an IBM lap held computer is being planned to be used.
2.3.5 Data Processing Equipment

The equipment used for data processing may vary from personal computers to large mainframe systems. For example, in Roanoke Valley Metro, an IBM PC XT is being used for data processing. At TRT, Norfolk, it is planned that an IBM lap top computer be used for data processing and later on the management reports and plots would be generated using Datapoint mainframe computers. The choice is generally a function of the amount of system size and level of reporting desired. Faster, more powerful units are required to handle large amounts of data or to produce very detailed reports. Some systems actually have contracts with private firms which handle all of the data processing and report generation [23]. Such an arrangement may be very convenient; however, it tends to increase data turnaround time and costs to the transit company.

2.4 Accuracy of ADAS:

Any discussion of automatic data acquisition systems eventually leads to the issue of accuracy. How good are the counts, and how do they compare to the counts obtained by manual ride checkers? A major study sponsored by the Urban Mass Transit Administration (UMTA) Office of Bus and Paratransit Systems was conducted to determine just that. The study was conducted by the Transportation Systems Center of Cambridge, Mass. and involved the following transit agencies: Minneapolis/St.Paul, (MTC); Columbus, (COTA); Seattle, (Metro); and Kalamazoo, (METRO TRANSIT) [6]. The study focussed on the accuracy of ADAS systems compared to manual systems and does not compare one ADAS system with another. The statistics given are of two types: Total passenger counts and stop by stop analysis.
2.4.1 Total Passenger Counts Summary

This is a measure of the difference in the total counted ons or offs by the ADAS and by ride checkers compared to a "truth team". The truth team was made up of two counters, each of whom was responsible for recording all passenger activities at one door only. (Normally, one ride checker is responsible for both doors, and at peak periods, he or she may have trouble keeping track of all activities). The two "truth" team members were not to compare notes in order to maintain the integrity of the test. The results of the test are presented in Table 1 from [6].

The results show that the "truth" standard was unreliable, and that the manual checkers were slightly more reliable than the ADAS's (2.6% error vs 5.8%). However, since the ride checkers knew they were being monitored, their performance was probably better than usual. This statistic is somewhat useful if the system is being used mainly to determine the gross number of passenger activities. But since it aggregates the counts and the errors can cancel each other out, it says very little about the actual accuracy of the system. The second set of statistics does a much better job.

2.4.2 Stop-by-Stop Analysis

This test tracked both the magnitude and frequency of errors at any given stop. The results indicate that the accuracy of the passenger counts, both manual and ADAS, are inversely proportional to the number of passengers boarding or alighting. This was also the experience at Valley Metro. The reason for this is that for manual checkers, the view is often blocked by other passengers, and also that it is impossible for one person to constantly monitor both doors simultaneously. With ADAS's, on the other hand, passengers tend to stand in the beams, or a passenger boarding blocks the beams so that they cannot detect another
Table 1. Total Passenger Counts Summary

<table>
<thead>
<tr>
<th>Day</th>
<th>Coach</th>
<th>Start Load</th>
<th>End Load</th>
<th>Expect Diff. (Off-On)</th>
<th>&quot;Truth&quot;</th>
<th>RIDE-CHECKER</th>
<th>APCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TEAM A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>A</td>
<td>20</td>
<td>2</td>
<td>18</td>
<td>380</td>
<td>392</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>22</td>
<td>1</td>
<td>21</td>
<td>434</td>
<td>445</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>32</td>
<td>0</td>
<td>32</td>
<td>378</td>
<td>426</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>17</td>
<td>0</td>
<td>17</td>
<td>423</td>
<td>441</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>A</td>
<td>31</td>
<td>3</td>
<td>28</td>
<td>400</td>
<td>426</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>16</td>
<td>0</td>
<td>16</td>
<td>383</td>
<td>492</td>
<td>109</td>
</tr>
<tr>
<td>TOTAL</td>
<td>A</td>
<td>83</td>
<td>5</td>
<td>78</td>
<td>1156</td>
<td>1244</td>
<td>86</td>
</tr>
<tr>
<td>TOTAL</td>
<td>B</td>
<td>55</td>
<td>1</td>
<td>54</td>
<td>1240</td>
<td>1378</td>
<td>138</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td>138</td>
<td>6</td>
<td>132</td>
<td>2398</td>
<td>2622</td>
<td>224</td>
</tr>
</tbody>
</table>

% Difference between On and Off

8.5%  2.6%  5.8%
passenger alighting. Therefore, the best way to evaluate the accuracy is by grouping it according to the number of passengers per activity. Table 2 summarizes the study results [6].

Again, it shows that manual checkers are slightly more accurate, but their results are probably biased on the high side for the reason stated earlier. The conclusion is that automatic passenger counters are very competitive with manual collection methods as far as accuracy is concerned. In addition, once the system is in place, it can continuously collect data for days at a time.

2.5 Costs of ADAS Systems

As mentioned earlier, one of the main justifications for ADAS systems is to save money. If the data collection program is not cost effective, there is no great advantage in having the system. In light of this fact, cost comparisons have been made between manual and automated collection methods. They reveal that ADAS systems have a higher initial cost due to the expense of equipment and installation. However, the annual costs for the installed system tend to be significantly lower than manual methods, and the payback period is usually less than five years. The cost per additional unit is much less for ADAS system, since most of that cost is software development and data processing. A comparison of the marginal cost of adding a full time checker (about $23,000) to that of an additional ADAS unit (about $6,000 per equipped bus) clearly illustrates this fact [5]. The larger the bus system, the larger the potential for savings with an ADAS system, as illustrated in Table 3 [5].

Note, however that on small systems (i.e. less than 50 peak hour buses) the projected costs for ADAS systems are higher than those for manual data collection. One major objective of this research is to determine the feasibility of ADAS systems for such transit companies.
Table 2. Stop by Stop Accuracy Comparison.

<table>
<thead>
<tr>
<th>Number of Passengers Boarding/Alighting</th>
<th>Number of Observations</th>
<th>% of Time No Count Errors</th>
<th>% of Time Count was Within ±1</th>
<th>% of Time Count was Within ±2</th>
<th>% of Time Count Error was Equal to or Greater Than 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>APCS</td>
<td>R/C</td>
<td>APCS</td>
<td>R/C</td>
</tr>
<tr>
<td>1</td>
<td>4100</td>
<td>87.4</td>
<td>91.8</td>
<td>99.4</td>
<td>99.4</td>
</tr>
<tr>
<td>2</td>
<td>1900</td>
<td>80.8</td>
<td>88.3</td>
<td>97.0</td>
<td>95.0</td>
</tr>
<tr>
<td>3</td>
<td>970</td>
<td>73.8</td>
<td>83.4</td>
<td>94.9</td>
<td>95.3</td>
</tr>
<tr>
<td>4</td>
<td>515</td>
<td>62.7</td>
<td>74.9</td>
<td>90.5</td>
<td>93.6</td>
</tr>
<tr>
<td>5</td>
<td>285</td>
<td>61.4</td>
<td>75.1</td>
<td>87.7</td>
<td>90.2</td>
</tr>
<tr>
<td>6</td>
<td>200</td>
<td>60.6</td>
<td>82.2</td>
<td>97.5</td>
<td>96.1</td>
</tr>
<tr>
<td>7</td>
<td>140</td>
<td>48.6</td>
<td>70.6</td>
<td>80.3</td>
<td>89.5</td>
</tr>
<tr>
<td>8</td>
<td>120</td>
<td>41.2</td>
<td>68.0</td>
<td>75.6</td>
<td>90.1</td>
</tr>
<tr>
<td>9</td>
<td>80</td>
<td>42.7</td>
<td>66.7</td>
<td>75.6</td>
<td>89.3</td>
</tr>
<tr>
<td>10</td>
<td>70</td>
<td>45.7</td>
<td>57.7</td>
<td>72.8</td>
<td>78.8</td>
</tr>
<tr>
<td>11</td>
<td>45</td>
<td>53.2</td>
<td>73.5</td>
<td>78.7</td>
<td>89.8</td>
</tr>
<tr>
<td>12</td>
<td>32</td>
<td>53.1</td>
<td>65.6</td>
<td>84.3</td>
<td>78.1</td>
</tr>
</tbody>
</table>

*Note that ride checker performance is probably biased high, since the ride checkers were aware that their results were being compared to two other count sources.

Table 4

Comparison of APCS and Ride-Checker Accuracy (Composite Data)

LITERATURE REVIEW
### Table 3. Cost Comparison of ADAS

<table>
<thead>
<tr>
<th>Peak Buses</th>
<th>Number of Traffic Checkers Required*</th>
<th>Number of APC Equipped Buses*</th>
<th>Annual Costs**</th>
<th>Annual Costs per Unit**</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Manual Program</td>
<td>APC Program</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Manual Program</td>
<td>APC Program</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>1</td>
<td>59,000</td>
<td>82,000</td>
</tr>
<tr>
<td>50</td>
<td>2</td>
<td>2</td>
<td>86,000</td>
<td>84,000</td>
</tr>
<tr>
<td>100</td>
<td>4</td>
<td>4</td>
<td>142,000</td>
<td>121,000</td>
</tr>
<tr>
<td>200</td>
<td>6</td>
<td>6</td>
<td>201,000</td>
<td>142,000</td>
</tr>
<tr>
<td>300</td>
<td>7</td>
<td>7</td>
<td>227,000</td>
<td>149,000</td>
</tr>
<tr>
<td>500</td>
<td>13</td>
<td>13</td>
<td>385,000</td>
<td>196,000</td>
</tr>
<tr>
<td>750</td>
<td>15</td>
<td>15</td>
<td>436,000</td>
<td>223,000</td>
</tr>
<tr>
<td>1000</td>
<td>19</td>
<td>19</td>
<td>532,000</td>
<td>245,000</td>
</tr>
<tr>
<td>2000</td>
<td>38</td>
<td>38</td>
<td>$1,027,000</td>
<td>$396,000</td>
</tr>
</tbody>
</table>

*Assumes the maximum number of units (checkers or APC buses) required as stated in the Bus Transit Monitoring Study.

**Assumes the costs accrued over a five-year period (discount rates were not applied to annualized costs).
2.6 ADAS Implementation Strategy

It is difficult to find a strategy for ADAS introduction that is realistic to implement on a system wide basis. Experience has led transit properties to believe that an Incremental approach for implementation should be adopted. A practical process for guiding the introduction of ADAS into a transit agency consists of three phases:

1. Planning and Strategy.
2. Initial Implementation.

Each of the above phases consists of several steps. These are shown in detail in the Figure 2.
Figure 2. ADAS Implementation Strategy (from 6).
3.0 AUTOMATIC DATA ACQUISITION SYSTEM

HARDWARE

3.1 Introduction

The material in this chapter deals exclusively with the various system components involved in the synthesis of an Automatic Data Acquisition System. The various modules supplied by M/S Red Pine Associates for the ADAS installed at Tidewater Rapid Transit at Norfolk, VA. were tested and analyzed by Misra of the Electrical Engineering Department at Virginia Tech under the guidance of Dr. Charles Nunnally [25] and valuable information about the system was documented. It is worth noting here that, the system modules of the ADAS installed at Roanoke Valley Metro at Roanoke, VA. were similar to the ones at Norfolk System.
3.2 System Description

The Automated Data Acquisition System consists of four modules interconnected to each other in a daisy-chained fashion, and four sensor pairs, each containing two pairs of infra-red transmitters and receivers.

The modules are:
1. Odometer / Power Module (OPM).
2. Passenger Count Module (PCM).
3. Data Storage Module (DSM).
4. Diagnostic Module (DGM).

The ADASs at Roanoke as well as at Norfolk comprises of four modules namely one OPM, one DSM and two PCMs, one each for front and rear doors. The Diagnostic Module can be connected to the system whenever certain operational and functional parameters of the system need to be examined. The functions of each of the modules are described below.

3.2.1 Odometer / Power Module: (OPM)

The odometer / power module is fed directly from a 12 V battery and the rest of the modules in the ADAS get their power from this module via interconnecting wires., i.e. the OPM serves as a power distribution center for the system. It also accepts pulses from the odometer as it’s inputs with the aid of a small transmission pick-up installed in the transmission of the bus. The incoming pulse train is divided and then counted to get a measure of the distance travelled by the bus. The division factor can be chosen to be between 1 and 4095, by using jumpers. In the Roanoke Valley Metro System, the division factor was set at approximately eight (actually 1 odometer click = 8.58 feet). The division factor set for operation at TRT,
Norfolk, was four (1 odometer click = 5.13 feet). The distance travelled by the bus is passed on to the other modules so that appropriate logs can be generated.

3.2.2 Passenger Count Modules

Every ADAS has two PCMs, one for the front door and another for the rear door monitoring. These are connected with the remaining two modules in a daisy-chained fashion. The PCM in conjunction with two pairs of infra-red beam sensors and a door open indicator, counts boarding and alighting passengers in a single stairwell. Each module monitors the status of doors via an ON/OFF contact signal and passenger activity via infra-red sensor pairs mounted on the inner walls of the passage of each door. Each sensor pair consists of two pairs of emitters and detectors generating four infra-red beams. The order in which these beams are broken and restored, due to passenger movements, determines the passenger activity as a boarding or alighting from that door. The vehicle movement information and the door open/close signal are used by these modules to generate appropriate logs. The PCM is an autonomous module with serial data communications facility. A set of such modules constitute an ADAS.

3.2.3 Data Storage Module

The DSM is an autonomous module whose primary function is to receive and retain the logs generated by the remaining modules and store them in the order of their arrival. The order of the arrival of the logs is the same as the order of their generation or creation. The other modules in the system use special flags to signal a particular type of event to the DSM so that it can initiate requisite data transfers from that module to prepare a log. This data stored in the module can be transferred either manually via a serial cable connection to a

AUTOMATIC DATA ACQUISITION SYSTEM HARDWARE
portable computer or automatically via a medium range (upto 8 feet) infra-red optical link to a stationary wayside receiver. The data transfer rates of the manual link and the automatic infra-red data link are 4800 bits per second and 64000 bits per second respectively. The ADA systems at both Roanoke and Norfolk, adopt a manual method of data transfer. A set of such modules are grouped together to form an ADAS. The data is stored on the DSM in the form of a set of records called ‘logs’ which are 5 bytes long. A detailed description of logs is given in the section 3.6.

3.2.4 Diagnostic Module

The diagnostic module provides diagnostic facilities for all the other modules in the ADAS. The DGM is an autonomous module with a serial data communications facility. The module is portable and is plugged into the ADAS only when diagnostics are required. The Diagnostic Module in the system at Roanoke is different from the one at Norfolk. The system at Roanoke has twelve logs overall, representing various events while the one at Norfolk has fourteen logs. The module at Roanoke has only one level of operation and provides diagnostics of the other modules within that level. The Diagnostic Module at Norfolk has a 16-key keypad, 80 character alphanumeric LCD Display. The various diagnostic functions are organized hierarchically in a tree structure form. This is described in the Chapter 7 titled “ADAS Project at TRT, Norfolk”.

3.3 Sensors

The ADAS has four pairs of infra-red sensor units. Each unit consists of two pairs of transmitters and receivers. Two pairs of sensor units are mounted on the inner sides of each
door-well of both the front and rear doors. These sensor units are connected to Passenger Count Modules and get the power from them. When powered, each transmitter produces a beam which is received by a corresponding receiver. Four beams are produced at each door. The states of the beams (broken or unbroken) in conjunction with the order of their making and breaking determine the logic for the count generation algorithm representing passenger activity. A detailed discussion on the count generation algorithm is given in the section 3.5.

3.4 Communications

Manual method of communication is provided for the ADAS at Norfolk as well as at Roanoke. These systems employ two different types of communication methods, one for intermodule communication and another for dumping (downloading) the data file to a computer. Both use a packet based approach for requests, acknowledgements and actual data transfers. The communications with the outside world is by means of an RS232 interface without any handshakes [12].

3.5 Count Generation Algorithm

The infrared sensor units at each door generate four beams when in a normal powered state. When an opaque object crosses the door, the beams are broken. The sequence of making and breaking of these beams determines whether a passenger is climbing aboard or alighting. The system, however, demands that all four beams be broken...
and made before any activity is registered. The on and off count generation occur when the following are realized:

    Referring to the Figure 3,

If all the beams are broken and then made in the order 1,2,3 & 4, then an ‘ON’ count is generated. If, however, all the beams are broken and then made in the order 4,3,2 & 1, then an ‘OFF’ count is generated.

    The sequences determining the count generation can be summarized as follows -
    1. On Count Generation

    • B1,B2,B3,B4 and U1,U2,U3,U4.

    • B1 followed by breaking of remaining three and then all are made or unbroken.

    2. Off Count Generation

    • B4,B3,B2,B1 and U4,U3,U2,U1.

    • B4 followed by breaking of remaining three and then all are made or unbroken.

Here, B1,B2,B3 and B4 represent broken states of the beams 1,2,3 and 4 respectively and U1,U2,U3 and U4 represent unbroken states of the beams 1,2,3 and 4 respectively.

### 3.6 Description of Logs and Log Structure

This section deals exclusively with what logs are, when they are created and their structure. The two systems at Roanoke and Norfolk are different in their log structure. A
Figure 3. Beams Lay Out at the Stairwells.
description of logs and their structure in both the systems are addressed in the following sections.

First of all, logs are records of fixed length which are created whenever a particular activity occurs. A set of these log records form a data file. Logs are created for the following reasons:

- The bus is idle for more than one minute.
- The counter sensors detect a passenger activity.
- The destination signboard is changed.
- The time or distance registers overflow and is reset.

These logs or records are five bytes long. The five bytes are the log type, the time elapsed in fifteen second units, the distance travelled in odometer impulses, passenger ons and passenger offs respectively.

3.6.1 Log Structure in Roanoke Valley Metro System

The log structure is shown in the following figure:

The functions of each of the logs are:
<table>
<thead>
<tr>
<th>REASON</th>
<th>TYPE</th>
<th>TIME</th>
<th>DIST</th>
<th>ONS</th>
<th>OFFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>POWER ON</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HOUR OVERFLOW</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ONE MIN. IDLE</td>
<td>3</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DIST OVERFLOW</td>
<td>4</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PASSENGER LOG</td>
<td>5</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>END LONG IDLE</td>
<td>6</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ELAPSED TIME</td>
<td>7</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>TOTAL DISTANCE</td>
<td>8</td>
<td>0</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
</tr>
<tr>
<td>SIGN BOARD LOG</td>
<td>11</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HEADER RECORD</td>
<td>12</td>
<td>MONTH</td>
<td>DAY</td>
<td>YEAR</td>
<td>ROUTE</td>
</tr>
<tr>
<td>HEADER RECORD</td>
<td>12</td>
<td>DAY OF WK</td>
<td>BLOCK</td>
<td>HOUR</td>
<td>MINUTE</td>
</tr>
<tr>
<td>HEADER RECORD</td>
<td>12</td>
<td>BUS ID</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 4. Log Structure in Roanoke Valley Metro System
<table>
<thead>
<tr>
<th>Log Type 1</th>
<th>Power up log - it is the very first log created after power on.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log Type 2</td>
<td>Hour Overflow log. It is generated when an hour expires. Since time is counted in 15 second units, one byte can store time clicks that make one hour and it will overflow beyond that.</td>
</tr>
<tr>
<td>Log Type 3</td>
<td>One minute idle log - generated when vehicle remains stationary for a period of a minute.</td>
</tr>
<tr>
<td>Log Type 4</td>
<td>Distance Overflow log - generated when distance, counted as odometer transitions, exceeds the value that can be held in a byte.</td>
</tr>
<tr>
<td>Log Type 5</td>
<td>Passenger log - generated after vehicle moves and if there was passenger activity at the stop.</td>
</tr>
<tr>
<td>Log Type 6</td>
<td>End of two or more minute idle log - generated when vehicle moves after being stationary for more than two minutes.</td>
</tr>
<tr>
<td>Log Type 7</td>
<td>Dump Forced Elapsed time log - generated when a dump is initiated and contains 24 bit value representing the time expired since last dump or power up.</td>
</tr>
<tr>
<td>Log Type 8</td>
<td>Dump Forced Total Distance - generated when a dump is initiated and contains 24 bit value representing total odometer transition count since last dump or power on.</td>
</tr>
<tr>
<td>Log Type 11</td>
<td>External Input (Signpost) log - generated whenever a signboard is changed by the action of the operator.</td>
</tr>
<tr>
<td>Log Type 12</td>
<td>Manually Input logs - Information given are Month, Day, Year, Route, Day of Week, Block, Hours, Minutes and Bus ID.</td>
</tr>
</tbody>
</table>
3.6.2 Log Structure in TRT, Norfolk System

The log structure is similar to the one used in Roanoke except that log 9, 10, 13 and 14 are also created. Also, there is a small variation in the function of log 12. The functions of all these logs are given as follows:

- **Log Type 9**: Front Passenger Count Module not responding log.
- **Log Type 10**: Rear Passenger Count Module not responding log.
- **Log Type 12**: Distance Overflow Overrun log.
- **Log Type 13**: Odometer Power Module not responding.
- **Log Type 14**: Vehicle Identification log - containing 8 digit vehicle ID.
4.0 AUTOMATIC DATA ACQUISITION SYSTEM

SOFTWARE

4.1 Overview

Volumes of data collected using the ADAS hardware are useless until they can be converted from this raw state to a processed output state from which meaningful reports and plots can be generated. This conversion of raw data into a processed form is called 'Data Processing' and this is done using ADAS software. This task is a challenging and a highly demanding one and at many transit properties, software development exceeds the cost of the hardware. In this chapter, the various software available on the ADAS to serve as decision aids are discussed.

All the programs constituting the ADAS software have been written in BASICA, an advanced form of Basic. They have all been compiled using Microsoft’s Quickbasic compiler into executable forms so that they are all efficient and have a short execution time. The programs are all interactive and easy to use. A user’s manual discussing the usage as well as handling of the software has been developed and is attached in the Appendix of the thesis.
Also, the software has a robust algorithm which does the matching accurately, removing the need for a radio signpost signal to be imbedded to establish locational accuracy. This consequently eliminates the signposts’ installation and maintenance worries and costs.

The ADAS software can be divided into the following two categories -

1. Operations Software.
2. Management Software.

### 4.2 Operations Software

This software is used for communications. The program ‘Datadump’ used for Roanoke Valley Metro, was written in Basic and compiled. This program establishes communications between the DSM on the bus and an external storage device like a lap held computer (DG/One in the case of Roanoke) so that the raw data file created and stored on the ADAS hardware can be downloaded to the lap held computer. It should be remembered, here, that the ADAS hardware remains ‘ON’ always and is generally reset at the beginning of the data collection period. The details on the usage of the communications software to establish the correct protocol and set up the ‘handshake’ for downloading the raw data file on the DSM to the lap held computer, are given in the user’s manual in the Appendix.
4.3 Management Software

The following are the management software available on the system -

1. APCDATA: This is used for data processing. The raw data file has to be processed before the reports and plots can be generated.

2. MAKEFILE: This is used for creating or editing reference files to be used with other programs for processing of the raw data file and later generation of reports and plots.

3. REPORTS: After the raw data file has been processed, this software is used to generate certain mandatory reports as well as certain decision aiding reports.

4. PLOTS: After the raw data file has been processed, this software is used for certain plots. These plots aid the planners and managers make better decisions regarding routing, scheduling and operations of routes.

4.3.1 APCDATA:

This is the main software program involved in the data processing stage. This program is completely interactive or userfriendly and offers the following options:

2. Generate Bus Stop Listing.
3. Quit (Exit to DOS).

This program has to be run using the option 1, before the reports or plots are generated using Reports or Plots program.
In order to identify the stops at which the various activities occurred, it is necessary to reference the activities to stops using files which contain the names of all the stops on the routes, the distances between them and the times at all timepoints. In order to create or edit the reference files mentioned above, the 'MAKEFILE' program is used.

The MAKEFILE program offers the following options:

1. Create or Edit Stop Distance File.
2. Create or Edit Stop Names File.
3. Create or Edit Schedule (Block) Reference File.
4. Quit (Exit to DOS).

This program is used, after the raw data file has been processed, for generating certain very useful reports. The following are the options available on this software -

1. Generate Time Point Profile Report.
2. Create Section15 Daily Report File.
5. Quit (Exit to DOS).

The fourth option namely 'Generate Summary Report File' was written and added to the existing package by the author to make the reports system more complete and versatile.
The Section 15 Report has to be sent by every transit agency to UMTA to continue to receive the operating subsidies. This is a condition laid down by the federal government - stated as follows:

"Under Section 15 of the Urban Mass Transportation Act of 1964, as amended, the Secretary of Transportation may not make any grant under Section 5 or 9 of the act unless the applicant for the grant and any beneficiaries are each subject to a reporting system and a uniform system of accounts and records as prescribed by the Secretary[6]."

4.3.4 PLOTS

From the processed output files, there are several plots that can be generated. These provide excellent graphical information support to the management for making decisions. Graphical presentation of the system performance through these plots reflect the trend in ridership as well as in schedule adherence. These are always better realized through graphical support than in the form of reports. Hence, this program offers four plots. The options available are:

1. Generate Passenger Load Plot.
2. Generate Route Demand Plot.
3. Generate Route Evaluation Plot.
5. Quit (Exit to DOS).
4.4 **External Files and Reference Files**

In order to reference the data to the individual stops, several external reference files are required. These are:

1. Stop Names Reference File.
2. Stop Distances Reference File.
3. Schedule (Block) Reference File.

### 4.4.1 Stop Names Reference File

Every route on which the ADAS is used for data collection will have a Stop Names Reference File. This file will contain the names of all the stops on the fixed route. This file is created using ‘MAKEFILE’ program. If an existing file has to be changed or edited, ‘MAKEFILE’ program can be used, again. An example Stop Names File is attached in the Appendix.

### 4.4.2 Stop Distance Reference File

There should be a stop distance file for every route. This file contains the distances between stops for the complete (fixed) route. However, the distances are expressed in number of odometer clicks. At Roanoke Valley Metro system, 1 odometer click = 8.58 feet, while at TRT Norfolk system, 1 odometer click = 5.13 feet. This file is also created using the ‘MAKEFILE’ program. An example Stop Distance File is attached in the Appendix.
4.4.3 Schedule (Block) Reference File

There should be a schedule reference file for every fixed route. This file contains information about maximum number of loops of operation per day, number of timepoints on the route and the times at each of the timepoints for the various loops for the day expressed in 15 second time units. This file is also created using the ‘MAKEFILE’ program. An example schedule reference file is attached in the Appendix.

4.4.4 System Files or Control Files.

There are several external files which contain control or system information. These files contain system parameters which can be changed depending on the needs of the transit company. The external files involved are -

4. Section.15A.

The variables contained in these files are explained in the Appendix.
5.0 DATA PROCESSING ALGORITHM

5.1 APCDATA

Data processing is done using this program. APCDATA is an interactive program written in BASICA. This program converts the raw data into activity records, matches the activities to stops and stores the results in the form of an output file. The output serves as the input for all the reports, plots and listings.

5.1.1 The Algorithm:

The data file generated during the collection period is dumped to the main computer for processing and eventual report generation. The data processing step is a key stage which determines the utility of the ADA system both in terms of present and future software needs of the transit property. The algorithm for executing this step is built modularly. It employs a simple, systematic and logical sequence of tasks to do the processing.

Three phases are involved in this software namely:
1. Data Integration Phase.
2. Data Matching Phase.
3. Output File Storage Phase.

This is shown in the figure 5 in the next page.

5.1.1.1 Data Integration Phase

The data file created by the ADAS hardware is a set of records of five bytes length. The five bytes are the log type, the time elapsed in fifteen second units, the distance travelled in odometer impulses, passenger ons and passenger offs respectively. Whenever a particular activity occurs, a particular log is created and this is stamped on the PCMs and stored on the DSM. Logs are created for the following reasons:

- The bus is idle for more than one minute.
- The counter sensors detect a passenger boarding or alighting (passenger activity).
- The destination signboard is changed.
- The time or distance registers overflow and is reset.

For example, when passenger activity like passengers getting on and/or off the bus occurs, a log type '5' is created and stored. Similarly, a log type '3' is created when the bus idles for a minute and a log type '6' is created when it idles for two or more minutes.

The Data Integration Phase involves the following simple tasks.

First of all, the external file containing all the system information is retrieved. This information includes the name of the transit company, the basenames of the reference files namely...
Retrieve External File.

Present Program Menu.

Retrieve Raw Data File.

Create Activity Records.

Retrieve Reference Files. Locate Activity Record near Stop 1 for loop 'L'.

Locate end of potential loop and calculate Matchratios.

Compare Matchratio(L) for all potential loops and select loop with best Matchratio.

Match all Activity Records within loops. Assign Matchvalue and stop number to Activity Records for all loops.

Backmatch and Frontmatch the remaining stray data and assign stop numbers to activities.

Ascertain the required output file format. Create the file format and store the same.

Figure 5. General Flow Diagram of the Data Processing Algorithm.
schedule, distance and names used for matching, the basename of the Section15 cumulative file, the disk drives where all of the above files are stored, the allowable distance variation in number of odometer impulses for determining when the matching routine should end (called \( \Delta L \)), the allowable time variance in number of 15 second time units for determining the end of the loop (called \( \Delta T \)), the outer and inner limits of a positive match based on distance in odometer impulses called \( \Delta S_1 \) and \( \Delta S_2 \) respectively, the number of feet per odometer impulse and so on.

The reference files used for matching are the distance, schedule and names files. The distance reference file includes information about distances between stops within a loop, number of timepoints along the route per loop, the route number and route name. The timepoints are the stops or checkpoints along the route where the bus operating on the loop has to be, at specific times. The schedule reference file includes information like the block number, number of loops of operation, number of timepoints along the route per loop, the starting time of the first loop, the ending time of the last loop and the various times at which the bus is scheduled to be at various timepoints. The names reference file includes information like the route number, route name, the number of stops per loop and the names of all the stops along the route.

Next, a program menu is presented and on choice, the raw data file to be processed is retrieved. Next, activity records are created. An activity record is any record of the log types 3, 5, 6, 11. The starting time of the operation of the route is calculated by tracking back from the time of data dumping. Then, the times and distances at which the various activity records occurred are calculated by a process of integration. This is done by adding all the time clicks till the particular record from the starting time. The same is done for the distance calculation. Finally, a count of the number of activity records is generated. This phase is represented by the flow diagram shown in the figure 6 in the next page.
Retrieve External File containing system parameters. Ascertain the basenames and drives in which the raw data file and reference files are stored.

Present Program Menu offering the following options: Process Raw Data, Generate Bus Stop Listing and Quit or Exit. On choice of 'Process Raw Data' option:

Retrieve Raw Data File to be processed from the drive specified above.

Create Activity Records by accumulating statistics for the data records. Track back from the time of data dumping to ascertain the starting time of operation of the route. Accumulate the time clicks and distance clicks to give the absolute time and distance for each of the activity records.

Go to Data Matching Phase.

Figure 6. Flow Diagram of Data Integration Phase.
5.1.1.2 Data Matching Phase

This is the key phase in the processing algorithm. In this phase, the following are executed: First of all, the reference files namely the Schedule and the Distance files are retrieved. Then the good loops are fitted. A good loop is one which passes a two fold test. The first one involves matching an activity to 'stop number one'. The first activity record which falls within 'DeltaT' of the scheduled starting time of a loop is taken as the temporary beginning of a potential loop. The second test involves locating the activity record occurring at the end of the loop. This is done by comparing the actual distance corresponding to the various activity records with the reference loop distance. If this difference is within DeltaL, then that activity record is taken as the temporary end of the potential loop. The match ratio for this loop is next calculated using the relation -

\[
MR = \frac{M}{(Te(L) - Tb(L))}
\]

where:

- MR = Match Ratio.
- Te(L) = Temporary end of the loop number L.
- Tb(L) = Temporary beginning of the loop L.
- M = The number of activity records that have their distances within DeltaS1 of the corresponding reference distance.

The matchratio reflects the percentage of activities which produced a match out of all the activities. This matchratio is stored. The next activity record which falls within DeltaT of the scheduled time for 'Stop number 1' is taken as the temporary beginning of the next potential loop and the end is located as before. It's matchratio is calculated and then compared with that of the of the previous one. If this matchratio is greater than that of the previous one, then it is stored and used for comparison with the following ones and hence the
loop with highest matchratio is chosen as the good loop. The temporary beginning and end of the potential loop with the highest matchratio becomes the beginning and end of the good loop. Next, all the good loops for all the activity records are located in the same manner. The activities within each of the good loops are matched. This is done by comparing the distances of each of the activity records with the reference distances for various stops. If their difference is within DeltaS2, then a matchvalue indicating an excellent match and the stop number from the reference file are assigned. If it is within DeltaS1 then a matchvalue indicating a good match and the stop number from the reference file are assigned. However, if it is beyond DeltaS1, then a matchvalue indicating a poor match along with the stop number giving the closest correspondence are assigned. This is done for all activity records within all good loops. The next stage involves matching the activity records which fall outside the good loops. This stray data is matched using two techniques namely backmatching and frontmatching techniques. In the backmatching technique, the stray data occurring before the beginning of the first good loop is matched. This is done again by comparing the difference between the actual and reference distances with DeltaS1. However the matching is backwards from the beginning of the good loop. In the frontmatching technique, the data occurring after the end of the last good loop is matched using the same procedure as above except that the matching is done in the front direction. Thus, in this way, matching is attempted for all the activity records that are created. This phase is represented by the flow diagram shown in the figure 7 in the next page.

5.1.1.3 Output File Storage Phase

In this phase, the processed data file is converted to an output file format and stored. This output file format depends upon the needs of the present and future management software of the transit company. It is observed that this is dependent upon the size, location and the management strategies of the transit property. However, owing to the modular nature
Retrieve Distance reference file and Schedule reference file used for matching.

Locate an activity record within DeltaT of the scheduled time for stop number 1 of loop ‘L’.

Locate end of loop by comparing cumulative distance with the reference loop distance. If difference is within DeltaL, then that activity record is the end of the loop.

Find the Match Ratio using the given formula, for the above loop and compare with that of other potential loops. Select loop with the best Match Ratio.

Match within loops. Check distance of activity records of log type ‘5’ (passenger activity) with the corresponding reference distances. If difference is within DeltaS1 or DeltaS2 then assign stop number to activity record with matchvalues of ‘2’ or ‘4’ indicating a good or excellent match. If not, find best correspondence for the activity record and assign the stop number with a matchvalue of ‘0’ indicating an inferior match. Do this for all activity records.

For activity records outside the good loops, do backmatching and frontmatching using the same allowable variance of distance concept as above.

Go to Output File Storage Phase.

Figure 7. Flow Diagram of Data Matching Phase.
of the algorithm, this can be easily taken care of. This phase is represented by the flow diagram shown in the Figure in the next page.

5.1.2 Algorithm Testing and Validation

Every algorithm that is built has to be tested over a wide range of data, representing all possible situations, to assess its usability and strength. This algorithm which matches real life data demands the creation of 'Test Scenarios'. These 'Test Scenarios' comprise of different data files each one representing a particular situation in reality. The data files are not big in terms of length per loop, since they get highly complicated and become both difficult and cumbersome to trace and validate. Hence, with this in view, the test route was designed to have about ten to twelve stops per loop. This would allow for comparatively easy tracing and validation.

It is to be recalled here that the algorithm is modular in nature. Hence, its analysis and testing would involve analyzing each of the modules. This step would require separating those modules which involve the main matching sequence and then testing to validate 'correct matching'. This testing is done for all data files to cover all the test scenarios. At the end of this step, the strength and consequently the usability of the algorithm would be highlighted.

Next, the variables in the algorithm are varied and the changes in matching that occur while running the test data files are noted. The variables that are varied are -

1. DeltaL.
2. DeltaT.
3. DeltaS1.
4. DeltaS2.

This would help test the sensitivity of the algorithm for the different scenarios.
Ascertain the existing software needs of the transit property, if any.

Ascertain the format of the processed file [Output file] for the existing management software, if any.

Create and store the processed data in the required format on Processing Computer for Report and Plot Generation.

Report and Plots Generation.

Figure 8. Flow Diagram of the Output File Storage Phase.
5.1.2.1 Creation of Test Scenarios.

This involves the following steps:

1. Conceptualization of real life situations.
2. Development of the Scenarios.
3. Creation of the Data Files.

Conceptualization of real life situations:

Situations in reality, as far as operations of buses on routes following a particular schedule are concerned, vary to a large extent. However, those situations which occur most commonly in practice can be divided into the following categories:

1. Routes operating on normal schedule for the full day.
2. Routes operating on normal schedule only for one loop in the morning peak / offpeak.
3. Routes operating on normal schedule for a part of the day greater than one loop.

Normal schedule is generally within DeltaT minutes of the actual scheduled time.

2. Development of Test Scenarios:

The development of Test Scenarios involve understanding the real life situations and applying them to the test under consideration. That is, a route operating on normal schedule for the full day would involve greater than 3 - 4 loops and hence reference files needed for this situation have to be created first. The times involved in the schedule must be realistic and they should resemble the situation in reality, closely. This involves calculating the distance on the loop, the number of stops and the time consumed to complete a loop. All these should confirm as close to a real life situation as possible. Next, the reference files for the route operating according to the scenarios were created, keeping all the factors mentioned above, in mind.
same was done for the other test scenarios. One has to recognize the changes to be made in
the reference files for different test scenarios and this involves understanding the difference
between various scenarios. Then, the next logical step was translating these differences to
reflect the changes to be made in the reference files.

Next, the different data files representing the various test scenarios were created, as
discussed in the next section.

### 5.1.2.2 Creation of Data Files:

The creation of the test data files for the various scenarios involves the following steps:

1. Understanding the test scenario under consideration.
2. Deciding on the upper limit of the number of activity records that occur within the data file,
i.e. Number of times the bus operating under particular scenario stopped.
3. Calculating the times and distances of each of the activity records from the beginning of the
loop making sure that they show the intended correspondence with the reference file of the
test scenario.

An example test data file creation using a table is shown in the Table in the next page.

### 5.1.2.3 Discussion of Test Results

All the datafiles along with their test reference files were run on the program under
test (APCDATA) and the following results were observed.

An observation of the results of runs with different system parameters, (i.e. for various
values of DeltaL, DeltaT, DeltaS1 and DeltaS2) indicates that the matching was done exactly
Table 4. Sample Test Data File Creation.

<table>
<thead>
<tr>
<th>Stop</th>
<th>#DFLS</th>
<th>CumDst</th>
<th>SchTme</th>
<th>PA</th>
<th>DFLA</th>
<th>CuDFLA</th>
<th>Time</th>
<th>DistAd</th>
<th>TimeAd</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>615</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>6 15</td>
<td>ADJT</td>
<td>ADJT</td>
</tr>
<tr>
<td>2</td>
<td>200</td>
<td>201</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>170</td>
<td>371</td>
<td>6 25</td>
<td>2</td>
<td>374</td>
<td>375</td>
<td>6 26.5</td>
<td>ADJT</td>
<td>ADJT</td>
</tr>
<tr>
<td>4</td>
<td>170</td>
<td>541</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>130</td>
<td>671</td>
<td>6 35</td>
<td>3</td>
<td>300</td>
<td>675</td>
<td>6 36.5</td>
<td>ADJT</td>
<td>ADJT</td>
</tr>
<tr>
<td>6</td>
<td>190</td>
<td>861</td>
<td>6 35</td>
<td>190</td>
<td>865</td>
<td>190</td>
<td></td>
<td>ADJT</td>
<td>ADJT</td>
</tr>
<tr>
<td>7</td>
<td>200</td>
<td>1061</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>390</td>
<td>1451</td>
<td>7 10</td>
<td>4</td>
<td>590</td>
<td>1455</td>
<td></td>
<td>ADJT</td>
<td>ADJT</td>
</tr>
<tr>
<td>9</td>
<td>290</td>
<td>1741</td>
<td>7 10</td>
<td>5</td>
<td>290</td>
<td>1745</td>
<td>7 12</td>
<td>ADJT</td>
<td>ADJT</td>
</tr>
<tr>
<td>10</td>
<td>160</td>
<td>1901</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>400</td>
<td>2301</td>
<td>6</td>
<td>6</td>
<td>555</td>
<td>2300</td>
<td></td>
<td>ADJT</td>
<td>ADJT</td>
</tr>
<tr>
<td>12</td>
<td>250</td>
<td>2551</td>
<td>7 30</td>
<td>7</td>
<td>251</td>
<td>2551</td>
<td>7 30</td>
<td>ADJT</td>
<td>ADJT</td>
</tr>
</tbody>
</table>
as intended and in all situations the results were in conjunction with what was conceptualized. These are shown in the Table in the next page.

5.1.3 Sensitivity Analysis on the Algorithm

A sensitivity analysis on the algorithm was performed to check the variation in the matching process with changes in the system parameters. This involved recognizing the parameters affecting the matching process and the roles played by these parameters during the various steps in matching. Thus, initially the following steps are involved -

1. Identifying the system variables involved in the matching process.
2. Identifying the roles played by the system variables and their effect on the matching process with a change in their values. This involves identifying the various stages in the matching process and the corresponding involvement of the system variables.

First of all, the system variables involved in the matching process are:

1. DeltaL - It is the number of odometer impulses in the range for determining when the matching routine should end or in other words, it is the range of difference or discrepancy allowed in locating the end of a potential loop.

2. DeltaT - It is the number of time clicks (15 second units) in the range for determining an activity record as the beginning of a potential loop.

3. DeltS1 - It is the number of odometer impulses which set the outer limit of a positive match.
Table 5. Results from different test scenarios.

<table>
<thead>
<tr>
<th>Run #</th>
<th>DeltaL</th>
<th>DeltaT</th>
<th>DeltaS1</th>
<th>DeltaS2</th>
<th>Matchng</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15,20,25</td>
<td>36</td>
<td>6</td>
<td>3</td>
<td>EXACT</td>
</tr>
<tr>
<td>2</td>
<td>30,35,40</td>
<td>36</td>
<td>6</td>
<td>3</td>
<td>EXACT</td>
</tr>
<tr>
<td>3</td>
<td>45,50,55</td>
<td>36</td>
<td>6</td>
<td>3</td>
<td>EXACT</td>
</tr>
<tr>
<td>2</td>
<td>35</td>
<td>12,16,20</td>
<td>6</td>
<td>3</td>
<td>EXACT</td>
</tr>
<tr>
<td>2</td>
<td>35</td>
<td>24,28,32</td>
<td>6</td>
<td>3</td>
<td>EXACT</td>
</tr>
<tr>
<td>2</td>
<td>35</td>
<td>32,40,44</td>
<td>6</td>
<td>3</td>
<td>EXACT</td>
</tr>
<tr>
<td>3</td>
<td>35</td>
<td>40</td>
<td>4,5,6</td>
<td>3</td>
<td>EXACT</td>
</tr>
<tr>
<td>3</td>
<td>35</td>
<td>40</td>
<td>7,8,9</td>
<td>3</td>
<td>EXACT</td>
</tr>
<tr>
<td>4</td>
<td>35</td>
<td>40</td>
<td>6</td>
<td>2,3</td>
<td>EXACT</td>
</tr>
<tr>
<td>4</td>
<td>35</td>
<td>40</td>
<td>6</td>
<td>4,5</td>
<td>EXACT</td>
</tr>
</tbody>
</table>
4. DeltaS2 - It is the number of odometer impulses which set the inner limit of positive match. Here, DeltaS2 << DeltaS1.

Next, to identify the roles played by the system variables and their effect on the matching process, the various stages or steps in the matching process have to be recognized. The various steps in the matching process from the point of view of performing sensitivity analysis are:

- Step 1 - Location of the beginning of a potential loop.
- Step 2 - Location of the end of a potential loop.
- Step 3 - Calculation of the Match Ratio for the potential loop.
- Step 4 - Determination of the best Match Ratio and consequently the good loop.
- Step 5 - Determination of all good loops.
- Step 6 - Matching within good loops.
- Step 7 - Matching outside good loops using backmatching and frontmatching techniques.

The roles played by the various parameters and their effect on the matching process are detailed in the following section.

5.1.3.1 Variation of the matching process with DeltaT

DeltaT: This is involved only in the location of the beginning of a potential loop ie. Step 1. Increasing the value of this parameter may lead to a greater number of iterations of the algorithm. However, it is to be chosen so that it would account for the worst possible delay...
at the beginning of a loop of the particular route. A sensitivity analysis using a real life data file shows that DeltaT does not affect the overall matching process. This is illustrated in the Table in the next page.

**DeltaL:** This is involved only in Step 2 i.e. location of the end of a potential loop. There is absolutely no change in the matching process with a change in DeltaL.

This is shown in the Table in the next page.

### 5.1.3.2 Variation of the matching process with DeltaS1

**DeltaS1:** This is involved in a number of steps of the matching process namely Steps 3, 4, 5, 6, 7 i.e. calculation of match ratio, determination of a good loop, determination of all good loops, matching within the good loops and matching outside the good loops. Hence, as far as the overall matching process goes, this parameter plays a crucial role and determines the actual precision of the matching process. The value for this parameter has to be very carefully selected so that most of the unforeseen circumstances like the bus stopping away from a stop due to one or more vehicles parked there etc. are taken care of. Also, the value of this parameter has to be less than the distance between any two stops. The variations in the matching process with change in DeltaS1 are indicated in the Tables in the next page. A mild effect is observed in Steps 6 and 7 but the overall matching process remains essentially the same.

### 5.1.3.3 Variation of the matching process with DeltaS2

**DeltaS2:** This is involved only in Step 6 namely matching within good loops with a high precision. There is practically no change in the matching process with a change in DeltaS2.
Table 6. Variation of the Matching Process with DeltaT

DeltaL = 35, DeltaS1 = 6, DeltaS2 = 3.

<table>
<thead>
<tr>
<th>DeltaT</th>
<th>Step 1</th>
<th>Step 2</th>
<th>Step 3</th>
<th>Step 4</th>
<th>Step 5</th>
<th>Step 6</th>
<th>Step 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>04</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>08</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>12</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>16</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>20</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>24</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>28</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>32</td>
<td>B</td>
<td>A</td>
<td>S</td>
<td>L</td>
<td>I</td>
<td>N</td>
<td>E</td>
</tr>
<tr>
<td>36</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>40</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>44</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>48</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

N = No Change. P = Partly Changed (5 - 10 percent).
Y = Change > 10 percent. BASLINE = Baseline for comparison.
Table 7. Variation of the Matching Process with DeltaL.

DeltaT = 32, DeltaS1 = 6, DeltaS2 = 3.

<table>
<thead>
<tr>
<th>DeltaL</th>
<th>Step 1</th>
<th>Step 2</th>
<th>Step 3</th>
<th>Step 4</th>
<th>Step 5</th>
<th>Step 6</th>
<th>Step 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>05</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>10</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>15</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>20</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>25</td>
<td>B</td>
<td>A</td>
<td>S</td>
<td>L</td>
<td>I</td>
<td>N</td>
<td>E</td>
</tr>
<tr>
<td>30</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>35</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>40</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>45</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

N = No Change. P = Partly Changed. BASELINE = Baseline for comparison.
Table 8. Variation of the matching process with DeltaS1 (DeltaT = 36).

DeltaL = 35, DeltaT = 36, DeltaS2 = 3.

<table>
<thead>
<tr>
<th>DeltaS1</th>
<th>Step 1</th>
<th>Step 2</th>
<th>Step 3</th>
<th>Step 4</th>
<th>Step 5</th>
<th>Step 6</th>
<th>Step 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>5</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>6</td>
<td>B</td>
<td>A</td>
<td>S</td>
<td>L</td>
<td>I</td>
<td>N</td>
<td>E</td>
</tr>
<tr>
<td>7</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>8</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>9</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>10</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>P</td>
</tr>
</tbody>
</table>

N = No Change. P = Partly Changed. BASLINE = Baseline for comparison.
P = Partly Changed (5 to 10 percent).
Table 9. Variation of the matching process with DeltaS1 (DeltaT = 40)

\( \text{DeltaL} = 35, \text{DeltaT} = 40, \text{DeltaS2} = 3. \)

<table>
<thead>
<tr>
<th>DeltaS1</th>
<th>Step 1</th>
<th>Step 2</th>
<th>Step 3</th>
<th>Step 4</th>
<th>Step 5</th>
<th>Step 6</th>
<th>Step 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>5</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>6</td>
<td>B</td>
<td>A</td>
<td>S</td>
<td>L</td>
<td>I</td>
<td>N</td>
<td>E</td>
</tr>
<tr>
<td>7</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>8</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>9</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>10</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
</tbody>
</table>

\( N = \text{No Change}. \ P = \text{Partly Changed}. \ \text{BASLINE} = \text{Baseline for comparison}. \)
This is shown in Table 9. More activity records were matched when the allowable variance for precise matching was increased. The change or difference was very negligible. In other words, additional number of activity records got matched.

5.1.3.4 Conclusions from the Sensitivity Analysis.

Thus, from the sensitivity analysis, it can be concluded that the algorithm does not change appreciably in its overall matching process with changes in the system parameters indicating that a robust and reliable routine is now available for use with the ADAS hardware. However, the variables affecting the matching in the order of their importance are DeltaS1, DeltaT, DeltaL and DeltaS2.

5.1.4 Advantages of the Algorithm

The advantages of this algorithm are multifarious. First of all, the techniques used for matching are simple. They are easy to trace since the algorithm is set up modularly. The concept of 'Matchratio' used in the matching process facilitates the selection of the best matched loop, possible. Also, the match values that are generated by the algorithm give the comparative goodness of the match of the activities. For example, a match value of '4' indicates that the activity was matched within a small variance 'DeltaS2'. A match value of '2' indicates that the activity was matched within a slightly larger variance of 'DeltaS1'. Both these match values are generated when the activities occur within a good loop. The stray data when matched within a variance of 'DeltaS1' by backmatching and frontmatching are given match values of '3' and '1' respectively. Thus, match values '1', '2' and '3' indicate the same precision in matching except that the match value of '2' is given only when the activity was
Table 10. Variation of the matching process with DeltaS2.

\( \text{DeltaL} = 35, \text{DeltaT} = 36. \)

<table>
<thead>
<tr>
<th>DeltaS1</th>
<th>DeltaS2</th>
<th>Step 1</th>
<th>Step 2</th>
<th>Step 3</th>
<th>Step 4</th>
<th>Step 5</th>
<th>Step 6</th>
<th>Step 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>3</td>
<td>B</td>
<td>A</td>
<td>S</td>
<td>L</td>
<td>I</td>
<td>N</td>
<td>E</td>
</tr>
<tr>
<td>4</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N*</td>
<td>N*</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N*</td>
<td>N*</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>B</td>
<td>A</td>
<td>S</td>
<td>L</td>
<td>I</td>
<td>N</td>
<td>E</td>
</tr>
<tr>
<td>4</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N*</td>
<td>N*</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N*</td>
<td>N*</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>B</td>
<td>A</td>
<td>S</td>
<td>L</td>
<td>I</td>
<td>N</td>
<td>E</td>
</tr>
<tr>
<td>4</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N*</td>
<td>N*</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N*</td>
<td>N*</td>
<td></td>
</tr>
</tbody>
</table>

\( N = \text{No Change.} \quad N^* = \text{Additional Matches.} \quad \text{BASLINE} = \text{Baseline for comparison.} \)
matched within the loop. Finally, a match value of ‘0’ indicates that the activity was not matched within the upper limit of variance ‘DeltaS1’ but even here the best possible match is done. Thus, the algorithm attempts a technique which allows for as precise a match as possible.

The activities that occur before the operation of the first good loop for the data collection period, as well as those that occur after the end of the last loop are all indicated by a match value of ‘3’ and ‘1’ respectively. Those that cannot be matched also indicate detours between operations of good loops. A series of match values of ‘1’s and ‘3’s between any two good loops indicate the activities occurred when the bus was off schedule by an amount greater than DeltaT. All this put together shows that this algorithm opens up excellent potential for further development of management software.

Also, the precision in matching is left as a variable to be set by the transit agency authorities and thus the algorithm is flexible in nature, too. This means that their values can be changed by the transit agency depending on their needs. The various disk drives where all the files are stored are again left as variables in the external file. This allows for the convenient location of all the files used in the operation of the software. This program written in BASICA allows itself to be compiled using either Microsoft’s Quickbasic or Borland Turbobasic, so that the execution time is reduced.

5.1.5 Assumptions and Limitations of the Algorithm

It should be recognized that there are a few implicit assumptions in the algorithm. The bus using the ADAS is assumed to be plying on a fixed route. The data is assumed to be collected for a greater than one loop length of the route. The bus is assumed to be operating on the normal route at normal schedule. Any changes have to be incorporated in the reference files before matching. And the operator is assumed not to allow boarding and alighting of
passengers in between stops along the route. These assumptions, however, are valid since the events disrupting them are very rare.

The algorithm calculates the time of the first activity by backtracking from the time of data dumping. The time of data dumping is manually input by the person doing the 'data dumping' process. Thus, the accuracy of the schedule adherence depends on the precision of the time manually input by that person. Also, if the time of data dumping is incorrect, then that error gets automatically transferred to the first activity being matched and depending on the magnitude of that error, the matching process can get thrown off. However, this is not seen as a serious limitation, since, the person doing the 'data dumping' process can be instructed to enter the time accurately. Also, with the integration of a clock chip into the circuitry, this problem can be eliminated.

The result of all the tests have now led us to believe that there is no real need for radio signposts along the routes for establishing locational accuracy. In this system, the algorithm is robust and can lock in and match the activities to stops.
5.2 Generation of Bus Stop Listing

Once the raw data file is processed and the output file is stored in a particular format, a Bus Stop Listing is generated. This is done by choosing the 'Generate Bus Stop Listing' option. This listing provides general information like the block number, route number, route name and the date of creation of the file. It also gives some crucial information like the locations and times where the passenger activity occurred, the precision of the above indicated by matchvalue(M), the distance and time (Dist & Depart) at which each of the activity records occurred, the number of passengers boarding and alighting(Ons & Offs) at each stop and the passenger load (Pass) on the bus at each stop. Also, at timepoints, the scheduled time (Sched) and the deviations(Dev) of the bus at these stops are indicated. The schedule deviations at particular timepoints along the route, serve as indicators of goodness of design of schedules. They also help in proper design and maintenance of routes and schedules. This, in turn would increase the efficiency and reliability of service of various routes, especially when schedule adherence is very critically regarded by choice riders. A sample output of the 'Bus Stop Listing' of the Roanoke Valley Metro APC System in Virginia is shown in the Table in the next page.
Table 11. Sample Sheet of Bus Stop Listing.

<table>
<thead>
<tr>
<th>STOP#</th>
<th>STOP NAME</th>
<th>M</th>
<th>DIST</th>
<th>DEPART</th>
<th>SCHED</th>
<th>DEV</th>
<th>ONS</th>
<th>OFFS</th>
<th>PASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>151</td>
<td>17TH/LOUDON</td>
<td>0</td>
<td>65.62</td>
<td>13:41:0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>156</td>
<td>22ND/LOUDON</td>
<td>4</td>
<td>65.94</td>
<td>13:43:15</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>181</td>
<td>GUN CLUB/MELROSE</td>
<td>0</td>
<td>69.32</td>
<td>14:00:30</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>RKE.SALEM PLAZA</td>
<td>4</td>
<td>72.90</td>
<td>14:18:0</td>
<td>14:15:0 - 3:0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>MELROSE/35TH</td>
<td>4</td>
<td>73.62</td>
<td>14:20:45</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>MELROSE/LAFAYETTE</td>
<td>4</td>
<td>74.54</td>
<td>14:23:45</td>
<td>14:22:0 - 1:45</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>MELROSE/24TH</td>
<td>0</td>
<td>77.45</td>
<td>14:35:15</td>
<td>14:37:15</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>WOOLWORTH'S</td>
<td>4</td>
<td>77.46</td>
<td>14:37:15</td>
<td>14:40:0 - 9:15</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>JEFF/CHURCH</td>
<td>4</td>
<td>77.60</td>
<td>14:41:15</td>
<td>14:42:30</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>MAIDEN LAKEN/DUNMORE</td>
<td>0</td>
<td>80.43</td>
<td>14:54:15</td>
<td>14:55:15</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>MEMORIAL/2229</td>
<td>0</td>
<td>80.52</td>
<td>14:54:45</td>
<td>14:56:0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>MEMORIAL/WESTOVER</td>
<td>0</td>
<td>81.98</td>
<td>15:00:15</td>
<td>15:02:0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>105</td>
<td>HAMPTON/DENNISTON</td>
<td>0</td>
<td>83.00</td>
<td>15:30:30</td>
<td>15:32:15</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>116</td>
<td>MAIN/FERDINAND</td>
<td>0</td>
<td>83.32</td>
<td>15:32:15</td>
<td>15:34:0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>CHURCH/5TH</td>
<td>0</td>
<td>85.59</td>
<td>15:15:45</td>
<td>15:17:45</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>135</td>
<td>RORER/5TH</td>
<td>0</td>
<td>85.63</td>
<td>15:15:45</td>
<td>15:17:45</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>169</td>
<td>35TH/MELROSE</td>
<td>0</td>
<td>89.76</td>
<td>15:30:45</td>
<td>15:32:15</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>RKE.SALEM PLAZA</td>
<td>4</td>
<td>90.41</td>
<td>15:34:0</td>
<td>15:43:30</td>
<td>15:40:0 - 3:30</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>MELROSE/FOREST PARK</td>
<td>2</td>
<td>92.14</td>
<td>15:47:45</td>
<td>15:50:30</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>SALEMAN/24TH</td>
<td>4</td>
<td>94.22</td>
<td>15:53:30</td>
<td>15:56:15</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>WOOLWORTH'S</td>
<td>4</td>
<td>95.18</td>
<td>16:00:15</td>
<td>16:05:0</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>JEFF/CHURCH</td>
<td>2</td>
<td>95.31</td>
<td>16:07:0</td>
<td>16:15:0 - 2:0</td>
<td>2</td>
<td>1</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>169</td>
<td>35TH/MELROSE</td>
<td>0</td>
<td>95.33</td>
<td>16:15:0 - 2:0</td>
<td>16:30:30</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>
6.0 MANAGEMENT REPORTS AND PLOTS

6.1 Overview

Once the raw data file has been processed and an output file created, certain crucial reports and plots are generated. These reports and plots deliver meaningful information into the hands of transit management. Crucial decisions regarding route design, schedules, fleet adequacy, quality and level of service offered, can be made with more confidence. Also the graphical information support rendered through this package would prove invaluable to the management in planning and operational activities since certain types of information are much easier to interpret if they are in graphical form as opposed to tabular or report form.

This chapter is dedicated to describing the various software packages made available through the years of ADAS software development at Virginia Tech. The management software can be divided into the following:

1. Management Reports.
6.2 Management Reports

There are several useful reports that can be generated. These reports yield a wealth of information to the decision makers aiding them in their activities. The information presented is in tabular form so that correct interpretation is made easy. The following are the various reports available:

1. Time Point Profile Report.

The Timepoint Profile file maintains records for each trip, with stop statistics between timepoints. The Section 15 Daily report gives very useful passenger ridership information and serves as input for the creation of a Cumulative Section 15 file called Section 15 Annual Report which has to be submitted to UMTA by every transit property. The Summary Report gives daily information of the major events on the route. Also, this file flags up abnormalities on the route, regarding passenger ridership, schedule deviations, etc.

6.2.1 Timepoint Profile Report

The Timepoint Profile Report maintains records for each trip, with stop statistics between timepoints. It is a very useful report for monitoring schedule performance. This report aggregates some of the data from the bus stop listing while providing additional information which the listing does not. The following statistics are reported for every timepoint:

- The Stop Number and Stop Name.
• The passenger load at that point.

• The scheduled time for that timepoint.

• The actual time at which the bus left the stop.

• The scheduled deviation at that timepoint given by the difference of the above two.

• The number of passenger ons and offs at that point.

• The maximum passenger load on the bus.

• The stop number at which the above occurred.

• The number of inter-timepoint stops the bus made.

• The inter-timepoint distance travelled between the two consecutive timepoints.

• The inter-timepoint time taken to travel between the two consecutive timepoints.

• The inter-timepoint operating speed of the bus.

The information presented in the report above can give an indication of certain very critical performance indicators like schedule adherence (schedule deviation), passenger activity (passenger ons and offs), travel time (time and distance between timepoints) and adequacy of speed (operating speed). If the operating speed between the timepoints falls below a certain value, then the ridership may be affected.

A sample page of Timepoint Profile Report file is shown in the Table in the next page.
### Table 12. Sample Sheet of Timepoint Profile Report.

**Roanoke Valley Metro**
**Time Point Profile Report**
**Prepared 01-01-1980**
**Block # 63  Route # 5  Melrose/Virginia Hts.  10-21-86**

<table>
<thead>
<tr>
<th>STOP#</th>
<th>STOP NAME</th>
<th>LOAD</th>
<th>DEPART</th>
<th>SCHED</th>
<th>DIST</th>
<th>TIME</th>
<th>SPEED</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RKE.SALEM PLAZA</td>
<td>3</td>
<td>6:37:0</td>
<td>6:35:0</td>
<td>2:0</td>
<td>14.55</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>CHURCH/JEFF</td>
<td>6</td>
<td>7:30:15</td>
<td>7:25:0</td>
<td>5:15</td>
<td>12.60</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>RKE.SALEM PLAZA</td>
<td>3</td>
<td>7:51:15</td>
<td>7:50:0</td>
<td>1:15</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STOP#</th>
<th>STOP NAME</th>
<th>LOAD</th>
<th>DEPART</th>
<th>SCHED</th>
<th>DIST</th>
<th>TIME</th>
<th>SPEED</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RKE.SALEM PLAZA</td>
<td>3</td>
<td>7:51:15</td>
<td>7:50:0</td>
<td>1:15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>MELROSE/LAFAYETTE</td>
<td>7</td>
<td>7:58:0</td>
<td>7:57:0</td>
<td>1:0</td>
<td>19.84</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>MALVERN/EDGEWOOD</td>
<td>8</td>
<td>8:35:0</td>
<td>8:35:0</td>
<td>0:0</td>
<td>12.15</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>CHURCH/JEFF</td>
<td>17</td>
<td>9:1:15</td>
<td>8:50:0</td>
<td>11:15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>RKE.SALEM PLAZA</td>
<td>1</td>
<td>9:27:30</td>
<td>9:25:0</td>
<td>2:30</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STOP#</th>
<th>STOP NAME</th>
<th>LOAD</th>
<th>DEPART</th>
<th>SCHED</th>
<th>DIST</th>
<th>TIME</th>
<th>SPEED</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RKE.SALEM PLAZA</td>
<td>1</td>
<td>9:27:30</td>
<td>9:25:0</td>
<td>2:30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>MELROSE/LAFAYETTE</td>
<td>4</td>
<td>9:32:30</td>
<td>9:32:0</td>
<td>0:30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>86</td>
<td>MALVERN/EDGEWOOD</td>
<td>1</td>
<td>10:5:00</td>
<td>10:5:00</td>
<td>0:0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>CHURCH/JEFF</td>
<td>1</td>
<td>10:21:15</td>
<td>10:20:0</td>
<td>1:15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>RKE.SALEM PLAZA</td>
<td>1</td>
<td>14:18:0</td>
<td>14:15:0</td>
<td>3:0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STOP#</th>
<th>STOP NAME</th>
<th>LOAD</th>
<th>DEPART</th>
<th>SCHED</th>
<th>DIST</th>
<th>TIME</th>
<th>SPEED</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RKE.SALEM PLAZA</td>
<td>1</td>
<td>14:18:0</td>
<td>14:15:0</td>
<td>3:0</td>
<td>17.11</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>MELROSE/LAFAYETTE</td>
<td>3</td>
<td>14:22:45</td>
<td>14:22:0</td>
<td>0:45</td>
<td>19.48</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>JEFF/CHURCH</td>
<td>5</td>
<td>14:41:15</td>
<td>14:40:0</td>
<td>1:15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>JEFF/CHURCH</td>
<td>6</td>
<td>14:49:15</td>
<td>14:40:0</td>
<td>9:15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>RKE.SALEM PLAZA</td>
<td>2</td>
<td>15:43:30</td>
<td>15:40:0</td>
<td>3:30</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6.2.2 Section 15 Daily Report

This report is a compilation of data obtained from various trips throughout the day on the same route. An aggregation of these daily reports for various routes gives the Cumulative Section 15 file. This is a mandatory report for all transit systems if they have to receive or continue to receive subsidies from the UMTA.

There are four different reporting levels as shown in the Table in the next page.

It can be seen from the table that the reporting level to be adopted, depends upon the size of the transit company. For example, a transit company with 300 revenue vehicles can use level B. However, as far as the federal government is concerned, any of the levels namely Level A, B or C satisfy the reporting requirements unless the transit company is applying for increased grants. Thus, it is imperative that the reporters check with the state or local authorities before selecting a reporting level.

At Roanoke Valley Metro, Form 406A is used for reporting. This annual report to UMTA shows operating data for fixed routes. Details such as passengers boarded, unlinked passenger trips, etc., are also indicated.

A sample Section 15 Daily Report is shown in the Table in the following page.

6.2.3 Section 15 Annual Report

This is the annual report under Section 15 requirements that is sent to UMTA by many transit companies to continue to receive operating subsidies. This report is formed by combining daily report files so that enough trips are included to satisfy the requirements for confidence and precision levels specified by UMTA. Also, the number of ADASs required to collect trip data that can be combined to ensure a statistically valid database is about ten
Table 13. Reporting Levels allowed for Transit Properties.

<table>
<thead>
<tr>
<th>Reporting Level</th>
<th>Level R</th>
<th>Level C</th>
<th>Level B</th>
<th>Level A</th>
</tr>
</thead>
<tbody>
<tr>
<td>required level</td>
<td>voluntary</td>
<td>voluntary</td>
<td>voluntary</td>
<td>voluntary</td>
</tr>
<tr>
<td>all systems</td>
<td>suggested for &lt;100 revenue vehicles</td>
<td>suggested for 100-500 revenue vehicles</td>
<td>suggested for &gt; 500 revenue vehicles, and all rapid rail systems</td>
<td></td>
</tr>
</tbody>
</table>
Table 14. Sample Sheet of Section 15 Daily Record File.

<table>
<thead>
<tr>
<th>TIME PERIOD</th>
<th>AM PEAK</th>
<th>MIDDAY</th>
<th>PM PEAK</th>
<th>NIGHT</th>
<th>SATURDAY</th>
<th>SUNDAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passengers Boarded</td>
<td>19</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Passengers On Board</td>
<td>110</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bus Trip Distance</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Passenger Miles</td>
<td>47</td>
<td>37</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bus Trip Time</td>
<td>40</td>
<td>33</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Passenger Minutes</td>
<td>56</td>
<td>38</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Capacity Miles</td>
<td>551</td>
<td>552</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Seat Miles</td>
<td>351</td>
<td>351</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Trips in Sample</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

SAMPLE AVERAGES

| Unlinked Pass/Trip   | 19      | 15     | 0       | 0     | 0        | 0      |
| Pass. Miles/Trip     | 47      | 37     | 0       | 0     | 0        | 0      |
| Unl. Pass. Trip Time | 56      | 38     | 0       | 0     | 0        | 0      |
percent of the fleet size. For example, a transit company with a fleet size of 100 buses should have about ten buses equipped with ADAS.

The sample sizes required by UMTA, depend on the frequency of sampling. Also, in addition to meeting the sample size requirement, the samples should be selected at random. The block numbers on different routes vary and hence by varying the choice of the samples, randomness can be ensured.

This annual report gives certain useful aggregated passenger ridership information for various trips like the following:

- Passengers boarded.
- Passengers on board.
- Bus trip distance.
- Passenger miles.
- Bus trip time.
- Passenger minutes.
- Capacity miles.
- Seat miles.
- Trips in sample.

Sample averages like the following are also given:

- Unlinked passengers per trip.
- Passenger miles per trip.
- Unlinked passenger trip time.

Also, annual totals of the following are also presented:

- Unlinked passenger trips.
- Passenger miles.
These are estimates obtained by random sampling at specified confidence and precision levels of 95 percent and 10 percent respectively. The variable items presented above are for different time periods of a weekday such as AMPEAK, MIDDAY, PMPEAK, NIGHT and SATURDAYS and SUNDAYS.

The details of some of the variables above are:

1. **Trip**: It is defined by UMTA as a one-way trip from one terminal point to another in a regular revenue service.

2. **Average Weekday**: Average Weekday is interpreted as being a typical or representative weekday in the operation of the transit system.

3. **Time Periods**: An average weekday has four time periods.
   - 1. AM Peak - Starting at 6 a.m and ending at 9 a.m
   - 2. Midday - Starting at 9 a.m and ending at 3 p.m
   - 3. PM Peak - Starting at 3 p.m and ending at 7 p.m
   - 4. Night - After 7 p.m

4. **Unlinked Passenger Trips**: This is obtained as follows: a. Divide the sample total number of passengers boarded by the total number of trips in the sample to get the average number of unlinked passengers per trip. b. Multiply the average unlinked passengers per trip by the total number of bus trips.

5. **Passenger Miles**: This is obtained as follows: a. Divide the sample total passenger miles by the total number of trips in the sample to get the average passenger miles per trip. b. Multiply the average passenger miles per trip by total number of bus trips.
The other variables are self-explanatory. This report is generally submitted at the end of the fiscal year.

A sample Annual report to UMTA from Roanoke Valley Metro is shown in the Table.

6.2.3.1 Summary Report File:

The Summary Report File is a report to the management of the bus routes performance for the day. It flags up abnormalities in the route for which data was collected by presenting certain very critical performance indicators in a tabular form. These indicators are:

1. Maximum Passenger Load Details:
All details of all the stops showing maximum passenger load on the bus is shown here. The details include the stop numbers, stop names, times and distances at which this event occurred.

2. Busy Stops - Details:
Here, all the crucial details of stops along the route that have a lot of passenger activity are given. A criterion called 'Activity Factor' which is the number of passenger ons at a stop equal to or greater than a particular value (say 5) is used to determine the busy stops. If the number of passengers boarding a bus is greater than the activity factor (say 5) at a particular stop, then that stop is taken a busy stop and all the pertinent details like the stopnumber, stopname, time, distance and the passenger load on the bus are all presented. This is done for all the bus stops along the route for the whole day.

3. Maximum Deviation Details:
Here the schedule deviations which were largest for the day at timepoints are all presented. The details that accompany the stopnumbers of the timepoints with the largest schedule deviations are the stopnames, the times and distances. This information would be particularly useful for operations managers dealing with schedules and schedule adherence.

4. Total Distance made on the route:
Table 15. Sample Sheet of Section 15 Annual Report File.

<table>
<thead>
<tr>
<th>TIME PERIOD</th>
<th>AM PEAK</th>
<th>MIDDAY</th>
<th>PM PEAK</th>
<th>NIGHT</th>
<th>SATDAY</th>
<th>SUNDAY</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passengers Boarded</td>
<td>19</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>34</td>
</tr>
<tr>
<td>Pass. On Board</td>
<td>110</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>190</td>
</tr>
<tr>
<td>Bus Trip Distance</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>Passenger Miles</td>
<td>47</td>
<td>37</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>84</td>
</tr>
<tr>
<td>Bus Trip Time</td>
<td>40</td>
<td>33</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>73</td>
</tr>
<tr>
<td>Passenger Minutes</td>
<td>56</td>
<td>38</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>93</td>
</tr>
<tr>
<td>Capacity Miles</td>
<td>551</td>
<td>552</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1103</td>
</tr>
<tr>
<td>Seat Miles</td>
<td>351</td>
<td>351</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>702</td>
</tr>
<tr>
<td>Trips in Sample</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Total Trips</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

SAMPLE AVERAGES

| Unl. Pass./Trip | 19 | 15 | 0 | 0 | 0 | 0 | 34 |
| Pass. Miles/Trip | 47 | 37 | 0 | 0 | 0 | 0 | 84 |
| Unl Pass Trip Time | 56 | 38 | 0 | 0 | 0 | 0 | 93 |

ANNUAL TOTALS

| Unl. Pass. Trips | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Passenger Miles | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
This variable gives the total distance travelled in miles for the whole day.

5. Total number of passengers carried per mile:

This item is obtained by accumulating the passenger ons for the whole day and then dividing the sum by the total distance travelled in miles for the whole day.

6. Average speed for the day:

The average speed is calculated by dividing the total distance travelled for the day by the time of operation of the route. This value should not be too low. Generally, for normal operations of the routes at normal conditions, the average speed for the day has to remain the same.

The summary report file can be obtained by running the reports program and choosing the option ‘4’ on the menu.

The program asks for which output file you want the report. Upon giving the filename, the output file is retrieved.

Next the question ‘Do you want a screen display or a printed copy?’ appears. Upon choice, the concerned reference files are retrieved and the report in a tabular form is displayed or printed. A sample page of the Summary Report File is shown in the Table in the next page.

### 6.2.4 Management Plots:

Computer generated graphics are used to create the management Plots. The following plots can be generated using the plots program:

1. Passenger Load Plot
2. Route Demand Plot
3. Route Evaluation Plot
4. Schedule Evaluation Plot
### Table 16. Sample Sheet of Summary Report File.

<table>
<thead>
<tr>
<th>STOP #</th>
<th>STOP NAME</th>
<th>MAXIMUM LOAD</th>
<th>DISTANCE</th>
<th>TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>150</td>
<td>16TH/LOUDON</td>
<td>18</td>
<td>45.79</td>
<td>9:4:15</td>
</tr>
<tr>
<td>154</td>
<td>20TH/LOUDON</td>
<td>18</td>
<td>46.14</td>
<td>9:6:30</td>
</tr>
<tr>
<td>158</td>
<td>24TH/MELROSE</td>
<td>18</td>
<td>46.57</td>
<td>9:8:0</td>
</tr>
</tbody>
</table>

#### Maximum Passenger Load Details

#### Busy Stops Details

<table>
<thead>
<tr>
<th>STOP #</th>
<th>STOP NAME</th>
<th>ONS</th>
<th>OFFS</th>
<th>DISTANCE</th>
<th>TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>127</td>
<td>WOOLWORTH'S</td>
<td>0</td>
<td>6</td>
<td>63.35</td>
<td>10:20:30</td>
</tr>
<tr>
<td>127</td>
<td>WOOLWORTH'S</td>
<td>7</td>
<td>1</td>
<td>102.94</td>
<td>16:41:0</td>
</tr>
</tbody>
</table>

The total distance covered for the day is 134.5094 miles.
The total number of passengers carried per mile is 1.3605.
The average speed for the day is 5.103913 MPH.

---

MANAGEMENT REPORTS AND PLOTS 78
6.2.4.1 Passenger Load Plot:

When the option ‘1’ in the Plots program menu is selected, the passenger load plot can be generated. This program gives a plot of the passenger load vs time of day. A sample output is attached as shown in the Figure in the next page. This plot can be very valuable in determining optimal headways and adequacy of vehicles at providing seating which matches the demand.

6.2.4.2 Route Demand Plot:

This plot can be generated using the option ‘2’ in the plots program menu. The steps involved in its generation are similar to those in the generation of passenger load plot. This option gives a bar chart indicating passengers boarded per hour versus time of day. A sample output is shown in the Figure.

This plot can be very helpful in determining ‘peaking’ effects and for identifying the required amount of service at various times throughout the day.

6.2.4.3 Route Evaluation Plot:

This plot can be generated using the plots program. This plot shows the passenger load at the various timepoints along the route. A timepoint is defined as ‘a designated point on a route where specific vehicle arrival and/or departure times are scheduled’.

This plot gives the management an idea of the trend of passenger boarding. The plot is obtained for all the loops of operation of a particular route for the whole day.

From this plot, it can be observed that the maximum and minimum load points at various times along the route can be noted. This would help the management evaluate the
Figure 9. Passenger Load Plot.
Figure 10. Route Demand Plot.
adequacy of service along the route both in terms of schedule as well as the number of buses plying on the route. Thus, this can be a very useful decision aiding tool.

The operation of this program is very similar to that of the previous plots. A sample output of the Route Evaluation Plot is shown in the Figure in the next page.

6.2.4.4 **Schedule Evaluation Plot:**

This plot can be generated by selecting the Schedule Evaluation Plot option from the menu. A plot of timepoints versus Schedule Deviation is obtained. This plot gives an idea of how well the schedule for the route has been set up for the entire day. An indication of schedule adherence of the bus for the particular route for displayed.

Depending upon the magnitudes of the Schedule Deviation, an idea of how well the route has been in operation for the day can be got. Also, if ‘peaks’ of schedule deviation keep appearing at the same timepoints for different days of operation, then the schedule might be too tight for the operators to keep up and changing the schedule then would not only relieve the pressure on the operators but also most certainly benefit the passengers in terms of waiting time.

Thus, this plot helps operations and schedule managers to better design the schedule along the route for better service.

The steps of this program are similar to that of Route Evaluation Plot and they are:

1. The Schedule Evaluation Plot option is selected from the menu. The program then asks which output file to retrieve in order to create the plot and responds to the users input.
2. The program displays a message indicating that a plot can be obtained by pressing the shift - PrtSc combination once the plot is completed on the screen.
3. The axes, headings and labels are printed.
4. The Schedule Deviation vs Timepoints is plotted on the screen.
Figure 11. Route Evaluation Plot.
The Figure on the next page shows a sample Schedule Evaluation Plot.

6.2.5 Summary:

In summary, the Management Reports and Plots can be generated everyday after the data has been collected along the particular route using the ADAS hardware and then processed using the APCDATA program. These reports and plots serve as very useful system performance indicators and act as decision aiding tools for the management in providing improved, economical and efficient service to the public.

The operation of the all the software addressed to in this chapter as well as in previous chapters is given in the 'Users Manual' presented in Appendix.
Figure 12. Schedule Evaluation Plot.
7.0 ADAS PROJECT AT TRT NORFOLK, VA.

7.1 Overview

The management of Tidewater Rapid Transit (TRT), the local transit company in the Tidewater area (Norfolk and Portsmouth) decided to implement an Automatic Data Acquisition System to collect ridership data as an experimental mass transit project supported by the Virginia Department of Highways and Transportation. The obvious advantages of ADAS could be the main impetus to the management to procure and implement such a system.

The project at TRT, Norfolk, has evolved under the guidance of Dr. A.G. Hobeika of the Civil Engineering Department and Dr. C.E. Nunnaly of the Electrical Engineering Department at Virginia Tech.

In this chapter, the evolution of the project from its birth to the stage when the implementation of the ADAS system will be in place, are described. The various stages in this project development have been the following:

- Procurement of the essential modules and other hardware components from M/S Red Pine Instruments and then testing them all at the laboratory.
• Deduction of the project implementation plan and dissemination of basic information to the concerned transit agency authorities.

• Recognition of hardware and software needs of the transit agency.

• Actual installation of the equipment on the buses.

• Calibration of the odometer impulse using the signal from the Transmission Pick up.

• Calibration of required software and their development.

7.2 Equipment at TRT, Norfolk:

The experience at Valley Metro has been that installation of a new ADAS is not without its problems. However, with the experience gained by the suppliers during this period, some of the equipment problems have been worked out. This section describes some of these problems, their solutions and ways in which they may be prevented. It also gives a detailed description of the particular equipment used at TRT, Norfolk so that this system can be compared to other ADAS installations. The equipment at TRT can be divided into the following types:

1. On-board Equipment.

2. Data Retrieval/Processing Equipment.

1. On-Board Equipment:

   Counters:
The counters selected for the project are infra-red, dual-beam types supplied by Red-Pine Instruments. There are a total of eight counters which function as two sets, one at each side of the doorway, directly opposite the other two for that door. Thus, there is a series of four light beams at each door. These sensors are skewed both horizontally and vertically by about 4° and are placed at a height of about 48" above the steps. The way, a passenger activity is recorded and the logic involved have already been discussed in chapter 3. The location of the sensors has been decided based on previous experience with Valley Metro at Roanoke, VA. The particular configuration, stated above was found effective at counting the passengers without creating false records. Of course, the new configuration needs to be checked under actual conditions to verify accuracy.

a. Distance Measuring Equipment:

The determination of distance is very important for ADA systems. With the experience gained at Valley Metro, the distance measurement equipment installation and usage has stabilized and the problems with variations in the impulses from the odometer have been alleviated.

Transmission Pick up: Examination revealed that electrical noise within the odometer itself was causing false signals to be sent to the onboard microprocessor. Hence, the decision to bypass the odometer altogether and to pick up the signal directly from the transmission drive coupling was made. Stemco Instruments of Dallas, Texas supplied a coupling which fits the Allison V-730 transmission and this was installed on the bus and clear signals received.

Impulse Converter: To take care of the fact that some odometer impulses were being lost at higher speeds because of their arrival at the microprocessor faster than it could handle, simple converter to divide the number of impulses coming from the
odometer pickup so that only every fourth one actually went to the microprocessor, was built. This converter has been built into the odometer Power Module and the conversion is done using jumpers. At present, the distance travelled per odometer impulse is 5.13 feet and calibration runs have shown that distance measurement is now remarkably consistent.

b. System Modules:

The equipment discussed in the previous pages serve to detect and transmit signals whenever certain events occur. The system modules receive those signals, interpret them and create and store the information. Together the modules form the ‘nucleus’ of the Automatic Data Acquisition System. All these modules are located behind the right front passenger seat, and the following are the modules involved:

- Front and Rear Passenger Count Modules: These modules interpret the sensor signals with the aid of a logic circuitry and thus they create all passenger activity records. These modules are similar to the ones used at Roanoke Valley Metro.

- Data Storage Module: This module forms the core of the system. It receives all the information from the other modules, creates appropriate logs, stores the records and then helps in dumping the set of these records to the external computer device for further processing.

- Diagnostic Module: This module is different from the one used at Roanoke Valley Metro. It has a lot more features and operates differently. This module provides diagnostic facilities for all other modules in the ADAS. It is a portable module with a 16-key keypad, 80 character alphanumeric LCD display and serial communications capabilities. Diagnostic functions are organized hierarchically in a tree structure form. On power up or by pressing key ‘E’ it resets Level 0 (root). The next level is entered by choosing the module identifier
keys. Access to diagnostics specific to a module or to module data bus can be invoked by a single key. Once a particular module has been selected, then the various diagnostic functions pertaining to that module or data bus can be selected by single keys. A menu helps user identify the function and its corresponding key.

- Odometer / Power Module: This module provides 12V to all the other modules, receives and records the odometer impulses.

c. Control Panel:

Access to the onboard equipment is by means of a control panel located behind the right front passenger seat just above the system modules. It contains the following components: a power on / off switch, power on / off indicator, a data transfer jack, which is a phono type jack which connects to retrieval computer, a diagnostic unit connection, a transmission signal port and a cover and a lock.

2. Data Retrieval / Processing Equipment:

Once the data has been collected using the onboard equipment, it is transferred and processed to create meaningful reports and plots. For doing so, a lap held or a portable computer is used. A communications software is used to dump (transfer) the data file to the storage device on the lap held or portable computer and then processing is done. The output is a file in the UMTA TRS80 Data File format which serves as input for Norfolk TRT's management software.

7.2.1 Calibration of Odometer Impulse

Once the equipment was installed, the transmission pickup signal was tested, proper jumpers were installed and then the odometer impulse was calibrated. This was done as
A distance of 200 feet was accurately measured and marked on the ground. The bus with the equipment in place, was driven over the distance several times, each time noting the distance in odometer impulses with the aid of the portable diagnostic unit. It was found, on the average, that 157 impulses were recorded for a distance of 200 feet. Next, the jumpers were installed in the proper place and then the bus run over the same distance. This was found, on the average, that 39 odometer impulses were recorded for 200 feet. This meant that, at present, 1 odometer click = 5.13 feet.

7.3 Software at TRT, Norfolk

TRT, Norfolk, has its own custom made management software on their Datapoint system. The software is written in a language called 'Databus'. The management software prepared by Virginia Tech has been discussed in Chapter 6. At the end of this phase of the project, it is intended that two parallel sets of reports and plots be generated and compared, one using the management software developed by Virginia Tech and the other using the custom made software of TRT, Norfolk. However, for doing the latter, the output file from the data processing program of ADAS software has to be converted to a different format (i.e. the format of UMTA TRS80 Data file). The reference files used by TRT are in a different form on a different system. Hence, an interfacing software is needed to convert the files to the format of the reference files used for ADAS software. A system's view of the ADAS implemented at Norfolk, TRT is shown in the figure.
Figure 13. Systems View of ADAS.
7.4 Needs of TRT, Norfolk

7.4.1 Interfacing Input Software:

This software has been designed to change the format of the reference files existing at TRT, Norfolk, to the format of the reference files used for the matching process in the APCDATA program built at Virginia Tech. The reference files at TRT, Norfolk, from this viewpoint, consists of two systems of files:

2. Headway Sheet System.

1. Brown Sheet System:

In the Brown Sheet System, the Brown Sheet Route file contains records for each route. There is a record for every intersection along the route both inbound and outbound. The first record contains zeroes in the intersection and intersection sequence. Each record has a length of 116 characters. The layout of the Brown Sheet Route file is shown in the figure in the next page.

From the figure, it can be seen that the brown sheet route file contains the route number, route direction (inbound or outbound), street names at intersections and the mileage post which is the accumulated distance in miles.

There are two data types involved namely A - Alphanumeric and N - Numeric. The size of each of the variables in the brown sheet route file is given in the third column and it indicates the number of characters allowed. A sample page of the Brown Sheet Route file is shown in the figure in the next page.

An inspection of the file reveals that the names and distances of all the stops on the route can be got. This means that the Names reference file as well as the Dist reference file...
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>SIZE</th>
<th>LOCATION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROUTE</td>
<td>A</td>
<td>4</td>
<td>1-4</td>
<td>Route Number</td>
</tr>
<tr>
<td>DIR</td>
<td>A</td>
<td>1</td>
<td>5-5</td>
<td>Direction</td>
</tr>
<tr>
<td>INTERS</td>
<td>A</td>
<td>30</td>
<td>6-35</td>
<td>Street Names at Intersection</td>
</tr>
<tr>
<td>IND</td>
<td>A</td>
<td>1</td>
<td>36-36</td>
<td>Intersection Sequence</td>
</tr>
<tr>
<td>LINKF</td>
<td>A</td>
<td>30</td>
<td>37-66</td>
<td>Link forward to next Intersect.</td>
</tr>
<tr>
<td>INDF</td>
<td>A</td>
<td>1</td>
<td>67-67</td>
<td>Sequence forward to next Intersection</td>
</tr>
<tr>
<td>LINKB</td>
<td>A</td>
<td>30</td>
<td>68-97</td>
<td>Link Backward to previous Intersection</td>
</tr>
<tr>
<td>INDB</td>
<td>A</td>
<td>1</td>
<td>98-98</td>
<td>Sequence Backward to previous Intersection</td>
</tr>
<tr>
<td>STPCD</td>
<td>A</td>
<td>5</td>
<td>99-103</td>
<td>STP Code</td>
</tr>
<tr>
<td>CITYCD</td>
<td>A</td>
<td>2</td>
<td>104-105</td>
<td>City Code</td>
</tr>
<tr>
<td>MSUS</td>
<td>N</td>
<td>3.2</td>
<td>106-111</td>
<td>Census Track</td>
</tr>
<tr>
<td>L/P</td>
<td>N</td>
<td>3.2</td>
<td>111-116</td>
<td>Mileage Post (accumulated mile)</td>
</tr>
</tbody>
</table>

TOTAL RECORD LENGTH: 116 characters

Figure 14. Lay Out of the Brown Sheet Route File (TRT, Norfolk).
### Bus Stop Information

<table>
<thead>
<tr>
<th>Bus Stop Name</th>
<th>Name</th>
<th>Code</th>
<th>City Code</th>
<th>Mile Post</th>
<th>Census Track</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gate 1</td>
<td>Little Creek</td>
<td>02</td>
<td>.00</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Shore Drive 1</td>
<td>Little Creek</td>
<td>02</td>
<td>.06</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Shore Drive 2</td>
<td>Dunning</td>
<td>02</td>
<td>.27</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Shore Drive 3</td>
<td>Middletown</td>
<td>02</td>
<td>.32</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Shore Drive 4</td>
<td>Pretty Lake</td>
<td>02</td>
<td>.63</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Shore Drive 5</td>
<td>Pleasant</td>
<td>02</td>
<td>.66</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Shore Drive 6</td>
<td>22nd Bay</td>
<td>02</td>
<td>.86</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Shore Drive 7</td>
<td>21st Bay</td>
<td>02</td>
<td>.92</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Shore Drive 8</td>
<td>Shore Drive</td>
<td>02</td>
<td>.98</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 1</td>
<td>13th Bay</td>
<td>02</td>
<td>1.18</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 2</td>
<td>18th Bay</td>
<td>02</td>
<td>1.19</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 3</td>
<td>17th Bay</td>
<td>02</td>
<td>1.24</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 4</td>
<td>16th Bay</td>
<td>02</td>
<td>1.30</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 5</td>
<td>15th Bay</td>
<td>02</td>
<td>1.44</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 6</td>
<td>14th Bay</td>
<td>02</td>
<td>1.57</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 7</td>
<td>9th Bay</td>
<td>02</td>
<td>1.70</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 8</td>
<td>7th Bay</td>
<td>02</td>
<td>1.83</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 9</td>
<td>5th Bay</td>
<td>02</td>
<td>1.94</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 10</td>
<td>3rd Bay</td>
<td>02</td>
<td>2.05</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 11</td>
<td>1st Bay</td>
<td>02</td>
<td>2.19</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 12</td>
<td>Middletown</td>
<td>02</td>
<td>2.30</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 13</td>
<td>Inlet</td>
<td>02</td>
<td>2.41</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 14</td>
<td>Middletown</td>
<td>02</td>
<td>2.69</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 15</td>
<td>Groves</td>
<td>02</td>
<td>2.76</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 16</td>
<td>Middletown</td>
<td>02</td>
<td>2.80</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 17</td>
<td>Willow</td>
<td>02</td>
<td>3.07</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 18</td>
<td>Sturges</td>
<td>02</td>
<td>3.22</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 19</td>
<td>Holiday Sands</td>
<td>02</td>
<td>3.31</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 20</td>
<td>Beachview</td>
<td>02</td>
<td>3.37</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 21</td>
<td>Warwick</td>
<td>02</td>
<td>3.39</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 22</td>
<td>Chesapeake St.</td>
<td>02</td>
<td>3.53</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 23</td>
<td>Eldora</td>
<td>02</td>
<td>3.72</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 24</td>
<td>Chesapeake Blvd</td>
<td>02</td>
<td>3.93</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 25</td>
<td>Norfolk</td>
<td>02</td>
<td>4.26</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>E Ocean View 26</td>
<td>Sherwood</td>
<td>02</td>
<td>4.18</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Granby</td>
<td>E Ocean View</td>
<td>02</td>
<td>4.92</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Granby</td>
<td>Balview</td>
<td>02</td>
<td>5.04</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Granby</td>
<td>Searsview</td>
<td>02</td>
<td>5.08</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Granby</td>
<td>Vidin</td>
<td>02</td>
<td>5.18</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Granby</td>
<td>Government</td>
<td>02</td>
<td>5.21</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Granby</td>
<td>Dupree</td>
<td>02</td>
<td>5.25</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Granby</td>
<td>Cherry</td>
<td>02</td>
<td>5.31</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Granby</td>
<td>Osborne</td>
<td>02</td>
<td>5.33</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Granby</td>
<td>Tidewater Drive</td>
<td>02</td>
<td>5.51</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Granby</td>
<td>Sloane</td>
<td>02</td>
<td>5.60</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Granby</td>
<td>Gilpin</td>
<td>02</td>
<td>5.66</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Granby</td>
<td>Lorengo</td>
<td>02</td>
<td>5.70</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Granby</td>
<td>Randall</td>
<td>02</td>
<td>5.72</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Granby</td>
<td>Leicester</td>
<td>02</td>
<td>5.80</td>
<td>.00</td>
<td></td>
</tr>
</tbody>
</table>

---

Figure 15. Sample Page of Brown Sheet Route File (TRT, Norfolk).

ADAS PROJECT AT TRT NORFOLK, VA. 95
can be created in the format needed from this file (Brown Sheet Route File). To do this, an input software has been developed. This software is modular in nature. It consists of the following input modules:

1. Receiving Module 1
2. Names Module
3. Distances Module

**Input Modules from the Brown Sheet Route File:**

The modules named above are the input modules from the Brown Sheet Route File. This section describes each of these modules in detail:

1. Receiving Module: This module receives the concerned or required Brown Sheet Route File and stores them as created. Here, the required parameters are recognized and they include street names at intersection, route number, direction and mileage post. These shall be used in the Names and Distance reference file creation. Thus, the receiving module inputs the Brown Sheet Route file for the route under consideration.

2. Names Module: In this module, the requirements as far as the creation of Names reference file for the route is concerned, are satisfied. These include receiving the following information and storing them in the required format:

   - Route Number.
   - Route Name.
   - Number of stops on the route for a complete loop. This is the sum of the number of stops from a starting point and back to it. This includes a total of the number of stops on both 'inbound' and 'outbound' trips.
   - Names of all the stops.
3. Distance Module: In this module, the requirements as far as the creation of Distance reference file for the route is concerned are satisfied. Here, the milepost values obtained from the brown sheet route file are cumulative distance in miles. Hence, these values have to be changed to give distances between two consecutive stops and also these distances have to be converted to 'odometer clicks' measure. The following are the relevant details that are incorporated:

- Route Number.
- Route Name.
- Total number of stops on the route.
- Turnaround point (Stop number at which the outbound trip ends).
- Stop numbers.
- Distances in odometer clicks.
2. Headway Sheet System:

This system gives all the information about the headway design at TRT, Norfolk. Here, the time headway between different buses on the same route are given. The various files that are laid out on the Headway Sheet System are:

1. Headway City File: It contains a one numbered code for each city in the headway system.

2. Headway Scheduled Timepoints File: It contains all scheduled timepoints for the routes in the headway system.

3. Headway Route File: It contains all inbound and outbound Scheduled Timepoints (STPs) for a route. It also contains miscellaneous route information like route number, description and so on.

4. Headway Active, Test and History Files: The Headway Active (Headact) Test (Headtest) and History (Headhist) files all have the same file layout and contain the same data. The Headact file contains active headway sheets for the route; the Headtest file contains headway sheets under test; and the Headhist file contains inactive or deleted headway sheets. These files contain an entire headway sheet, with all STPs, timepoints and mileage information. The files contain two different types of records namely Header Record and Detail Record. There is one header record for each route and multiple detail records for each trip on the route.

5. Headway Note File: The Headway Note File (Headnote) contains up to ten lines of driver instructions for the headway route.

An examination of the various files on the Headway System reveals that the information needed for the creation of the schedule (block) reference file include:
• Block Number
• Number of loops of operation for the day.
• Number of timepoints on the route per loop (Inbound + Outbound STPs).
• Starting time of operation of the route (last loop), in time clicks where 1 time click = 15 seconds.
• Stop numbers of all the timepoints on the route.
• Time (Arrival time) at all the timepoints for the route on the particular block for the complete day in 15 second time clicks.

It has to be realized here that all of these are not directly available. For example, number of timepoints on the route is the sum of Inbound STPs and Outbound STPs. All the times at the timepoints are converted to 15 second time clicks. Finally, all the details mentioned above are received or calculated in the receiving module 2, as the case may be, and stored in the required format.

Summarizing, the following three modules create the required reference files:

• Names Module
• Distance Module
• Schedule Module

The first two modules are formed from the Brown Sheet Route File while the Schedule Module is formed from the Headway Active File. In all, the interfacing input software will have the following modules with their respective functions:

1. Receiving Module 1: This module receives the concerned Brown Sheet Route File and identifies the key variables involved.
2. Names Module: This module creates a names reference file for the particular block on the route in the required format by taking the names from the Receiving Module 1 and stores the file in the drive specified.

3. Distance Module: This module creates a distance reference file for the particular block on the route in the required format by taking the distances from the Receiving Module 1, calculating the distances between stops and then converting the distances in miles to odometer clicks.

4. Receiving Module 2: This module receives the Headway Active File from the Headway System for the particular block on the route and then identifies the key variables involved.

5. Schedule Module: This module creates the schedule reference file in the required format by taking all the relevant data needed from the Receiving Module 2 and then stores the same in the drive specified.
Algorithm for Interfacing Input Software

1. Get information on the route under consideration i.e. Route Number and Block Number on which the bus with the ADAS equipment is operating.

2. Retrieve the Brown Sheet Route File that is desired from the Brown Sheet System.

3. Store the key variables namely the stop names and the milepost value. Note the total number of stops = number of inbound + number of outbound stops.

4. Calculate the distance between stops by subtracting cumulative distance of stop (n-1) from that of stop n.

5. Convert the distance into odometer clicks.

6. Note the total number of stops, stop number at turnaround point.

7. Write the stop names in the required format into a file and save as NamesXX.Ref where XX is the route number, in the drive specified in Systinfo.Ext.

8. Write the Distances in odometer clicks in the required format into a file and save the file as DistXX.Ref where XX is again the route number, in the drive specified in Systinfo.ext.

9. Retrieve the concerned Headway Active File from the Headway System.

10. Prepare the key variables namely number of inbound STPs, number of outbound STPs, Run number, Block number, Miles to stop 1, Arrival time.

11. Write the details into a file in the required format.
12. Create the Schedule Reference file as described and store in the drive specified in Systinfo.Ext.
7.4.2 Interfacing Output Software

This software has been designed to change the format of the output file (processed file) formed by the ADASs APCDATA program to the format of UMTA TRS80 Data File. It is built modularly. The UMTA TRS80 Data File (UMTA.DO) is the file to be loaded into the Datapoint system. There are four types of records namely: Header Record, Trip Record, Detail Record and Fare Record. There is one Header Record that identifies the trip to be uploaded; there is a trip record for the start of each direction of the sample trip to be loaded; the detail record contains passenger count information; the fare record contains the ending farebox reading for the sampled trip. This data is uploaded into the UMTADAY file.

The header record is 23 characters long, the trip record is 40 characters long and the detail record is 15 characters long.

The output file created by the ADAS software gives the following details:

- General information concerning the route like Route number, Route name, Number of Stops, Trip Distance, Loop Distance, Turnaround Point, Number of activity records, Block number, Date, Day of Week & Bus ID.

- Specific activity information like activity type, stop number at which the particular activity occurred, the distance, the time of arrival or occurrence, scheduled time, number of passenger ons, number of passenger offs, passenger load and the match value allocated to the specific activity record. This information is procured for all activity records.

The format of the UMTA TRS80 Data file is shown in the figure in the next page. It can be observed that all the information required for the Header Record and the Detail Records are directly available from the output (processed) file of APCDATA program. However, certain Trip Record information will have to be given or deduced. This information includes Trip type (In or Out), Driver employee number, Observer employee number and weather.
**HEADER RECORD:**

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>SIZE</th>
<th>LOCATION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>A</td>
<td>1</td>
<td>1</td>
<td>&quot;H&quot;</td>
</tr>
<tr>
<td>DATE</td>
<td>A</td>
<td>8</td>
<td>2-9</td>
<td>Date in MM/DD/YY format</td>
</tr>
<tr>
<td>DAY</td>
<td>A</td>
<td>3</td>
<td>10-12</td>
<td>Day</td>
</tr>
<tr>
<td>ROUTE</td>
<td>A</td>
<td>4</td>
<td>13-16</td>
<td>Route number</td>
</tr>
<tr>
<td>BLOCK</td>
<td>A</td>
<td>7</td>
<td>17-23</td>
<td>Block number</td>
</tr>
</tbody>
</table>

**TOTAL HEADER RECORD LENGTH:** 23 characters

<table>
<thead>
<tr>
<th>FIELD NO.</th>
<th>DATA ELEMENTS</th>
<th>FIELD SIZE</th>
<th>RECORD SIZE</th>
<th>SYSTEM NAME</th>
<th>KEY</th>
<th>CONTENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Record Type</td>
<td>1</td>
<td>1</td>
<td>RT$</td>
<td></td>
<td>T-trip Record</td>
</tr>
<tr>
<td>2</td>
<td>Type of Trip</td>
<td>3</td>
<td>2-4</td>
<td>TD$</td>
<td></td>
<td>Out or In</td>
</tr>
<tr>
<td>3</td>
<td>Driver Emp.#</td>
<td>4</td>
<td>5-8</td>
<td>DR$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Observer Emp.</td>
<td>4</td>
<td>9-12</td>
<td>OS$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Weather</td>
<td>7</td>
<td>13-19</td>
<td>WE$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Time Trip</td>
<td>8</td>
<td>20-27</td>
<td>TIMES$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Passenger on Board</td>
<td>3</td>
<td>28-30</td>
<td>OBZ</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DETAIL RECORD:**

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>SIZE</th>
<th>LOCATION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>A</td>
<td>1</td>
<td>1</td>
<td>&quot;D&quot;</td>
</tr>
<tr>
<td>PASSON</td>
<td>N</td>
<td>3</td>
<td>2-4</td>
<td>Number of passengers getting on</td>
</tr>
<tr>
<td>ARRIVE</td>
<td>A</td>
<td>8</td>
<td>5-12</td>
<td>Arrival time (HH:MM:SS)</td>
</tr>
<tr>
<td>PASSOFF</td>
<td>N</td>
<td>3</td>
<td>13-15</td>
<td>Number of passengers getting off</td>
</tr>
</tbody>
</table>

**TOTAL DETAIL RECORD LENGTH:** 15 characters

Figure 16. Format of the UMTA TRS 80 Data File (TRT, Norfolk)
Thus, the interfacing output software will have the following modular structure:

- **Receiving Module 1**: This module receives all the information from the output file, created and stored at the end of the data processing stage in using the ADAS software.

- **File Creation Module**: In this module, the following steps are executed: The header record is created. Here, Date, Day, Route number and Block number are used. Next, the Trip record is created for every trip (Inbound and Outbound). Driver Employee number, Observer Employee number and weather information are all obtained from the user. Time at which the bus started the trips and passengers on board are also noted. Then, Detail Records are created for all stops where activities occurred. Here, the number of passengers getting on, Arrival time and number of passengers getting off are all addressed.

- **Receiving Module 2**: In this module, Detail Records at stops where no activity occurred are created and inserted between those where it occurred. Before that, the brown sheet route file is retrieved to ascertain all the stops and then detail records with zeroes in all the fields showing activity are inserted.

- **Storage Module**: All these are then stored in a file named as needed. This file is then uploaded into the UMTADAY file.
Algorithm for Interfacing Output Software:

1. Retrieve the output file which has to be converted to UMTA TRS80 Data file.

2. Create header record in the required format.

3. Create trip record for every trip of type (In or Out) in the required format.

4. Create detail records at stops where activities occurred, in the required format.

5. Retrieve the brown sheet route file for the route.

6. Assess all the stops where activity did not occur.

7. Insert detail records for those stops in the required format at the location.

8. Store in a file named the same way as UMTA TRS80 Data file.
7.5 Implementation of the System

An incremental approach to the implementation of the ADA system was followed. This section describes the actual installation of the equipment in place and a specific view of the implementation of the system.

7.5.1 Actual Installation of the System:

The four modules were installed in the bus at the back of the front seat, away from the sight of any passengers. The two pairs of infra-red sensors were fixed to specific fixtures which had to be made up of thick steel plates and fixed in place. These sensors were fixed at 48” height at both front and rear doors and they were connected to the corresponding modules. The on/off signal was installed in place and so was the transmission pick up. The wiring and other electrical connections and the installation as well as testing of the equipment was done by Dr. Charles Nunnaly of the Electrical Engineering Department. It is to be noted that the sensors get activated only when the doors are open.

7.5.2 Summary of the ADAS Implementation and Application

This section describes the implementation and application of the ADA system along with their corresponding functions.

- The ADAS hardware on the bus collects data for a particular route and returns to the garage.
• This data file is dumped using communications software to the storage device on the ADAS processing computer (Portable or Lap Top Computer).

• Reference files namely Names, Dist and Schedule are created using interfacing input software and the brown sheet route file and headway active file for the route.

• The raw data file is processed using APCDATA program and the reference files created above.

• The output file is created and stored.

• UMTA TRS80 Data file for the route is next created using the interfacing output software on the output file and then stored on the five and a quarter inch diskette.

• This file is then transferred to the Datapoint mainframe through the Datapoint PC and further processed to yield management reports and plots.

• Also, the output file created using the data processing software is further processed using Management Reports and Plots in the ADA system software developed at Virginia Tech into specific reports and plots, and printed.
8.0 CONCLUSIONS AND RECOMMENDATIONS

8.1 Conclusions

Automatic Data Acquisition Systems are becoming more popular due to the stabilization of both hardware and software associated with it. Also, the advantages of ADAS, with practically unlimited information processing potential, are being realized by various small and large transit properties. While there have been certain problems with the systems, the general consensus is that with a better information dissemination, the maintenance and handling of the ADAS would become routine, alleviating some of the above problems. It is obvious that ADAS offer a wealth of information which would otherwise be very expensive and cumbersome to obtain by manual data collection methods. Experience with both the systems at Roanoke and Norfolk has led the author to firmly believe that ADAS could be invaluable in setting up a database which would help in better system monitoring and performance through wiser and more informed decisions.
Summarizing, the following conclusions can be drawn from this work:

1. An easy to implement, cheap and accurate ADAS is now available.

2. The ADAS has proven software capabilities which remove the need for radio signposts used for establishing locational accuracy. This means the removal of installation and maintenance costs and worries as regards radio signposts. This is a clear advantage over most other systems using them.

3. The software is modular in nature, allowing for further development. The potential for software development in this area is tremendous and this algorithm promotes the flexibility.

4. The software is interactive and menu driven. This gives the user a chance to understand and visualize the sequence of the matching process.

5. The output file format can be easily changed depending upon the requirements of the transit company. This is possible because of the structured approach employed in the algorithm.

6. The software can be run on any IBMPC or Compatible which is generally available as a part of the Office Automation System in any transit company. Thus, this renders the ADAS to simply fit in without any extra investment for data processing machines.

7. With the present ADAS, transit agencies can now establish a comprehensive database to monitor various routes and schedules. This in turn, would allow managers to exercise better system surveillance and hence make better decisions.

8. A great potential for developing statistically valid demand models for each of the routes is created and this again would help in better system surveillance.
9. Advancement of the state-of-the-art of transit evaluation methods is bound to occur with the stabilization and improvement of both the hardware and software technologies of ADAS and the present experimental implementations of this ADAS will yield a wealth of insight into this.

10. Finally, with the versatile software package, put together with the hardware, certain very useful management reports and plots can be generated to aid the decision makers in both better planning and operation of routes.
8.2 Recommendations

There are several areas in which further work can be done. Firstly, the accuracy of the equipment installed in terms of passenger counting can be ascertained by having a "truth team" collect data simultaneously and then comparing the two. Next, the software development area has tremendous potential. Several other programs can be written and added to highlight the system performance.

Also, for passenger activity sensing, infra-red sensors are being used. These sensors are not error-free, obviously. To increase the reliability of the process of passenger movement sensing, a combination of different types of sensors could be investigated. This itself could form the content of a wholly new project. Hence, there is a lot of work that needs to be done in this area.

For various values of DeltaS1, the program should be run on several real life data files, to assess the best value for DeltaS1. This is the most important parameter in the matching process and hence a value which will allow for maximum number of activities to be matched without sacrificing accuracy should be assessed.

Finally, development of demand models for the various routes has now been made possible through the various data files that can be collected using the ADA system. These demand models are very useful in assessing the route performance and the changes in the route's performance with changes in service. The demand models could involve the following variables namely travel time, distance between stops, speed and frequency of operation and number of passengers on board. These variables can be calculated from the data using a simple procedure. Thus, a simple program can be written to generate these data from the raw data collected, and then these values can be used to build the model. This, however, demands more investigation and can very well become the content of another whole thesis.
References


12. Documentation of Hardware Componants - Red Pine Instruments Inc.

13. Communications with Mr Jeff Becker of TRT, Norfolk and Ms, Doreen of PSC Computer Systems.


A.1 INTRODUCTION

A.1.1 WHAT IS AN ADA SYSTEM?

As the name suggests, an Automatic Data Acquisition System (ADA) is a composition of hardware and associated computer software which work together to yield certain basic reports, crucial in making certain decisions concerning operations, routes and schedules in Transit Bus Systems.

The hardware can be divided into the following -
a. Basic Hardware
b. Interfacing Hardware and
c. Office Automation System in use (Ex - IBM PC)

The Software can be divided into the following -

a. Operations Software and
b. Management Software.
A.1.2 APPLICATION AND USAGE OF ADA SYSTEMS

In the recent past, a number of transit operators have begun to use computers to improve the collection, analysis, presentation and distribution of management information including very useful performance indicators. One such company using the ADA System is the Valley Metro Transit Company at Roanoke, Va.

ADA systems, with considerable stabilisation of available hardware and associated software, are beginning to replace manual checkers used to collect passenger data. Apart from their evident superiority in data collection, in the long run, the ADA systems also work out more economical and efficient. The ADA systems allow for a continuos inflow of raw system data and when used with the associated software, they prove to be better and more reliable performance indicators.

It has also been indicated that if 5% of the fleet size is equipped with ADA systems, one would be able to get a statistically valid database of all trips made on the system. This could lead to the establishment of a database for the system which would be useful to assess daily, weekly or yearly performance characteristics of the system and also for comparisons to be made within the system.

In addition to critical performance indicators of the Transit System being highlighted, one can also use ADA Systems Software Technology to generate some mandatory reports to UMTA etc. and other reports and plots. We will look at them in detail later.

A typical ADA system, as in Roanoke Valley Metro, has excellent potential for development of additional integrated software especially towards Graphical Information Support. This is another great advantage of the existing system.
A.1.3 HOW TO USE THIS MANUAL

This manual deals, in detail, with the Operation, Maintenance and Care of the ADA system built at Virginia Tech, Blacksburg, Va.

- The first section addresses, in general, what ADA Systems are and what they do.

- The second section addresses the Application and Usage of ADA systems. This section is useful to all those who have just started using the system to give them a broad view of the features of the system. This section also gives the user, the directions to using this manual.

- The third section addresses the various hardware features of the ADA System including a discussion on the components used, layout of the components, installation and testing of the system, maintenance, inspection and care of the system, possible problems and the precautions from the hardware point of view. An idea as to the handling of the equipment is also given.

- The fourth section addresses the software features of the system including available software, their applications, their usage, some hints on efficient storage of data and their handling and some of the problems and their alleviation. A lot more could be written on the problems section but the authors would like to receive some feedback from the users regarding this. Any suggestions, additions etc. would be welcome.

- The fifth section addresses the features of the interfacing system including the various components (devices) involved, the process of data dumping and the process of data transfer.

- Finally, the sixth section gives the ‘Operations in a Nutshell’.
Depending on the operation being undertaken like data dumping, data transfer, data processing, report generation, plot generation, maintenance, care and handling of the equipment etc., the user should look at the contents sheet to ascertain under which section the particular operation falls and then go from there to the particular section and then to the subsection for guidelines or for other information. This way the user is led through the various operations smoothly.
A.2 FEATURES OF THE SYSTEM (HARDWARE):

This section is divided into the following five subsections namely:

1. Hardware Components.
2. Lay Out of the Components.
3. Installation & Testing of the System.
5. Possible Problems - Precautionary Measures.

A.2.1 HARDWARE COMPONENTS

BASIC HARDWARE COMPONENTS.

The various basic hardware components involved are:

The Control Storage Module (CSM) or Data Storage Module (DSM).
The Passenger Count Module (PCM).
The Odometer / Power Module (OPM).
Three pairs of infra-red sensors - (2 in front and 1 rear).
The Diagnostics Module (DGM).
A.2.2 INTERFACING HARDWARE COMPONENTS.

The interfacing hardware consists of a lap held microcomputer with optional equipment like a cassette player and cassettes or a 2 disk drive so that by using a compatible software, with the right protocol, the data can be dumped from the CSM to the secondary storage units. An RS-232C serial port interface, would be very useful, on the microcomputer. Ex. An IBM Laptop Computer or Data General's DG/ONE.

A.2.3 OFFICE AUTOMATION SYSTEM COMPONENTS

The Office Automation System (OAS) involves a Computer or a Microcomputer (like an IBMPC). Color Monitor with Standard Resolution Graphics capability including the Graphics Card and a Graphics Printer.

A.2.4 LAY OUT OF THE COMPONENTS:

The diagram in the next page gives the Lay Out of the System Components.
1. DATA ACQUISITION
   - COUNTING SENSORS
   - SIGNPOST SIGNAL (Optional)

2. DATA RECORDING & STORAGE
   - LOGIC ALGORITHM
   - ODOMETER + CLOCK
   - DATA STORAGE

3. DATA TRANSFER
   - CENTRAL COMPUTER FACILITY
     - APC DATA PROCESSING
     - ADDITIONAL SYSTEM DATA (Schedule Times + Distances)

4. REPORTING & ANALYSIS
   - REPORTS
A.2.5 INSTALLATION AND TESTING OF THE SYSTEM:

INSTALLATION OF THE SYSTEM:

The system components are installed on the bus according to the documentation provided by the manufacturers and as detailed. [25].

TESTING OF THE INSTALLED EQUIPMENT:

Once the equipment including all the modules, sensors etc., are all fixed in place as shown in the photographs and diagrams and the power supply is ready, the diagnostic unit is hooked up to the DSM on the bus for testing the various modules. The registration of proper 'logs' and their storage are checked. The infra-red sensors should be properly aligned. When the beams are broken in a particular sequence, the activity recorded should show it. In other words, the 'built-in algorithm' is checked to indicate the passenger ons/off s (activity) at both the front and rear doors.

The following are the steps for 'Testing the Installed Equipment':

1. Turn on the ADA system by switching on the ON/OFF switch on the island.

2. Hook up the DGM to the other modules.

3. Test each of the modules by looking at the menu (tree structure) shown in the next page.

4. Check and verify passenger ons and offs in front and rear doors. Also, check beam and door status. All this is done by checking
5. Check and verify distances from the Odometer Power Module.

6. Check and verify the storage of logs from DSM (Log Area).

7. Check and verify the creation of each of the logs.

8. Check and verify the creation of log 11 when signboard is changed.

9. Check and verify the creation of manually input logs.

10. Finally, check and verify the working of the back-up battery.
A.2.6 MAINTENANCE, HANDLING AND CARE OF THE SYSTEM:

Normal maintenance procedures, with care to see that only authorized, qualified personnel handle the system should be exercised. Care should be taken to see that none of the wires nor the sensor units get damaged from vandalism or other reasons. A good approach would be to have all these wires and sensor units enclosed and concealed. Careful handling of the wires and the portable diagnostic module should be exercised. Also, care should be taken not to allow unauthorized personnel handle the switch on the island, cables and connecting wires between modules. Any damage to any cables or connecting wires should be immediately reported to the concerned authorities.
A.2.7 PROBLEMS - PRECAUTIONARY MEASURES.

Major number of problems in the hardware arise in the cable connections or connectors. Very careful handling of these components is highly recommended.

A connector connects the communications cable from the lap top to the DSM on the bus. It is recommended that the cable end of the lap top be not disconnected after each transfer or dump. This will allow that end of the connector cable and the outlet on the laptop remain in tact for a longer period of time.

A.3 FEATURES OF THE SYSTEM {SOFTWARE}:

AVAILABLE SOFTWARE:

The following is the available software in the usual order of usage -

a. OPERATIONS SOFTWARE.
b. MANAGEMENT SOFTWARE.

A.3.1 OPERATIONS SOFTWARE [COMMUNICATIONS SOFTWARE]:

Once the data has been collected throughout the time period, the data is dumped to auxiliary storage (like the 3.5 inch disk on the DG/ONE computer) as in Roanoke. The person
in charge of the 'data dumping' process turns on the System at the beginning of the day before data collection and then after the data has been dumped at the end of the time period, he resets the CSM module by turning the key on the 'island' off and on. For the complete 'data dumping process' refer to section 'Usage - How to use the Software', under Operations Software.
A.3.2 MANAGEMENT SOFTWARE:

The following are the management software available:

1. ADADATA
   This is used for data processing. The raw datafile has to be first of all processed.

2. MAKEFILE
   This is used for creating ‘Reference Files’. While performing data processing and report generation, the reference files are used for ‘Matching’ and ‘Creating’ activity records.

3. REPORTS
   After data processing, this is used to generate reports. The reports that can be generated include Section 15 daily/annual and Time Point Profile & Summary reports.

4. PLOTS
   After data processing, this is used to generate graphs. The graphs include Passenger Load Plot, Route Demand Plot, Route Evaluation Plot and Schedule Evaluation Plot.

For a step by step instruction on using the above software, see ‘Usage - How to use the software’ section of this manual.
A.3.3 APPLICATIONS:

OPERATIONS SOFTWARE:

This is a communications package to be used with the particular alternative of BASIC namely GWBASIC as in Roanoke Valley Metro. The package at Norfolk is a compiled program written in Basica. This program provides the protocol and sets up the "handshake" between the CSM on the bus and the secondary storage device (lap held microcomputer). The instructions to operating this software is given in section 'Usage - How to use the software'. It is to be noted here that depending on the secondary storage device used, the operations software may vary. The operations software discussed here is the one used by 'Roanoke Valley Metro Co' at Roanoke, VA.

MANAGEMENT SOFTWARE:

The instructions for using this software is given in 'Usage - How to use the software' section.

ADADATA: This program disk offers the following options:

- PROCESS RAW DATA FILE.
- GENERATE BUS STOP LISTING.
- EXIT TO DOS.

MAKEFILE: In order to identify the stops at which the various activities occur, it is necessary to reference the activities to files which contain the names of all of the stops on the routes and the distances between them. A reference file is also necessary in order to determine the schedule deviations at the timepoints. In order to create the following reference files, the

Appendix A. AUTOMATIC DATA ACQUISITION SYSTEM (User's Manual) 129
‘Makefile’ program is used. The data required for these files are to be collected in the manner described in Section 3.3(b). The following options are available on the ‘MAKEFILE’ program:

CREATE OR EDIT STOP DISTANCE FILE.
CREATE OR EDIT STOP NAMES FILE.
CREATE OR EDIT SCHEDULE (BLOCK) REFERENCE FILE.
QUIT (EXIT TO DOS).

REPORTS:
This program is used, after the raw datafile has been processed, for generating certain very useful reports. The following options are available:

GENERATE TIMEPOINT PROFILE REPORT.
CREATE SECTION15 DAILY REPORT FILE.
UPDATE SECTION15 ANNUAL REPORT FILE.
GENERATE SUMMARY REPORT FILE.
QUIT (EXIT TO DOS).

PLOTS:
From the ADA output files, there are 4 different plots that can be created using the ‘PLOTS’ program. This should be provide some graphical information support to the management for making certain decisions.

The following options are available on the ‘PLOTS’ & ‘PLOTS1’ program:
PASSENGER LOAD PLOT.
ROUTE DEMAND PLOT.
ROUTE EVALUATION PLOT.
SCHEDULE EVALUATION PLOT.
QUIT (EXIT TO DOS).
A.3.4 USAGE : HOW TO USE THE SOFTWARE:

OPERATIONS SOFTWARE: [DATA DUMPING]

o The process described here, is the one for the system at Valley Metro at Roanoke, VA.

o Plug the ‘bus’ end of the communications chord into the large round phono jack behind the first seat on the right side of the bus. Plug the ‘computer’ end into a similar jack on the left side of the Data General computer. Check to make sure that the ADA DATADUMP.BAS PROGRAM diskette is in the front drive and the data storage diskette is in the rear drive. However, if the device used to dump and transfer the data is a different one, the procedure involved may be a little different.

o Turn on the computer. Make sure that no error messages appear. Several messages are displayed and then the message “OK” appears. If any number (Ex: ‘The following tests have failed’ - ‘11’) appears, then the computer error has occurred. Look up the manual for the computer and take the action directed by that manual.

o Press the key marked “F3”. The word ‘load’..’ will appear. Type ‘DUMP’ and press return. OK message will appear.

o Press the key marked “F2”. The program will run and a menu will appear on the screen. Type “T” and press return to transfer the data.

o A message reading: “S-Screen only, D-Disk only, B-Both. Enter Choice.” will appear on the screen. Press “D” and strike return.
The computer will ask for the "OUTPUT FILE NAME". Give the name as follows: Type "B" followed by the number of the block on which the route was operating that day and then the first three letters of the month name followed by the day of the month. For example, if the bus was operating on block # 10 on March 21st, then the output file name would be "B10MAR21".

The computer will ask for the "24 HOUR REAL TIME HOUR". Enter the current hour and press return. (If the hour is past noon, add 12 to the current hour. For example, 6 p.m. would be entered as 18). Next, the computer will ask for the "24 HOUR REAL TIME MINUTES". Enter the number and press return. The computer will dump the data to the storage disk displaying a flashing code indicating the amount of data stored. The process may take a couple of minutes.

When the data has been dumped, the computer will ask for the month. Enter the number of the month and press return. Do the same for the day. For the year, enter only the last two digits and press return.

Next, the computer will ask for the "ROUTE ID". Enter the number of the route on which the bus was operating that day and press return.

The computer will ask for the day of the week. Enter the appropriate number and press return.

Finally, the computer will ask for the block number for the bus for that day. Enter the number and press return.

The computer will then display two sets of numbers that echo the information which had just been entered. The numbers will be in the form given in the next page.
<table>
<thead>
<tr>
<th>12</th>
<th>MONTH</th>
<th>DAY</th>
<th>YEAR</th>
<th>ROUTE ID.</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>DAYOFWK</td>
<td>BLOCK</td>
<td>HOUR</td>
<td>MINUTE.</td>
</tr>
<tr>
<td>12</td>
<td>BUS ID</td>
<td>0</td>
<td>0</td>
<td>0.</td>
</tr>
</tbody>
</table>

- Check to make sure these numbers are correct. If any are incorrect, make a note of which ones they are and submit it with data diskette.

- The data dumping process is now complete. Turn off the computer and disconnect the communications chord. Insert the key into the power on/off switch and turn it to the left until the power on indicator light goes off. Then turn it back to the right until the light comes back on and remove the key. The computer is reset and ready to collect the next day's data.
DATA TRANSFER

The process described here is the one for the Valley Metro at Roanoke, VA. Since the data is retrieved on 3-1/2 inch diskettes but processing is done on 5-1/4 inch disk drive, a transfer step is required. It is to be noted that the data may be retrieved on other secondary storage devices and that the 'Data Transfer' step would be different then. However, the transfer for the former case (3-1/2 to 5-1/4 inch diskettes) is accomplished by following these simple steps:

1. Plug in the auxiliary disk drive to the back of the computer and also connect it to the power supply. Insert the data disk (containing the raw data retrieved using the DG/ONE) into the auxiliary drive and turn it on using the red switch at the back.

2. Insert the 'APCPROGRAM DISK' (with the 'Manzana' software in its directory) into the A: drive of the main data processing computer (i.e. the Office Automation System in use) and turn on the computer.

3. When the A> prompt appears, strike D: and return. The auxiliary drive designated as D> drive is now the current drive. It operates exactly like the other drives. Data can be transferred using the DOS "copy" command. For example, "copy B1OMAR21.DAT C: " copies the file to the hard disk C: drive.

This completes the DATA DUMPING and DATA TRANSFER. The data file with the raw data is now available for processing.
MANAGEMENT SOFTWARE:

CREATING REFERENCE FILES

The process described here is the one used for the Valley Metro at Roanoke, VA. In order to identify the stops at which the various activities occur, it is necessary to reference the activities to the names of all of the stops on the routes and the distances between them. This is done using the reference files. A reference file is also necessary in order to determine the schedule deviations at the timepoints. This section describes the data required for these files, along with some techniques for obtaining them. Then it describes MAKEFILE, the program which enables the user to easily create the necessary reference files. Note here that the reference files for a particular route need to be created only once and once they are stored on the processing computer, they can be continually used till changes to the bus route or schedule are made.

Collecting Data for the Stop Distance and Stop Names Reference Files

Collecting the data for the distance file requires two people: a bus driver and someone familiar with the route who will record the data. They should take the equipped bus, the diagnostic unit, and the stop distance and names forms (there is a sample at the back of this guide) and perform the following steps:
1. Drive the bus to the beginning of the route (one of the terminal points, preferably the stop at which the drivers change the destination sign at the beginning of a one-way trip).

2. When the bus reaches that point, the recorder should reset the distance registers of the onboard computer by turning the ADA power on/off switch to the left until the red light goes out. Then he should turn the power back on and connect the diagnostic unit and press the “2” key.

3. Next, record the stop number (in this case 1), a description or name which identifies the stop, and the distance reading from the diagnostic unit (which is equal to or close to zero at this point).

4. The driver then drives to the next stop, where the recorder again writes the number, name, and distance. The process is repeated until the bus returns to the starting point, where the final record is made.

5. If a stop is a “timepoint” (a stop at which a scheduled time is listed in the route schedule), the recorder marks a “T” to the left of the stop number.

6. Finally, the recorder notes the total number of stops, along with the stop number of the turn around point at the far end of the route, where the bus begins the return trip to the starting point. This stop is the “turnpoint”.

Appendix A. AUTOMATIC DATA ACQUISITION SYSTEM (User’s Manual)
Collecting Data for the Schedule (Block) Reference File.

The data for the schedule reference file is very easy to obtain. In fact it is all contained in the schedule sheets which the dispatcher keeps for each route. There will be a reference file for each block. This file will identify the numbers and names of the timepoints, and will record the scheduled time for each of these timepoints for every loop the bus makes during the day. In order to simplify the creation of the file, the following steps should be taken:

1. Make a copy of the schedule sheet for each route.

2. Read across the top of the columns until the name of the beginning of the loop (as determined in the collection of the stop information) is located. Mark the number “1” next to the name. This will be the first timepoint.

3. Move to the next column to the right; if it is identical to the column just numbered, then mark out the whole column; if it is different, then give it number “2”.

4. Continue numbering the timepoints as above. When the last column is reached, continue with the first column on the left until all columns are either numbered or marked out.

5. Next determine the number of loops. This is done by locating the column for the last timepoint, and then counting the number of times the number of the block for this file is encountered while reading down the column. If there are any scheduled stops beyond the last one in that column, then add one to the count. This final number is the number of loops. The sample shows a schedule sheet which has been prepared.

6. Finally, determine the stop number of each timepoint from the stop names and distance data sheets for that route.
Once this information is obtained, the file can be created using the Schedule (Block) option of the MAKEFILE program. The MAKEFILE program is next described.
MAKEFILE:

To run MAKEFILE, place the ADA program disk in the A: drive and turn on the computer. If the computer is already on, press Ctrl-Alt-Del keys together to reset. When the A> prompt appears, type MAKEFILE. A menu will appear on the screen asking the user to choose one of the four options namely:

1. Create or Edit Stop Distance File.
2. Create or Edit Stop Names File.
3. Create or Edit Schedule (Block) Reference File.
4. Quit (Exit to DOS).

Create or Edit Stop Distance File:

The user chooses to either edit or create a new file by typing either an "e" or a "c". If the create option is chosen, the program will ask a series of questions which the user should answer using the information from the stop distance forms. Then the program asks for the cumulative distance to each stop. The user should simply type in the corresponding number from the distance column of the forms. When the last stop is completed, the user is given a chance to view the file and edit any incorrect entries. Finally, the program will give what the file name will be. This name will include the drive to which the file will be sent, followed by "DIST", the route number, and the extension ".REF". The user should accept this name unless the file is for a temporary route, i.e. one with a detour or other irregularity. In that case, another name may be desired to distinguish the file from the regular one. If the edit option is
chosen, the user is asked which file to retrieve. Then it displays the contents and from that point on it functions just like the create option.

*Create or Edit Stop Names File:*

This option is very similar to the Stop Distance option, except that it asks for the stop names rather than the cumulative distance. The other difference is that the file name includes the drive, “NAMES”, the route number, and the extension “.REF”. All other functions are practically identical, and the discussion above applies to this option as well.

*Create or Edit Schedule (Block) Reference File:*

This option follows the same pattern as the previous two. After the information about the number of stops, the block number, and the number of timepoints are entered, the program will prompt the user for the stop number of each timepoint. Next it prompts the user for the scheduled time at each timepoint by loops. If there is no entry (often the first or last loop is incomplete), then enter 0,0. Otherwise, enter the hour, followed by a comma, and then the minutes. Again, use 24 hour time for the hour (e.g. 6:00 p.m. = 18 hours). When all the loops and timepoints are entered, the program allows the user to view and or edit the values. Finally, it displays the filename, consisting of the drive, “BLOCK”, block number, and the extension “.REF”. Unless the schedule is for a holiday or saturday, then accept the name given by the computer. For Saturday blocks, include the number “6” between “BLOCK” and the block number. Holiday or temporary schedules can be given any other name.
Quit (Exit to DOS):

This option terminates the program and returns to DOS. Type "3" to use this option when the main menu shows up on the screen.
DATA PROCESSING:

Once the reference files are created and the data file is transferred on to the main computer, the raw data can be processed. This is carried out by using the program ‘ADADATA’. A step by step procedure for running this program is as follows:

1. Type “ADADATA” when the DOS prompt appears after the computer is booted up using a DOS 2.1 or higher version. The following menu appears on the screen:
   (1) Process Raw Data File.
   (2) Generate Bus Stop Listing.
   (3) Quit (Exit to DOS).

2. Next, enter the number of your choice (i.e 1) and hit return to process the raw data file (remember the raw data file has to be processed before any report or plot can be generated).

3. Type the name of the raw data file you are processing(ex.”B10Mar21”) and hit return(to the query ‘Which data file do you want to process?’).

4. It says ‘Retrieving C: B10Mar21.Dat’ meaning that it is retrieving from the C> drive the data file ‘B10Mar21.Dat’. It then asks you if you want to view the raw data file. Type ‘N’ if you dont want to. Else type ‘Y’ and a screen of raw data is seen. If you want to view more, hit any key.
except 'Q' and another screen of raw data shows up. Else, type 'Q' to quit viewing the data and start processing again.

5. It says 'Creating activity records...'. Wait till it shows you the names of the reference files to be used in the data matching process. If they are the correct files, then type 'Y' and press return. Else type 'N' and press return. It will ask you for the names of the reference files and you type in the required files and press return.

6. It says 'Retrieving Reference Files' quickly and then the message 'Matching Activities to Stops - Please Wait' appears. If there are no errors in the processing, it says 'Matching Complete....' and then it returns to the main menu.

Note: The user needs to type in only the filename when the query 'Which data file do you want to process' shows up, while processing the raw data file. The drive and extension are added by the program. For example, entering "B10Mar21" will cause the data collected on Block #10 on March 21st to be processed.

This completes the data processing step. An output file has been created.
**GENERATE BUS STOP LISTING:**

1. Once the raw data file has been processed, the main menu of ADADATA Program appears.

2. Choose the number of selection ‘2’ (i.e. Type “2” and press return).

3. The question ‘For which data file do you want the bus stop listing?’ appears. Type the filename (Ex - B10Mar21) and press return.

4. The question ‘Do you prefer a screen display or a printed copy?’ appears. Depending upon your needs type “S” for screen display or “P” for a printed copy after getting the printer ready and press return.

5. The message ‘The reference files will be.... Are these the files you wish?’ If the correct reference files are shown, type “Y” and press return. Else type “N” and press return. It will ask you for the correct reference files. Type in the names of the correct reference files (Ex - C: Names1.ref).

6. Then the bus stop listing appears on the screen or gets printed depending upon your selection.

7. When screen display is selected, the first 25 lines of BUS STOP LISTING appears on the screen. If you want to quit viewing the data, type ‘Q’ and press return. Else, hit any other key to continue viewing the file.

**NOTE:** In this program, activity records including the type of activity (Ons, Offs, long idle, etc) and the time of day and the distance from the day’s starting point are created. This program requires a great number of steps and calculations and takes a couple of minutes to complete.
Also the user is given a chance to change the reference files if the data was collected on a special route or a modified schedule, but in a majority of cases, these files will be the appropriate ones.

QUIT (Exit to DOS):

If you want to return to DOS to generate reports or plots, type "3" and press return when the main menu appears.
GENERATION OF REPORTS:

From the processed data file (i.e. output file), several useful reports can be created. This is accomplished using the 'REPORTS' program.

REPORTS:

1. To enter the 'REPORTS' program, type "REPORTS" and press return.

2. A program menu appears offering the following options:

   a - Generate Time Point Profile Report.
   b - Produce Section15 Daily Record File.
   c - Update Section15 Annual Report File.
   d - Generate Summary Report File.
   e - Quit (Exit to DOS).

3. A query 'what is the number of your choice?' appears.

4. Type "1" and press return to generate 'Time Point Profile Report'.
5. A query 'For which data file do you want the report?' appears. Type the filename (Ex - B10Mar21) and press return.

6. The query 'Do you prefer a screen display or a printed output?' appears. Depending upon your needs, type "S" for screen display or "P" for a printed output after getting the printer ready. When a screen display is chosen, it says 'Retrieving Reference Files' and then 'The Reference Files are ...' 'Is this the file you wish?'. If correct, press "Y" and hit return. If the data was collected on a special route or on a changed schedule, then type "n" and type in the correct reference file name and press return.

7. The Time Point Profile Report appears on the screen or gets printed depending upon your selection.

8. To continue viewing the report on the screen (When screen display was selected), press any other key other than "Q". If you want to quit viewing the file, press "Q". It will take you back to the menu.
**PRODUCE SECTION15 DAILY RECORD FILE:**

1. When the menu appears, type "2" and press return.

2. The query 'For which output file do you want the report?' appears. Type the filename (Ex - B10Mar21) and press return.

3. It says 'Creating the report' and 'Storing the report' and then a series of messages appear asking the user if a screen display or a printed output is preferred.

4. Press return and the Section15 Daily Record tile appears on the screen and then if a printed output is desired, get your printer ready and then press "Shift-PrtSc" keys together. Else press return to get back to the menu.

**UPDATE SECTION15 ANNUAL REPORT FILE:**

1. When the menu appears, type "3" and press return.

2. The query 'Which Section15 Daily Record File do you want to add?' appears. Type in the filename (Ex - B10Mar21) and press return.

3. The daily file values will be added to the cumulative file and then you are given the option to print out the updated cumulative file. If the print option is chosen, the program asks whether or not to include annual totals. If the report is to be completed for the current fiscal year, then the annual totals are necessary. If not, they are unnecessary.
4. If the annual totals are included, the program asks for the total annual number of trips for each time period - a.m. peak (6 to 9); midday (9 to 3:30); p.m. peak (3:30 to 7:00); night (after 7 p.m.); Saturday (all day) and Sunday (all day). The report will then be printed after you have answered all the above questions. Upon completion, the program returns to the menu.

GENERATE SUMMARY REPORT FILE:

1. When the menu appears, type ‘4’ and press return.
2. The query ‘For which output file do you want the report?’ appears. Type the filename (Ex-B10MAR21) and press return.
3. An option to print the report on the screen or send the report to the printer appears. On choice, the names reference file is confirmed and the report will be printed on the screen or printer as the case may be.

QUIT (Exit to DOS):

To generate plots, you have to return to DOS and this is done now. Type “5” and hit return.
GENERATION OF PLOTS:

Many types of information are better presented in the form of a graph. In view of this fact, there are a couple of plots that are created from the output files using the PLOTS Program.

PLOTS AND PLOTS1:

To enter these programs, type "PLOTS" or "PLOTS1" upon A> prompt with the diskette containing these programs in A: drive. Menus offering the following options are presented.

a - Generate Passenger Load Plot.
b - Generate Route Demand Plot.
c - Generate Route Evaluation Plot.
d - Generate Schedule Evaluation Plot.
e - Quit (Exit to DOS).

Generate Passenger Load Plot:

1. Type "1" to generate the 'Passenger Load Plot'.

2. The query 'For which output file do you want the plot?' appears. Type the filename (Ex -B10Mar21) and press return.

3. The message 'Retrieving C: B10Mar21.out' appears and then a series of messages appear.
4. If you want a printed copy, then press return and after the screen display shows up, get the printer with graphics capability ready and press the ‘Shift - PrtSc’ keys together. Else type return to get back to the menu.

Note: Once the plot is complete, the user can get a large or small hardcopy by pressing the ‘Left Shift and PrtSc’ keys together or ‘Right Shift and PrtSc’ keys together, respectively. Sample Outputs are given in the next page.
Appendix A. AUTOMATIC DATA ACQUISITION SYSTEM (User's Manual) 153
Generate Route Demand Plot:

1. Type "2" and press return when the query at the end of the menu offering appears.

2. All the other questions are the same as the ones for the 'Passenger Load Plot' and type in
the same answers.

3. The route demand plot is a bar chart showing the number of passengers boarded during
each hour of the day.
Generate Route Evaluation Plot:

1. Type "1" at menu prompt from ‘PLOTS1’ program, to generate ‘Route Evaluation Plot’ and hit return.
2. The query ‘For which output file do you want the plot?’ appears and upon choice the concerned file is retrieved.
3. A plot of passenger load vs timepoints is displayed.
4. If you want a printed copy, then press the Shift Prt Sc key after getting the printer ready.

Generate Schedule Evaluation Plot

1. Type "2" from “PLOTS1” program menu to generate Schedule Evaluation Plot and press return.
2. The Query ‘For which output file do you want the plot?’ appears and upon choice, the concerned file is retrieved.
3. A plot of schedule deviations vs timepoints is displayed.
4. If you want a printed copy, then press Shift Prt-Sc keys together after getting the printer ready.

QUIT (Exit to DOS):

Once the plots have been generated, you can quit the program by typing "3" when the menu appears. This takes you back to DOS.
A.3.5 USEFUL HINTS ON EFFICIENT STORAGE:

Once the raw data file has been dumped and transferred on to the hard disk of the IBMPC(OAS), make sure that the raw data file and reference files namely Names, Dist and Block(Schedule) are all on the C: drive. Then, one can run the management software and generate reports and plots.

After the daily record files are created like Section15 reports and plots, the annual Section15 report(Cumulative) is updated, if needed.

A typical data file for the Roanoke Valley Metro consisted of 5000 to 15000 bytes of data on a normal day on the normal bus route, at an average. This can be determined for the particular route after a few days of data collection for every new transit agency implementing this system. This means that a typical formatted double sided double density diskette can hold up to a month’s data.

It is better to index the data files and the indexing can be made sequential so that it is easy to retrieve at a later date.

A step by step procedure to maintain data files and index them is given in the following section in the next page.
STEP BY STEP PROCEDURE TO STORE AND INDEX DATA FILES:

1. First of all, the double sided double density diskette (5-1/4") is formatted. This needs to be done only once, i.e. in the beginning of every diskette.

2. After several data files have been stored on the diskette and there is no more space left for any more files, it is labelled X1 - Xn, where X1 to Xn are the dates of the data files, i.e. Label the diskette 'DATAFILES X1 - Xn'.

3. Store the diskette away from any device that can create a magnetic field. X1 and Xn are the numbers of the data files (first and last) on the diskette.

4. Do not touch the diskettes at the places where they are read, since 'static' can also cause damage to the data stored.

5. At the end of each year, the cumulative updated section15 annual record is stored on a separate diskette indexing them in a way that can be recognised at a later date.
A.3.6 PROBLEMS : THEIR ALLEVIATION

One of the common problems that the users of the software encounter is the 'File not found' problem.

While running the software to process the raw data files, to create reports and to generate plots, the raw data file and the reference files (Names, Dist, Schedule) should all be on the hard disk of the PC being used for processing.

It is always useful to have backup copies of the program disks as well as the data files. This way, the data or the programs would not be lost if the diskettes are damaged or lost.
A.4 FEATURES OF THE INTERFACING SYSTEM:

The data collection and storage are done on the bus using the ADA System Modules namely the Passenger Count Modules and the Control Storage Module. This data is then dumped to the secondary storage units (Viz. Data General DG/ONE 3-1/2 “ disk) using the communications software (DATADUMP.BAS). This is then transferred onto the 5-1/2 “ disk drive of the IBMPC using the Auxiliary Drive booted up using the ‘Manzana’ software. The data is then processed on the IBMPC.

A.4.1 COMPONENTS USED:

The following components assist in the interfacing operations at Valley Metro at Roanoke, VA. It is an optional set of equipment used depending upon the way the system is set up and it varies from system to system.

- DG/ONE lap-held microcomputer along with a rechargeable battery kit to be used for charging and two 3-1/2 “ disk drives. Front drive holds "DATADUMP.BAS" disk and the rear drive has the working disk on which the raw data files are stored.

- Communications chord connecting the DG/ONE with the CSM on the bus.

- Auxiliary drive unit connected to the IBMPC. The unit is connected to the power source. It has the working disk with the desired raw data file in its drive.
A.4.2 PROCESS OF 'DATA DUMPING':

The data collected and stored on the bus by the PCMs and CSM has to be dumped to a secondary storage unit from which it is then transferred onto the processing computer. This process of 'Data Dumping' is done in the case of Roanoke Valley Metro Co., where a DG/ONE lap-held microcomputer is used as the secondary storage device, using the Communications Software (Operations Software). The step by step process for doing so is given in the section 'Usage - How to use the software'.

A.4.3 PROCESS OF 'DATA TRANSFER':

Once the raw datafile is dumped from the bus to the secondary storage unit, it has to be transferred to the processing computer to process and further, generate management reports and plots. This process of 'Data Transfer' is accomplished using the auxiliary unit (in the case of Roanoke Valley Metro where the datafile is transferred from a 3-1/2" to a 5-1/2" diskette). The step by step process is described in detail in the section 'Usage - How to use the software'.
Appendix B. EXTERNAL FILES

B.1 SYSTINFO.EXT FILE

The variables contained in the external file are listed below, along with an explanation of the meaning of each one.

SYST$
DISTBASE$,NAMEBASE$,BLOCKBASE$,SECT15BASE$
DATADRV$,DISTDRV$,NAMEDRV$,BLOCKDRV$,OUTDRV$,SECT15DRV$
DELTAL,DELTAT,DELTAS1,DELTAS2
CLICK,AF,SEATS,CAP

SYST$ - The name of the transit property.
DISTBASE$, NAMEBASE$, BLOCKBASE$, SECT15BASE$ - Base Names of the Distance, Names, Schedule Reference files and Section 15 file.

DATADRV$, DISTDRV$, NAMEDRV$, BLOCKDRV$, OUTDRV$, SECT15DRV$ - The disk drives from which the raw data file, distance, names and schedule reference files are retrieved respectively.

DELTAL, DELTAT - The distance variance and the time variance parameters.

DELTAS1 & DELTAS2 - The # of odometer impulses in the range for determining the outer and inner limit of a positive match.

CLICK - # of feet per odometer impulse.

AF - Adjustment Factor.

SEATS - # of seats per bus.

CAP - Capacity of the bus.

The actual systinfo.ext file used at Valley Metro, Roanoke, VA is

ROANOKE VALLEY METRO
DIST,NAMES,BLOCK,SECT15
A:, A:, A:, A:, A:
35, 40, 6, 3
8.58, 0.26, 42, 66
This external file contains information about the transit system which is required for creating the section 15 reports. The variables involved are listed below:

- **TRANSITID$**: Identification # assigned to the transit company.
- **MODE$**: Type of service (bus etc).
- **LEVEL$**: Reporting level for the transit company.
- **FISCALYR$**: The date of the end of the fiscal year.
- **WKDYBEGIN$**: The time (in 15 secs) at which the first service begins on a typical weekday.
- **AMPEAK**: The time at which the morning peak period begins (15 sec units).
- **MIDDAY**: The time at which the midday period begins (15 sec units).
- **PMPEAK**: The time at which the evening peak period begins (15 sec units).
- **NIGHT**: The time at which the night service begins (15 sec units).
- **WKDYEND**: The time at which the last service ends on a typical weekday.
- **SATBEGIN**: The time at which service begins on a typical saturday.
- **SATEND**: The time at which service ends on a typical saturday.
- **SUNBEGIN**: The time at which service begins on a typical sunday.
- **SUNEND**: The time at which service ends on a typical sunday.
The following are the actual values used at Roanoke Valley Metro, Roanoke VA:

3007, MB, R, 6/30/86
1360, 1440, 2160, 3600, 4560, 4700
2160, 4080, 0, 0
Appendix C. REFERENCE FILES

NAMES1.REF

1,"GRANBY",290
"MONT & CHARLOTTE"
"MONT & FREEMASON"
"MONT & MARKET"
"MONT & TAZEWELL"
"MONT & CITYHALL"
"MONT & BANK"
"CITYHALL & COURT"
"CITYHALL & CUMBERLAND"
"CITYHALL & ST PAUL"
"ST PAUL & PLUME"
"ST PAUL & MAIN PLAZA"
"ST PAUL WATERSIDE"
"WATERSIDE & COMMERCIAL"
"WATERSIDE & ATLANTIC"
"ATLANTIC & MAIN"
"MAIN & BANK"
"BANK & PLUME"
"PLUME & ATLANTIC"
"PLUME & GRANBY"
"PLUME & RANDOLPH"
"PLUME & BOUSH"
"BOUSH & CITYHALL"
"CITYHALL & RANDOLPH"
"CITYHALL & GRANBY"
"CITYHALL & FREEMASON"
"MONT & TAZEWELL"
"MONT & MARKET"
"MONT & FREEMASON"
"MONT & CHARLOTTE"
"MONT & BUTE"
"MONT & BRAMBLETON"
"MONT & OLNEY"
"MONT & VA BEACH"
"MONT & PRINCESS ANNE"
"MONT & 13TH"
"MONT & 14TH"
"MONT & 15TH"
"MONT & 16TH"
"MONT & 17TH"
"MONT & 18TH"
"MONT & 19TH"
"MONT & 20TH"
"MONT & 21ST"
"MONT & 25TH"
"MONT & 26TH"
"MONT & 27TH"
"MONT & 28TH"
"MONT & 29TH"
"MONT & GRANBY"
"GRANBY & 31ST"
"GRANBY & BROADWAY"
"GRANBY & 33RD"
"GRANBY & 34TH"
"GRANBY & 35TH"
"GRANBY & 36TH"
"GRANBY & 37TH"
"GRANBY & 38TH"
"GRANBY & LA VALLETTE"
"GRANBY & 39TH"
"GRANBY & PENNSYLVANIA"
"GRANBY & DELAWARE"
"GRANBY & 40TH"
"GRANBY & 41ST"
"GRANBY & 42ND"
"GRANBY & WILLOW WOOD"
"GRANBY & SO.ARDEN"
"GRANBY & ARDEN"
"GRANBY & SEVERN"
"GRANBY & BELVEDERE"
"GRANBY & WINDHAM"
"GRANBY & KINGSLEY"
"GRANBY & AFTON"
"GRANBY & BEVERLY"
"GRANBY & CONWAY"
"GRANBY & DUMONT"
"GRANBY & ELWOOD"
"GRANBY & FAYTON"
"GRANBY & SIR OLIVER RD"
"GRANBY & BLAKE"
"GRANBY & FILBERT"
"GRANBY & SEEKEL"
"GRANBY & THOLE"
"GRANBY & FORSYTH"
"GRANBY & FIFE"
"GRANBY & GLEN ECHO"
"GRANBY & GRANBY PARK"
"GRANBY & SUBURBAN"
"GRANBY & CROMWELL"
"GRANBY & LOUISIANA"
"GRANBY & LITTLE CREEK"
"GRANBY & RR TRACK"
"GRANBY & TAUSSIG"
"GRANBY & CEMETRY GATE"
"GRANBY & CAMP ASPASS"
"GRANBY & BAYVIEW"
"GRANBY & EVANS"
"GRANBY & WESTMONT"

Appendix C. REFERENCE FILES 168
"GRANBY & HOWE"
"GRANBY & CHESTER"
"GRANBY & BAY"
"GRANBY & OCEAN"
"GRANBY & LEICESTER"
"GRANBY & RANDALL"
"GRANBY & LORENKO"
"GRANBY & GILPIN"
"GRANBY & TIDEWATER DR"
"GRANBY & BATTERSEA"
"GRANBY & CHERRY"
"GRANBY & GOVERNMENT"
"GRANBY & SEAVIEW"
"GRANBY & E.OCEAN VIEW"
"A VIEW & 1ST VIEW"
"E.OCEAN VIEW & WELLS PARKWAY"
"E.OCEAN VIEW & HAMMETT"
"E.OCEAN VIEW & SHERWOOD"
"E.OCEAN VIEW & NORFOLK"
"E.OCEAN VIEW & CHESAPKE BLVD"
"E.OCEAN VIEW & ELMORA"
"E.OCEAN VIEW & CHESAPEAKE"
"E.OCEAN VIEW & ATLANS"
"E.OCEAN VIEW & WARWICK"
"E.OCEAN VIEW & BEACHVIEW"
"E.OCEAN VIEW & HOLIDAY SANDS"
"E.OCEAN VIEW & STURGIS"
"E.OCEAN VIEW & WILLOW"
"E.OCEAN VIEW & BEAUMONT"
"E.OCEAN VIEW & MIDBLOCK"
"E.OCEAN VIEW & GROVE"
"E.OCEAN VIEW & MIDBLOCK"
"E.OCEAN VIEW & CAPEVIEW"
"E.OCEAN VIEW & MIDBLOCK"
"E.OCEAN VIEW & INLET"
"E.OCEAN VIEW & MIDBLOCK"
"E.OCEAN VIEW & 1ST BAY"
"E.OCEAN VIEW & 2ND BAY"
"E.OCEAN VIEW & 3RD BAY"
"E.OCEAN VIEW & 4TH BAY"
"E.OCEAN VIEW & 5TH BAY"
"E.OCEAN VIEW & 6TH BAY"
"E.OCEAN VIEW & 7TH BAY"
"E.OCEAN VIEW & 8TH BAY"
"E.OCEAN VIEW & 9TH BAY"
"E.OCEAN VIEW & 10TH BAY"
"E.OCEAN VIEW & 11TH BAY"
"E.OCEAN VIEW & 12TH BAY"
"E.OCEAN VIEW & 13TH BAY"
"E.OCEAN VIEW & 14TH BAY"
"E.OCEAN VIEW & 15TH BAY"
"E.OCEAN VIEW & 16TH BAY"
"E.OCEAN VIEW & 17TH BAY"
"E.OCEAN VIEW & 18TH BAY"
"E.OCEAN VIEW & 19TH BAY"
"E.OCEAN VIEW & 20TH BAY"
"SHORE & 21ST BAY"
"SHORE & 22ND BAY"
"SHORE & PLEASANT"
"SHORE & PRETTY LAKE"
"SHORE & DUNNING"
"SHORE & HACIENDA"
"SHORE & CALIENTE"
"SHORE & LITTLE CREEK"
"SHORE & GATE 1"
"MIDWAY & EXCHANGE"
"GUADALCANAL & MIDWAY ROAD"
"GUAM ROAD & GUADALCANAL"
"GATE 1 & LITTLE CREEK"
"SHORE DRIVE & LITTLE CREEK"
"SHORE & DUNNING"
"SHORE & MIDBLOCK"
"SHORE & PRETTY LAKE"
"SHORE & PLEASANT"
"SHORE & 22ND BAY"
"SHORE & 21ST BAY"
"SHORE & SHORE DRIVE"
"E.OCEAN VIEW & 19TH BAY"
"E.OCEAN VIEW & 18TH BAY"
"E.OCEAN VIEW & 17TH BAY"
"E.OCEAN VIEW & 16TH BAY"
"E.OCEAN VIEW & 15TH BAY"
"E.OCEAN VIEW & 13TH BAY"
"E.OCEAN VIEW & 11TH BAY"
"E.OCEAN VIEW & 9TH BAY"
"E.OCEAN VIEW & 7TH BAY"
"E.OCEAN VIEW & 5TH BAY"
"E.OCEAN VIEW & 3RD BAY"
"E.OCEAN VIEW & 1ST BAY"
"E.OCEAN VIEW & MIDBLOCK"
"E.OCEAN VIEW & INLET"
"E.OCEAN VIEW & MIDBLOCK"
"E.OCEAN VIEW & MIDBLOCK"
"E.OCEAN VIEW & GROVE"
"E.OCEAN VIEW & MIDBLOCK"
"E.OCEAN VIEW & WILLOW"
"E.OCEAN VIEW & STURGIS"
"E.OCEAN VIEW & HOLIDAY SANDS"
"E.OCEAN VIEW & BEACHVIEW"
"E.OCEAN VIEW & WARWICK"
"E.OCEAN VIEW & CHESAPEAKE ST."
"E.OCEAN VIEW & ELNORA"
"E.OCEAN VIEW & CHESAPKE BLVD"
"E.OCEAN VIEW & NORFOLK"
"E.OCEAN VIEW & SHERWOOD"
"GRANBY & E.OCEAN VIEW"
"GRANBY & BALVIEW"
"GRANBY & SEAVIEW"
"GRANBY & D VIEW"
"GRANBY & GOVERNMENT"
"GRANBY & DUPREE"
"GRANBY & CHERRY"
"GRANBY & OSBORNE"
"GRANBY & TIDEWATER DR"
"GRANBY & SLOANE"
"GRANBY & GILPIN"
"GRANBY & LORENGO"
"GRANBY & RANDALL"
"GRANBY & LEICESTER"
"GRANBY & OCEAN"
"GRANBY & BAY"
"GRANBY & CHESTER"
"GRANBY & WESTMONT"
"GRANBY & LANDALE"
"GRANBY & EVANS"
"GRANBY & CAP"
"GRANBY & BAYVIEW"
"GRANBY & CAMP ASPASS"
"GRANBY & CEMETRY GATE"
"GRANBY & TAUSSIG"
"GRANBY & RR TRACK"
"GRANBY & LITTLE CREEK"
"GRANBY & BURLEIGH"
"GRANBY & MAYCOX"
"GRANBY & NORTH SHORE"
"GRANBY & SUBURBAN"
"GRANBY & ELVIN"
"GRANBY & MARCY"
"GRANBY & GLEN ECHO"
"GRANBY & THOLE"

Appendix C. REFERENCE FILES
"GRANBY & OXFORD"
"GRANBY & BRACKENBRIDGE"
"GRANBY & ST CLAIR"
"GRANBY & OAKGROVE"
"GRANBY & CARLISLE WAY"
"GRANBY & TALBOT HALL"
"GRANBY & WHITING"
"GRANBY & PAINTER"
"GRANBY & KINGSLEY"
"GRANBY & RIDGELY"
"GRANBY & SO. RIDGELY"
"GRANBY & BELVEDERE"
"GRANBY & SEVERN"
"GRANBY & ARDEN"
"GRANBY & SO. ARDEN"
"GRANBY & WILLOW WOOD"
"GRANBY & CONNECTICUT"
"GRANBY & DELAWARE"
"GRANBY & PENNSYLVANIA"
"GRANBY & 39TH"
"GRANBY & 38TH"
"GRANBY & 37TH"
"GRANBY & 36TH"
"GRANBY & 35TH"
"GRANBY & 34TH"
"GRANBY & 33RD"
"GRANBY & BROADWAY"
"GRANBY & 32ND"
"GRANBY & 31ST"
"MONT & 30TH"
"MONT & 29TH"
"MONT & 28TH"
"MONT & 27TH"
"MONT & 26TH"
"MONT & 25TH"
"MONT & 21ST"
"MONT & 20TH"
"MONT & 19TH"
"MONT & 18TH"
"MONT & 17TH"
"MONT & 16TH"
"MONT & 15TH"
"MONT & 14TH"
"MONT & 13TH"
"MONT & PRINCESS ANNE"
"MONT & 11TH"
"MONT & VA BEACH"
"MONT & WILSON"
"MONT & OLNEY"
"MONT & STARKE"
"MONT & BRAMBLETON"
"MONT & BUTE"
"MONT & CHARLOTTE"
DISTANCE REFERENCE FILE

DIST1.REF
1, "GRANBY", 290, 166
1, 1
2, 117
3, 18
4, 105
5, 62
6, 42
7, 149
8, 71
9, 119
10, 9
11, 101
12, 80
13, 61
14, 74
15, 97
16, 52
17, 12
18, 128
19, 19
20, 40
21, 50
22, 30
23, 30
24, 47
193,184
194,173
195,96
196,86
197,137
198,247
199,95
200,66
201,166
202,131
203,787
204,119
205,75
206,29
207,69
208,78
209,59
210,34
211,190
212,85
213,55
214,42
215,30
216,80
217,26
218,68
219,130
220,189
**SCHEDULE REFERENCE FILE**

**BLOCK2.REF**

<table>
<thead>
<tr>
<th>2,3,8,1240,2156</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>40</td>
</tr>
<tr>
<td>90</td>
</tr>
<tr>
<td>111</td>
</tr>
<tr>
<td>166</td>
</tr>
<tr>
<td>203</td>
</tr>
<tr>
<td>229</td>
</tr>
<tr>
<td>277</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1240</td>
</tr>
<tr>
<td>1280</td>
</tr>
<tr>
<td>1320</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1320</td>
</tr>
<tr>
<td>1360</td>
</tr>
<tr>
<td>1420</td>
</tr>
<tr>
<td>1436</td>
</tr>
<tr>
<td>1496</td>
</tr>
<tr>
<td>1532</td>
</tr>
<tr>
<td>1592</td>
</tr>
<tr>
<td>1680</td>
</tr>
<tr>
<td>1740</td>
</tr>
<tr>
<td>1780</td>
</tr>
</tbody>
</table>
Appendix D. PROGRAM LISTINGS

10 ' <<APCDATA>>
20 ' WRITTEN BY P. ANDERSON AND R. SRINATH
40 ' THIS PROGRAM PROCESSES THE RAW DATA INTO AN OUTPUT FILE OF A PARTICULAR
41 ' FORMAT. IN THE SECOND OPTION A BUS STOP LISTING FACILITY IS PROVIDED.
42 ' IT SHOULD HOWEVER, BE REMEMBERED THAT ANY RAW DATA FILE HAS TO BE
43 ' PROCESSED BEFORE ANY LISTING OR REPORTS OR PLOTS ARE GENERATED.
50 CLEAR
60 '---------retrieve system parameters--------------------------------------
70 ' RETRIEVE SYSTEM PARAMETERS FROM SYSTINFO.EXT FILE. SYST$ - TRANSIT
71 ' PROPERTY NAME., DISTBASE$, BLOCKBASE$, NAMEBASE$ AND SECT15BASE$ - BASE
72 ' NAMES OF THE DISTANCE, SCHEDULE, NAMES REFERENCE FILES AND SECTION 15
73 ' FILE RESPECTIVELY., DATADR$, DISTDR$, BLOCKDR$, NAMEDR$, OUTDR$, SECT15DR$
74 ' ARE THE DRIVES ON WHICH THE DATA FILE, THE ABOVE REFERENCE FILES, THE OUTPUT
75 ' FILES ARE STORED., DELTAL - # OF ODOMETER CLICKS TO SET DIST VARIANCE
76 ' PARAMETER., DELTAT - # OF TIME CLICKS WHICH SET THE TIME VARIANCE PARAMETER
77 ' DELTAS1 - # OF OD CLICKS WHICH SET THE OUTER LIMIT OF A POSITIVE MATCH.,
78 ' DELTAS2 - # OF OD CLICKS WHICH SET THE INNER LIMIT OF A POSITIVE MATCH.,
79 ' CLICK - NO. OF FEET PER ODOMETER CLICK., AF - ADJUSTMENT FACTOR.,
81 'SEATS - # OF SEATS PER BUS., CAP - CAPACITY OF THE BUS.
85 OPTION BASE 1
90 OPEN "SYSTINFO.EXT" FOR INPUT AS 1
100 INPUT #1,SYST$
110 INPUT #1,DISTBASE$,BLOCKBASE$,NAMEBASE$,SECT15BASE$
120 INPUT #1,DATADR$,DISTDRV$,BLOCKDRV$,NAMEDRV$,OUTDRV$,SECT15DRV$
130 INPUT #1,DELTAL,DELTAT,DELTAS1,DELTAS2
140 INPUT #1,CLICK,AF,SEATS,CAP
150 CLOSE #1
155 'TO ACCOUNT FOR THE BUS OPERATOR GETTING ON AT THE BEGINNING OF THE DAY.
160 PASS = -1
165 'SET THE MODE TO HIGH(standard) RESOLUTION GRAPHICS MODE. DRAW THE WINDOWS.
170 SCREEN 2
171 LINE (60,30) * (590,175),1,8
172 LINE (58,28) * (592,177),1,8
174 LOCATE 6,25,1 : PRINT "AUTOMATIC DATA ACQUISITION SYSTEM"
175 LOCATE 10,35,1 : PRINT "DEVELOPED BY"
176 LOCATE 14,30,1 : PRINT "DEPARTMENT OF CIVIL ENGG"
177 LOCATE 16,41,1 : PRINT "$"
178 LOCATE 18,30,1 : PRINT "ELECTRICAL ENGINEERING"
179 LOCATE 20,27,1 : PRINT "VIRGINIA TECH, BLACKSBURG, VA."
180 FOR I = 1 TO 5000 : REM TO MAKE SCREEN WAIT FOR A LITTLE WHILE.
181 NEXT I
182 '---------------PRESENT MENU-------------------------------
183 '
184 'A DEBUG MODE IS BUILT IN HERE. TO WATCH THE VARIOUS VARIABLES IN THE
185 'DATA PROCESSING STEP, SET THE DEBUG$ = "T". ELSE KEEP IT AS IT IS "F".
186 'THIS WAY THE LOGIC CAN BE FOLLOWED.
187 DEBUG$ = "F"
190 KEY OFF
200 CLS
210 CLS
212 'SET MODE TO STD RESOLUTION GRAPHICS MODE. DRAW WINDOWS AND THEN PRESENT MENU.
215 SCREEN 2
220 LINE (80,35) - (550,150),1,8

Appendix D. PROGRAM LISTINGS
225 LINE (77,32) - (553,152),1,8
230 LOCATE 6,20,0 :PRINT SYST$, " APC-DATA PROGRAM MENU"
240 LOCATE 8,20,0 :PRINT "Please choose one of the following: "
250 LOCATE 10,25,0 :PRINT "1 - PROCESS DATA FILE"
260 LOCATE 12,25,0 :PRINT "2 - GENERATE BUS STOP LISTING"
270 LOCATE 14,25,0 :PRINT "3 - QUIT (EXIT TO DOS)"
280 LOCATE 16,20,0 :INPUT "What is the number of your choice ";CHOICE
290 IF CHOICE < 1 OR CHOICE > 3 GOTO 300 ELSE 310 : REM ERROR TRAP.
300 LOCATE 18,20,0 :PRINT "Invalid choice - try again ":GOTO 280
305 'GOTO VARIOUS SUBROUTINES DEPENDING UPON THE CHOICE.
310 ON CHOICE GOSUB 330,3690,5120
320 'THIS OPTION PROCESSES THE RAW DATA INTO AN OUTPUT FILE.
330 '<<<<<<<<<<<<<<< PART 1 - PROCESS RAW DATA >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
331 CLS 1 GOTO 350
350 CLS : PRINT
360 'THE CONCERNED RAW DATA FILE IS RETRIEVED FROM THE DRIVE SPECIFIED IN
370 'THE SYSTINFO.EXT FILE.
372 '
380 '------------------retrieve raw data file-------------------------------------
390 '
395 'THE DIMENSION STATEMENTS BELOW ARE SET TO RECEIVE THE RAW DATA FILE OF
396 'SIZE = 1000 RECORDS LONG. FOR BIGGER FILES, REDIMENSION THE ARRAYS BELOW.
397 '
400 DIM TYPE(1000),TIME(1000),DIST(1000),ONS(1000),OFFS(1000)
402 'DRAW WINDOWS.
405 LINE (50,25) - (500,150),1,8
406 LINE (47,23) - (503,152),1,8
408 LOCATE 5,10,0 : PRINT "Just give the filename (Ex B10MAY07)"
410 LOCATE 10,10,0
420 INPUT "Which raw data file do you want to process?";RAWDATAS
425 'ADD THE DRIVE AND EXTENSION(.DAT) TO THE FILENAME.
430 F$ = DATADRV$ + RAWDATAS + ".DAT"
435 'RECIEVE THE RAW DATA FILE(F$) FROM THE DRIVE SPECIFIED.
440 OPEN F$ FOR INPUT AS #1
450 LOCATE 12,25,0 : PRINT "Retrieving ";F$
455 'IF # OF RECORDS IN THE DATA FILE IS GREATER THAN 1000, CHANGE I TO THAT
456 'THAT VALUE. FOR EX IF # IS 2000 CHANGE I = 1 TO 2000(INSTEAD OF 1000) BELOW.
457 'REMEMBER THE DIMENSION STATEMENT IN 400. CHANGE THAT BEFORE CHANGING THIS.
460 FOR I = 1 TO 1000
470 IF EOF(1) THEN 570
480 INPUT # 1,TYPE(I),TIME(I),DIST(I),ONS(I),OFFS(I)
485 'BRANCH OFF DEPENDING UPON THE LOG TYPE (1,2,3,4,5,6,7,8,9,10,11,12).
486 'FOR TRT NORFOLK INSERT BRANCHES FOR LOGS 9,10,12,13. LOGS 14 & 15 ARE
487 'SAME AS LOG 12 FOR ROANOKE VALLEY METRO.
490 ON TYPE(I) GOTO 530,520,500,510,500,500,530,530,530,530,500,560
495 'LOGS 3,5,6,11 ARE ACTIVITY RECORDS.
500 ACTNO = ACTNO + 1 :GOTO 530
505 'ACCUMULATE # OF LOG 4s = DISTANCE OVERFLOW LOGS.
510 FOURS = FOURS + 1 :GOTO 530
515 'ACCUMULATE # OF LOG 2s = HOUR OVERFLOW LOGS.
520 TNOS = TNOS + 1 :GOTO 530
525 'ACCUMULATE TIME IN 15 SEC UNITS FOR ALL RECORDS.
530 SUMTIME = SUMTIME + TIME(I)
535 'ACCUMULATE DIST IN ODOMETER CLICKS FOR ALL RECORDS.
540 SUMDIST = SUMDIST + DIST(I)
545 'CALC TOTAL TIME IN 15 SEC UNITS TILL THE DUMP TIME SINCE LOG 7 IS CREATED.
546 'LOG 7,8,12(14,15) ARE ALWAYS CREATED AT THE TIME OF DATA DUMPING.
550 IF TYPE(I) = 7 THEN TOTTIME = SUMTIME + 240*TNOS
560 NEXT I
565 'LAST 3 RECORDS IN THE DATA FILE ARE MANUALLY INPUT AT THE TIME OF DATA
566 'DUMPING.
570 A = I - 3 : B = I - 2 : C = I - 1
580 MONTH = TIME(A) :DAY = DIST(A) :YEAR = ONS(A) :ROUTENO = OFFS(A)
590 DAYOFWK = TIME(B) :BLOCK = DIST(B) :HOURS = ONS(B) :MINUTES = OFFS(B)
600 BUSID = TIME(C)
610 NUMLOGS = C
620 TOTDIST = SUMDIST + 256*FOURS : IF DEBUG$ = "T" THEN 621 ELSE 630

Appendix D. PROGRAM LISTINGS
621 PRINT "THE TOTAL TIME RECORDED =";TOTTIME
622 PRINT "THE TOTAL DISTANCE RECORDED =";TOTDIST
623 PRINT "STRIKE ANY KEY TO CONTINUE"
624 WS = INKEY$: IF WS = "" THEN 624
630 DATADATES = STR$(MONTH) + "·" + RIGHT$(STR$(DAY),LEN(STR$(DAY)) - 1)
   + "·" + RIGHT$(STR$(YEAR),LEN(STR$(YEAR)) - 1)
640 CLOSE #1
660 '-------------allow user to view the data-------------------------------
670 '
671 'GIVE AN OPTION TO VIEW THE RAW DATA.
680 LOCATE 14,10,1
690 INPUT "Would you like to view the raw data (Y or N)";A$ 
700 IF A$ = "Y" OR A$ = "y" THEN 740
710 IF A$ = "N" OR A$ = "n" THEN 890
720 LOCATE 15,10,1: PRINT "INVALID CHOICE - ENTER EITHER Y OR N - TRY AGAIN"
730 PRINT: GOTO 680
740 CLS
750 FOR I = 1 TO NUMLOGS
760 PRINT USING "#####";TYPE(I);TIME(I);DIST(I);ONS(I);OFFS(I)
770 NUMPRINT = NUMPRINT + 1
780 IF NUMPRINT = 20 THEN 800 ELSE 850
790 WS = ""
800 PRINT: PRINT "Strike Q to quit viewing the data - strike any other key to continue."
810 WS = INKEY$: IF WS = "" THEN 810
820 IF WS = "q" OR WS = "Q" THEN 890
830 CLS: PRINT
840 NUMPRINT = 0
850 NEXT I
860 WS = "": PRINT: PRINT "strike any key to continue"
870 WS = INKEY$: IF WS = "" THEN 870
880'
890 WS = ""
900 '---------create activity records from raw data-----------------------------
910 '
920 CLS
925 'THE NO OF ACTIVITY RECORDS = ACTNO.
930 Z = ACTNO
940 DIM ACTTYPE(Z),ACTTIME(Z),ACTDIST(Z),LSTDIST(Z),ACTONS(Z),ACTOFFS(Z),PASSLOAD(Z)
945 'CALC STARTING TIME OF DATA COLLECTION PERIOD.
950 STRTIME = HOURS*240 + MINUTES*4 - TOTTIME
960 K = 0
962 'DRAW WINDOWS.
965 LINE (70,70) - (575,120),1,B
966 LINE (67,68) - (578,122),1,B
970 LOCATE 13,15,0 Z PRINT "Creating activity records from raw data - Please wait."
980 FOR I = 1 TO NUMLOGS - 3
985 'ACCUMULATE TIME.
990 CUMTIME = CUMTIME + TIME(I)
995 'ACCUMULATE DIST.
1000 CUMDIST = CUMDIST + DIST(I)
1005 'BRANCH OFF DEPENDING UPON THE LOG TYPE.
1010 ON TYPE(I) GOTO 1180,1020,1040,1030,1050,1040,1180,1180,1180,1180,1180,1180
1015 'ACCUMULATE LOG 2(HOUR OVERFLOW) TIMES.
1020 CUMTIME = CUMTIME + 240 :GOTO 1180
1025 'ACCUMULATE LOG 4(DISTANCE OVERFLOW) DISTANCES.
1030 CUMDIST = CUMDIST + 256 :GOTO 1180
1040 K = K + 1
1050 ACTTYPE(K) = TYPE(I)
1060 ACTTIME(K) = STRTIME + CUMTIME
1070 ACTDIST(K) = CUMDIST
1080 IF K = 1 THEN LSTDIST(K) = CUMDIST :GOTO 1100
1085 'CALCULATE DIST FROM PREVIOUS ACTIVITY RECORD.
1090 LSTDIST(K) = ACTDIST(K) - ACTDIST(K-1)
1095 'ACTTYPE OF '5' IS PASSENGER ACTIVITY.
1100 IF ACTTYPE(K) = 5 THEN 1130
1101 IF ACTTYPE(K) = 11 THEN 1130

Appendix D. PROGRAM LISTINGS
1110 IF ACTTYPE(K) = 3 OR ACTTYPE(K) = 6 THEN PASSLOAD(K) = PASS
1120 GOTO 1160
1130 ACTONS(K) = ONS(I)
1140 ACTOFFS(K) = OFFS(I)
1150 PASSLOAD(K) = PASS + ONS(I) - OFFS(I)
1160 IF PASSLOAD(K) < 0 THEN PASSLOAD(K) = 0
1165 'CALCULATE PASSENGER LOAD
1170 PASS = PASSLOAD(K)
1180 NEXT I
1185 'RESET VARS.
1190 ERASE TYPE,TIME,DIST,ONS,OFFS
1200 '
1210 '----------retrieve distance and schedule reference files------------------
1220 '
1230 CLS :PRINT
1235 'ADD DRIVE AND EXTENSION
1240 D$ = DISTDRVS + DISTBASE$ + RIGHT$(STR$(ROUTENO),LEN(STR$(ROUTENO)) - 1) + "REF"
1250 OPEN D$ FOR INPUT AS 2
1260 IF DAYOFNK = 6 THEN 1270 ELSE 1280
1270 SS = BLOCKDRVS + BLOCKBASE$ + "6" + RIGHT$(STR$(BLOCK),LEN(STR$(BLOCK)) - 1) + "REF" : GOTO 1310
1280 IF DAYOFNK = 7 THEN 1290 ELSE 1300
1290 SS = BLOCKDRVS + BLOCKBASE$ + "7" + RIGHT$(STR$(BLOCK),LEN(STR$(BLOCK)) - 1) + "REF" : GOTO 1310
1300 SS = BLOCKDRVS + BLOCKBASE$ + RIGHT$(STR$(BLOCK),LEN(STR$(BLOCK)) - 1) + "REF"
1303 'DRAW WINDOWS
1305 LINE (100,50) - (575,150),1,B
1306 LINE (97,48) - (577,152),1,B
1310 LOCATE 11,20,0 :PRINT "Reference files will be the following:"
1320 LOCATE 12,20,0 :PRINT D$; " $ SS
1325 'GIVE OPTION TO CHANGE THE REFERENCE FILES.
1330 LOCATE 14,20,0 :INPUT "Are these the files you wish (Y or N)?",A$
1340 IF A$ = "y" OR A$ = "Y" THEN 1410
1350 IF A$ = "n" OR A$ = "N" THEN 1365
1360 LOCATE 16,20,0 :PRINT "INVALID INPUT - ENTER EITHER Y OR N - TRY AGAIN" :
GOTO 1330
1365 CLS : LINE (50,50) - (590,150),1,8
1366 LINE (47,48) - (593,152),1,8
1367 LOCATE 10,10,1 : PRINT "Indicate Drive & Extension : EX - A:DIST1.REF"
1370 LOCATE 11,10,1
1380 INPUT "What distance reference file do you prefer ? ",D$
1390 LOCATE 13,10,1 : PRINT "Indicate Drive & Extension : EX - A:BL10.REF"
1395 LOCATE 14,10,1
1400 INPUT "What schedule (block) reference file do you prefer ? ",S$
1410 CLS
1420 OPEN S$ FOR INPUT AS 3
1425 LINE (80,60) - (500,120),1,8
1426 LINE (78,58) - (502,122),1,8
1430 LOCATE 11,20,0
1440 PRINT "Retrieving reference files ":
1445 REMETING DISTANCE REFERENCE FILE.
1450 INPUT #2,ROUTENO,ROUTENAME$,NOSTOPS,TURNPOINT
1460 DIM DFLS(NOSTOPS)
1470 FOR I = 1 TO NOSTOPS
1480 INPUT #2,JUNK,DFLS(I)
1485 '1490 GIVES AN OPTION TO ADJUST DIST BY AN ADJT FACTOR FOR THE BUS PULLING
1486 'IN AND OUT OF A STOP.
1490 'DFLS(I) = DFLS(I) * AF
1500 TEMP = TEMP + DFLS(I)
1510 IF I = TURNPOINT THEN TRIPDIST = TEMP
1520 NEXT I
1530 LOOPDIST = TEMP : IF DEBUG$ = "T" THEN 1531 ELSE 1540
1531 PRINT "THE REFERENCE DISTANCE =";LOOPDIST
1532 PRINT "STRIKE ANY KEY TO CONTINUE"
1533 W$ = INKEY$: IF W$ ="" THEN 1533
1540 CLOSE #2
1550 

Appendix D. PROGRAM LISTINGS
1555 'RETRIEVE BLOCK REFERENCE FILE.
1560 INPUT #3,BLOCKNO,LOOPS,TIMEPTS,STARTTIME,ENDTIME
1570  DIM TIMEPT(TIMEPTS)
1580  FOR K = 1 TO TIMEPTS
1590      INPUT #3,TIMEPT(K)
1600  NEXT K
1610  DIM SCHED(LOOPS,TIMEPTS)
1620  FOR I = 1 TO LOOPS
1630      FOR J = 1 TO TIMEPTS
1640         INPUT #3,SCHED(I,J)
1650      NEXT J
1660  NEXT I
1670  '
1680  CLOSE #3
1690  '
1700 '----------------MATCH ACTIVITIES TO STOPS-----------------------
1710  '
1720  CLS
1722 'DRAW WINDOWS
1725  LINE (60,60) - (550,120),1,B
1726  LINE (58,58) - (552,122),1,B
1730  LOCATE 10,15,0
1740  PRINT "Matching activities to stops - please wait."
1750  DIM STOPNUM(ACTNO),MATCH(ACTNO),SCHEDTIME(ACTNO),BEGLOOP(LOOPS),
1750       ESTMATCH(LOOPS),ENDLOOP(LOOPS),TENDLOOP(LOOPS),TBEGLOOP(LOOPS),MATCHRATIO(LOOPS)
1760  '
1770 '----------------locate activities with time near scheduled time for stop #1-----------------
1780  '
1785 'TAKE THE FIRST COMPLETE LOOP BY SEEING THAT TIME AT TIMEPOINT 1 OF LOOP
1786 'IS NOT EQUAL TO 0
1790  IF SCHED(1,1) > 0 THEN L = 1 ELSE L = 2
1800  REFTIME = SCHED(L,1) : IF DEBUG$ = "T" THEN 1801 ELSE 1810
1801  PRINT "REF TIME IS";SCHED(L,1)
1802  PRINT: PRINT "HIT ANY KEY TO CONTINUE"
1803 W$ = INKEY$: IF W$ ="" THEN 1803
1810 IF DEBUG$ = "T" THEN 1811 ELSE 1820
1811 PRINT "ACTIVITY TIME OF I IS";ACTTIME(I)
1812 PRINT "REF TIME IS";REFTIME
1813 PRINT : PRINT "HIT ANY KEY TO CONTINUE"
1814 W$ = INKEY$: IF W$ ="" THEN 1814
1817 'CALCULATE TIME DIFFERENCE BETWEEN ACTUAL AND SCHEDULED TIME
1820 TIMEDIFF = ABS(ACTTIME(I) - REFTIME)
1825 'CHECK IF TIME DIFFERENCE ABOVE IS LESS THAN DELTAT FOR THE ACT. RECORD
1826 'TO BE CONSIDERED AS A POTENTIAL BEGINNING OF A LOOP.
1830 IF TIMEDIFF < DELTAT THEN 1890
1831 IF DEBUG$ = "T" THEN 1832 ELSE 1840
1832 PRINT "THE TIME DIFFERENCE IS";TIMEDIFF
1835 'IF ACTTIME IS GREATER THAN REFTIME AND TIMEDIFF>=DELTAT THEN GO TO
1836 'REFTIME OF NEXT LOOP
1840 IF ACTTIME(I) > REFTIME THEN 1870
1845 'ELSE GOTO NEXT ACTIVITY RECORD.
1850 I = I + 1 : IF I > ACTNO THEN 2200
1860 GOTO 1820
1870 L = L + 1 : IF L > LOOPS THEN 2200
1880 REFTIME = SCHED(L,1) : IF L <= LOOPS THEN 1820
1890 GOTO 1900
1900 ' 1910 '--------calculate the match ratio for the potential loop----------
1920 K = I
1925 'SET TEMP BEGINNING OF POTEN LOOP
1930 TBEGLOOP(L) = K : IF DEBUG$ = "T" THEN 1931 ELSE 1940
1931 PRINT "TBEGLOOP IS";TBEGLOOP(L)
1932 PRINT : PRINT "HIT ANY KEY TO CONT"
1933 W$ = INKEY$: IF W$ ="" THEN 1933
1940 REFSUM = 0 : LDIST = 0 : J = 1 : MATCHES = 0 :LOOPSUM = 0
1950 FOR I = K TO 1000
1960 GOTO 1990
1970 J = J + 1 : IF J > NOSTOPS THEN 2110

Appendix D. PROGRAM LISTINGS
1980     REFSUM = REFSUM + DFLS(J) : GOTO 2030
1990     IF I = ACTNO THEN 2200
1992 PRINT "ACTIVITY NO IS";I
1995 'LSTDIST(I) IS DIST FROM PREVIOUS RECORD
2000     LDIST = LDIST + LSTDIST(I)
2005 '
2010 LOOPSUM = LOOPSUM + LSTDIST(I) : IF DEBUGS = "T" THEN 2011 ELSE 2014
2011 PRINT "LOOPSUM=";LOOPSUM
2012 PRINT : PRINT "HIT ANY KEY"
2013 W$ = INKEY$: IF W$ ="" THEN 2013
2014 LOOPDIFF = ABS(LOOPSUM - LOOPDIST) :
REM FIND LOOPDIFF IN DIST BET ACTUAL AND REF DIST.
2015 IF LOOPDIFF <= DELTAL THEN 2101 REM CHECK TO SEE IF LOOPDIFF
2020 IF LOOPSUM > LOOPDIST + DELTAL THEN 2110 REM TAKE END OF LOOP
AS PREV RECORD.
2021 IF DEBUGS = "T" THEN 2022 ELSE 2030
2022 PRINT "LSTDIST(I) IS";LSTDIST(I)
2024 PRINT "LDIST = ";LDIST
2025 PRINT : PRINT "HIT ANY KEY"
2026 W$ = INKEY$: IF W$ ="" THEN 2026
2027 'FIND DIFF BETWEEN ACTUAL AND REFERENCE DISTANCE.
2030 DIFF = ABS(LDIST - REFSUM) : IF DEBUGS = "T" THEN 2031 ELSE 2040
2031 PRINT "DIFF IS";DIFF
2032 PRINT : PRINT "HIT ANY KEY"
2033 W$ = INKEY$: IF W$ ="" THEN 2033
2034 PRINT "REFSUM IS";REFSUM
2035 PRINT : PRINT "HIT ANY KEY"
2036 W$ = INKEY$: IF W$ ="" THEN 2036
2037 'CONDITION FOR MATCHES
2040 IF DIFF <= DELTAS1 THEN 2060
2050     IF LDIST < REFSUM THEN 2100 ELSE 1970
2060 "

Appendix D. PROGRAM LISTINGS 199
2070 MATCHES = MATCHES + 1
2080 REFSUM = 0 : LDIST = 0

2100 NEXT I
2101 TENDLOOP(L) = I : GOTO 2120
2102 DEBUG$ = "T"
2110 TENDLOOP(L) = I - 1 : IF DEBUG$ = "T" THEN 2111 ELSE 2120
2111 PRINT "TENDLOOP(L) ="; TENDLOOP(L)
2112 PRINT : PRINT "HIT ANY KEY"
2113 W$ = INKEY$ : IF W$ = "" THEN 2113
2115 'CALC MATCHRATIO AS INDICATOR OF BEST POSSIBLE MATCHING.
2120 MATCHRATIO(L) = MATCHES/(TENDLOOP(L) - TBEGINLOOP(L))
2121 IF DEBUG$ = "T" THEN 2122 ELSE 2130
2122 PRINT "MATCHRATIO =$"; MATCHRATIO(L)
2125 'CHECK FOR BEST OR HIGHEST VALUE OF MATCHRATIO.
2130 IF MATCHRATIO(L) > BESTMATCH(L) THEN 2160 ELSE 2140
2140 I = K + 1 : TRAVISUM = 0
2150 IF I = ACTNO THEN 2200 ELSE 1820
2160 BESTMATCH(L) = MATCHRATIO(L) : BEGLOOP(L) = TBEGINLOOP(L)
2170 ENDOLOOP(L) = TENDLOOP(L) : TRAVISUM = 0
2171 'PRINT "BESTMATCH(L) IS"; BESTMATCH(L)
2172 'PRINT : PRINT "HIT ANY KEY"
2173 W$ = INKEY$ : IF W$ = "" THEN 2173
2180 I = K + 1 : GOTO 1820
2190 'PRINT "THIS IS A TEST"
2200 IF DEBUG$ = "T" THEN 2201 ELSE 2210
2201 PRINT "MATCHING ACTIVITIES WITHIN GOOD LOOPS"
2210 'MATCHING ACT. WITHIN GOOD LOOPS
2220 FOR L = 1 TO LOOPS
2230 IF BEGLOOP(L) = 0 THEN 2540
2240 REFSUM = 0 : LDIST = 0 : J = 1
2250 FOR I = BEGLOOP(L) TO ENDOLOOP(L)
2260 GOTO 2290
2270 J = J + 1 : IF J > NOSTOPS THEN 2540
2280 REFSUM = REFSUM + DFLS(J) : GOTO 2300

Appendix D. PROGRAM LISTINGS
2290 LDIST = LDIST + LSTDIST(I) : IF DEBUG$ = "T" THEN 2291 ELSE 2300
2291 PRINT "LDIST = " : LDIST
2300 DIFF = ABS(LDIST - REFSUM) : IF DEBUG$ = "T" THEN 2301 ELSE 2310
2301 PRINT "DIFF IS ": DIFF : PRINT "REFSUM = ": REFSUM
2302 PRINT : PRINT "HIT ANY KEY TO CONTINUE"
2303 W$ = INKEY$ : IF W$ = "" THEN 2303
2310 IF DIFF < DELTAS1 THEN 2340
2315 IF LDIST < REFSUM THEN 2530
2320 IF DIFF > LASTDIFF AND LASTDIFF > 0 THEN 2430
2330 LASTDIFF = DIFF : GOTO 2270
2340 IF DEBUG$ = "T" THEN 2350 ELSE 2360
2350 PRINT "ASSIGNING STOP NUMBER AND MATCH VALUE TO STOPS"
2360 ' ASSIGN MATCHVALUE OF 2 FOR GOOD MATCHING(DIFF<DELTAS1)
2370 STOPNUM(I) = J : JAY = J : MATCH(I) = 2 : NOTWOS = NOTWOS + 1
2371 IF DEBUG$ = "T" THEN 2372 ELSE 2380
2372 PRINT "MATCHVALUE = ": MATCH(I)
2373 PRINT "STOPNUMBER(I) = ": STOPNUM(I)
2374 PRINT "JAY = ": JAY
2380 IF STOPNUM(I) = NOSTOPS THEN STOPNUM(I) = 1
2385 ' ASSIGN MATCHVALUE OF 4 FOR TIGHTER MATCHING(DIFF<DELTAS2)
2390 IF ABS(LDIST - REFSUM) < DELTAS2 THEN MATCH(I) = 4 : NOFOURS = NOFOURS + 1
2391 IF DEBUG$ = "T" THEN 2392 ELSE 2400
2392 PRINT "MATCHVALUE = ": MATCH(I)
2400 LDIST = 0 : REFSUM = 0 : LASTDIFF = 0
2410 GOTO 2470
2420 ' ASSIGN TIMES FOR ACTIVITES AT TIMEPOINTS.
2430 STOPNUM(I) = J - 1 : MATCH(I) = 0 : JAY = J - 1 : LASTDIFF = 0
2440 NOZEROS = NOZEROS + 1 : IF DEBUG$ = "T" THEN 2450 ELSE 2460
2450 PRINT "ASSIGNING SCHEDTIME TO ACTIVITIES AT TIMEPOINTS"
2451 PRINT : PRINT "HIT ANY KEY TO FINISH"
2452 W$ = INKEY$ : IF W$ = "" THEN 2452
2460 ' ASSIGN TIMES FOR ACTIVITES AT TIMEPOINTS.
2470 FOR K = 1 TO TIMEPTS

Appendix D. PROGRAM LISTINGS
2480 IF STOPNUM(I) = TIMEPT(K) THEN 2490 ELSE 2520
2490 SCHEDTIME(I) = SCHED(L,K)
2491 IF DEBUG$ = "T" THEN 2492 ELSE 2500
2492 PRINT "SCHEDTIME(I) =";SCHEDTIME(I)
2493 PRINT : PRINT "HIT ANY KEY"
2494 W$ = INKEY$ : IF W$ = "" THEN 2494
2500 IF SCHEDTIME(I) = 0 AND STOPNUM(I) = 1 THEN SCHED(L,K) = SCHED(L+1,1)
2510 SCHEDTIME(I) = SCHED(L,K)
2520 NEXT K
2530 NEXT I
2540 NEXT L
2541 '
2542 '
2550 GOTO 3340
2560 '-------back match the stray data ---------------------------------------------
2570 'LOOPS IS THE NUMBER OF LOOPS FOR THE WHOLE DAY
2580 FOR L = 1 TO LOOPS
2590 IF BEGLOOP(L) = 0 THEN 2930
2600 '
2610 ' match activities to stops
2620 '
2630 J = NOSTOPS : REFSUM = 0 : BDIST = 0 : LASTDIFF = 0
2635 'START AT BEGINNING OF A GOOD LOOP AND GO BACK TILL 2ND RECORD IN DATA FILE.
2636 'BECAUSE 1ST RECORD IN DATA FILE IS A LOG 1 INDICATING POWER ON.
2640 FOR I = BEGLOOP(L) TO 2 STEP -1
2650 GOTO 2680
2660 J = J - 1 : IF J = 0 THEN J = NOSTOPS
2667 REFSUM = REFSUM + DFLS(J) :GOTO 2690
2670 'BDIST IS BACK DISTANCE.
2680 BDIST = BDIST + LSTDIST(I)
2690 DIFF = ABS(BDIST-REFSUM)
2700 IF STOPNUM(I-1) > 0 AND I-1 < BEGLOOP(L) THEN 2930
2710 IF ACTTIME(I) < STARTTIME - DELTAT/2 THEN 2950
2720 IF ABS(BDIST - REFSUM) < DELTAS1 THEN 2760
2730 IF DIFF > LASTDIFF AND LASTDIFF > 0 THEN 2820
2740 LASTDIFF = DIFF : GOTO 2660
2750 ' assign stop number and match value to activity i
2760 ' JAY = J - 1 : LASTDIFF = 0
2770 STOPNUM(I-1) = JAY : MATCH(I-1) = 1 : REFSUM = 0 : BDIST = 0
2780 GOTO 2840
2810 ' STOPNUM(I-1) = J : MATCH(I-1) = 0 : JAY = J : LASTDIFF = 0
2830 ' IF STOPNUM(I) = NOSTOPS THEN STOPNUM(I) = 1
2850 ' assign schedtime to activity i if j is a timepoint
2860 ' FOR K = 1 TO TIMEPTS
2880 IF JAY = TIMEPT(K) THEN 2890 ELSE 2910
2890 IF L = 1 THEN 2920
2900 SCHEDTIME(I-1) = SCHED(L-1,K)
2910 NEXT K
2920 NEXT I
2930 NEXT L
2940 '------------------front match remaining stray data------------------------
2950 ' FOR L = 1 TO LOOPS - 1
2970 FOR I = ENDLOOP(L) TO ACTNO
2990 ' match activities to stops
3000 ' IF ENDLOOP(L) = 0 THEN 3320
3020 J = 1 : REFSUM = 0 : FDIST = 0 : LASTDIFF = 0
3030 FOR I = ENDLOOP(L) TO ACTNO
3050 IF I = 0 OR I = ACTNO THEN 3320 ELSE 3070
3060 J = J + 1 : IF J > NOSTOPS THEN J = 1
3080 REFSUM = REFSUM + DFLS(J) : GOTO 3080

Appendix D. PROGRAM LISTINGS 203
3070 FDIST = FDIST + LSTDIST(I)
3080 DIFF = ABS(FDIST - REFSUH)
3090 IF ACTTIME(I) > ENDTIME + DELTAT/2 THEN 3340
3100 IF I > ENDOLOOP(L) AND MATCH(I) > 0 THEN 3320
3110 IF DIFF < DELTAS1 THEN 3150
3120 IF DIFF > LASTDIFF AND LASTDIFF > 0 THEN 3220
3130 LASTDIFF = DIFF : GOTO 3050
3140 ' assign stop number and match value to activity I
3150 ' assign schedtime to activity I if stop j is a timepoint
3160 JAY = J : LASTDIFF = 0
3170 STOPNUM(I) = J : MATCH(I) = 3 : REFSUH = 0 : FDIST = 0
3180 IF STOPNUM(I) = NOSTOPS THEN STOPNUM(I) = 1
3190 GOTO 3240
3200 ' assign schedtime to activity I if stop j is a timepoint
3210 STOPNUM(I) = J - 1 : MATCH(I) = 0 : JAY = J - 1 : LASTDIFF = 0
3220 FOR K = 1 TO TIMEPTS
3230 IF JAY = TIMEPT(K) THEN 3290 ELSE 3300
3240 IF L = LOOPS THEN 3310
3250 SCHEDTIME(I) = SCHED(L+1,K)
3260 NEXT K
3270 NEXT I
3280 NEXT L
3290 ' store the output file
3300 ADD DRIVE AND EXTENSION TO THE FILENAME.
3310 FS = OUTDRVS + RANDATAS + ";.OUT"
3320 LOCATE 13,15,0
3330 PRINT "Matching complete - Storing ";FS
3340 OPEN FS FOR OUTPUT AS 1
3350 WRITE #1,ROUTENO,ROUTENAMES,NOSTOPS,TRIPDIST,LOOPDIST,TURNPOINT

Appendix D. PROGRAM LISTINGS
3410 WRITE #1,ACTNO,DATADATE$,BLOCK,DAYOFWK,BUSID,SEATS,CAP,CLICK
3415 WRITE #1,TOTDIST,TOTTIME
3420 FOR I = 1 TO ACTNO
3430 WRITE #1,ACTTYPE(I),STOPNUM(I),ACTDIST(I),ACTTIME(I),SCHEDTIME(I),
   ACTONS(I),ACTOFFS(I),PASSLOAD(I),MATCH(I)
3440 NEXT I
3450 CLOSE #1
3455 'GO BACK TO MENU.
3460 GOTO 190
3470 ' 
3480 ' 
3490 ' <<<<<<<< PART 2 - GENERATE BUS STOP LISTING >>>>>>>>>>>>>>>>>>>>>>>>>>>>
3500 ' 
3510 ' 
3520 CLS
3530 'DRAW WINDOWS
3535 LINE (40,50) - (575,120),1,8
3536 LINE (38,48) - (577,122),1,8
3540 LOCATE 10,10,1
3550 PRINT "For which data file do you want the bus stop listing ?"
3560 LOCATE 11,10,1
3570 INPUT "(remember - the raw data must be processed first) ",L$
3580 LOCATE 13,10,1
3590 INPUT "Do you prefer a screen display (S) or a printed copy (P) "; C$
3600 IF C$ = "S" OR C$ = "s" THEN 3640
3610 IF C$ = "P" OR C$ = "p" THEN 3640
3620 LOCATE 15,10,1 :PRINT "INVALID CHOICE - SELECT EITHER S OR P - TRY AGAIN
3630 GOTO 3580
3635 'RETRIEVE OUTPUT FILE.
3640 F$ = OUTDRVS + L$ + ".OUT"
3650 OPEN F$ FOR INPUT AS #1
3660 INPUT #1,ROUTENO,ROUTENAMES,NOSTOPS,TRIPDIST,LOOPDIST,TURNPOINT
3670 INPUT #1,ACTNO,DATADATE$,BLOCK,DAYOFWK,BUSID,SEATS,CAP,CLICK

Appendix D. PROGRAM LISTINGS 205
3675 INPUT #1, TOTDIST, TOTTIME
3680 DIM ACTTYPE1(ACTNO), STOPNUM1(ACTNO), ACTDIST1(ACTNO), ACTTIME1(ACTNO),
       SCHEDTIME1(ACTNO), ACTONS1(ACTNO), ACTOFFS1(ACTNO), PASSLOAD1(ACTNO), MATCH1(ACTNO)
3690 FOR I = 1 TO ACTNO
3700 INPUT #1, ACTTYPE1(I), STOPNUM1(I), ACTDIST1(I), ACTTIME1(I), SCHEDTIME1(I),
       ACTONS1(I), ACTOFFS1(I), PASSLOAD1(I), MATCH1(I)
3710 NEXT I
3720 CLOSE #1
3725 "RETRIEVE NAMES REFERENCE FILE.
3730 N$ = NAMEDRV$ + "NAMES" + RIGHT$(STR$(ROUTENO), LEN(STR$(ROUTENO)) - 1 )
     + ".REF"
3735 CLS: LINE (80,50) - (575,180),1,B
3737 LINE (78,48) - (577,182),1,B
3740 LOCATE 13,15,0 :PRINT "The reference file will be "; N$
3750 LOCATE 14,15,0 :INPUT "Is this the file you wish (Y or N)? ", R$
3760 IF R$ = "y" OR R$ = "Y" THEN 3810
3770 IF R$ = "n" OR R$ = "N" THEN 3790
3780 LOCATE 18,15,0 :PRINT "INVALID CHOICE - CHOOSE EITHER Y OR N -
     TRY AGAIN" : GOTO 3750
3790 LOCATE 20,15,0 :PRINT "What file do you prefer?"
3800 LOCATE 21,15,0 :INPUT "(Remember to indicate the drive and extension) ", N$
3810 OPEN N$ FOR INPUT AS #2
3820 INPUT #2, ROUTENO, ROUTENAME$, NOSTOPS
3830 DIM STOPNAME$(NOSTOPS)
3840 FOR J = 1 TO NOSTOPS
3850 INPUT #2, STOPNAME$(J)
3860 NEXT J
3870 CLOSE #2
3880 "
3890 "----------prepare and print bus stop listing-----------------------------
3900 CLS
3910 "FOR PRINTING ON THE SCREEN
3920 IF C$ = "s" OR C$ = "S" THEN 3930 ELSE 4460
3930 CLS

Appendix D. PROGRAM LISTINGS 206
3940 LOCATE 1,3,1
3950 PRINT SYST$;"":PRINT "BUS STOP LISTING"
3960 LOCATE 1,44,1
3970 PRINT ROUTENAME$;"":PRINT DATADATES
3980 LOCATE 3,1,1
3990 PRINT "STOP# STOP NAME M DIST DEPART SCHED DEV ONS OFFS PASS"
4000 LNUM = 5
4010 PRINT
4020 PAGE = PAGE + 1
4030 IF PAGE > 1 THEN 4300
4040 FOR I = 1 TO ACTNO
4050 IF ACTTYPE1(I) = 1 OR ACTTYPE1(I) = 3 THEN 4410
4060 IF ACTTYPE1(I) = 6 OR ACTTYPE1(I) = 11 THEN 4410
4070 J = STOPNUM1(I)
4080 IF J = 0 THEN SNAME$ = "" ELSE SNAME$ = STOPNAME$(J)
4090 REALOIST = ACTDIST1(I)/5280
4100 ATIME = ACTTIME1(I)
4110 GOSUB 4940 : REM TIME CONVERSION SUBROUTINE
4120 REALTIME$ = TEMPS
4130 IF SCHEDTIME1(I) = 0 THEN 4210
4140 ATIME = SCHEDTIME1(I)
4150 GOSUB 4940 : REM TIME CONVERSION SUBROUTINE
4160 REALSCHED$ = TEMPS
4170 ATIME = SCHEDTIME1(I) - ACTTIME1(I)
4180 GOSUB 4940 : REM TIME CONVERSION SUBROUTINE
4190 SCHEDDEVS = TEMPS
4200 GOTO 4220
4210 REALSCHED$ = "":SCHEDDEVS = ""
4220 'NUMBER OF LINES = LNUM.
4230 LNUM = LNUM + 1
4250 W$ = ""

Appendix D. PROGRAM LISTINGS
LOCATE 25,5,0: PRINT "Strike Q to quit - strike any other key to continue";
W$ = INKEY$: IF W$ = "" THEN 4270
IF W$ = "q" OR W$ = "Q" THEN 4450
GOTO 3930
IF STOPNUM1(I) > 0 THEN 4310 ELSE 4320
PRINT USING "##8";STOPNUM1(I); PRINT " ": GOTO 4330
PRINT " * 
PRINT USING " ";SNAMES;
PRINT USING "##$";MATCH1(I);
PRINT " "; PRINT USING "###.";REALDIST; PRINT " ";
PRINT USING " ";REALTIME$;REALSCHED$;
PRINT USING " ";SCHEDDEV$;
PRINT USING "###";ACTONS1(I);
PRINT USING "###";ACTOFFS1(I);
PRINT USING "###";PASSLOAD1(I)
NEXT I
PRINT : PRINT "Strike any key to return to menu."
W$ = ""
W$ = INKEY$: IF W$ = "" THEN 4440
GOTO 190 : REM RETURN TO MENU
"--------print hardcopy of bus stop listing-------------------------
CLS: LOCATE 13,15,0
PRINT "Ready printer - strike any key to begin printing "
P$ = ""
P$ = INKEY$: IF P$ = "" THEN 4500
PAGE = PAGE + 1
LPRINT : LPRINT : LPRINT: LPRINT: LPRINT
LPRINT TAB(3) SYST$: LPRINT TAB(30) "BUS STOP LISTING ": LPRINT TAB(50)
"PREPARED ": LPRINT TAB(71) "PAGE ": PAGE
LPRINT TAB(3) "BLOCK ": LPRINT (15) "ROUTE ": ROUTENO 
LPRINT TAB(30) ROUTENAMES$: LPRINT TAB(71) DATEDATE$: LPRINT
LPRINT "STOP# STOP NAME M DIST DEPART SCHED DEV ONS 
OFFS PASS" : LPRINT
LNUM = 9
4570 IF PAGE > 1 THEN 4800
4580 FOR I = 1 TO ACTNO
4590 IF ACTTYPE1(I) = 1 OR ACTTYPE1(I) = 3 THEN 4910
4600 IF ACTTYPE1(I) = 6 OR ACTTYPE1(I) = 11 THEN 4910
4610 J = STOPNUM1(I)
4620 IF J = 0 THEN SNAME$ = " " ELSE SNAME$ = STOPNAME$(J)
4625 'CONVERT DIST FROM ODOMETER CLICKS TO MILES.
4630 REALDIST = ACTDIST1(I)* CLICK/5280
4640 ATIME = ACTTIME1(I)
4650 GOSUB 4940 : REM TIME CONVERSION SUBROUTINE.
4660 REALTIME$ = TEMP$
4670 IF SCHEDTIME1(I) = 0 THEN 4760
4680 ATIME = SCHEDTIME1(I)
4690 GOSUB 4940 : REM TIME CONVERSION SUBROUTINE.
4700 REALSCHED$ = TEMP$
4710 ATIME = SCHEDTIME1(I) - ACTTIME1(I)
4720 GOSUB 4940 : REM TIME CONVERSION SUBROUTINE.
4730 SCHEDDEV$ = TEMP$
4740 GOTO 4770
4750 'CONVERT DIST FROM ODOMETER CLICKS TO MILES.
4760 REALDIST = ACTDIST1(I)* CLICK/5280
4770 LNUM = LNUM + 1
4775 'PRINT 60 LINES AT A TIME PER PAGE
4780 IF LNUM > 60 THEN 4790 ELSE 4800
4800 IF STOPNUM1(I) > 0 THEN 4810 ELSE 4820
4810 LPRINT USING "###";STOPNUM1(I); LPRINT " "; GOTO 4830
4820 LPRINT " ";
4830 LPRINT USING " ";SNAME$;
4840 LPRINT USING"###";MATCH1(I);
4850 LPRINT "; LPRINT USING "###.###";REALDIST; LPRINT " ";
4860 LPRINT USING" ";REALTIME$;REALSCHED$;
4870 LPRINT USING" ";SCHEDDEV$;
4880 LPRINT USING"###";ACTONS1(I);
4890 LPRINT USING "####";ACTOFFS1(I);
4900 LPRINT USING "####";PASSLOAD1(I)
4910 NEXT I
4920 GOTO 190 : REM RETURN TO MENU
4930 STOP
4940 '----------TIME CONVERSION SUBROUTINE-----------------------------
4950 ATIME = ATIME + .001
4960 SIGN = SGN(ATIME)
4970 ATIME = ABS(ATIME)
4980 HR = FIX(ATIME/240)
4990 MINIT = FIX(((ATIME/240) - HR) * 60)
5000 SEC = FIX(ABS(60 * ((ATIME/240 - HR) * 60 - MINIT)) + .2)
5010 IF SEC = 60 THEN 5020 ELSE 5050
5020 SEC = 0 : IF MINIT < 0 THEN MINIT = MINIT - 1 ELSE MINIT = MINIT + 1
5030 IF MINIT = 60 THEN 5040 ELSE 5050
5040 MINIT = 0 : HR = HR + 1
5050 IF HR = 0 THEN 5060 ELSE 5090
5060 TEMP$ = RIGHT$(STR$(MINIT),2) + "":"" + RIGHT$(STR$(SEC),2)
5070 IF SIGN = -1 THEN TEMP$ = "-" + TEMP$ ELSE TEMP$ = " " + TEMP$
5080 GOTO 5100
5090 TEMP$ = RIGHT$(STR$(HR),2) + "":"" + RIGHT$(STR$(MINIT),2) + ":"
 + RIGHT$(STR$(SEC),2)
5100 RETURN
5110 STOP
5120 CLS :END
I
I
10 ' <<< REPORTS >>>
20 '
30 ' WRITTEN BY PAUL ANDERSON & SRINATH RAJU
40 '
50 CLS : SCREEN 2
52 LINE (60,30) - (590,175),1,B
54 LINE (57,27) - (593,178),1,B
60 '------retrieve system parameters----------------------------------------
70 '
80 OPTION BASE 1
90 OPEN "SYSTINFO.EXT" FOR INPUT AS 1
100 INPUT #1,SYST$
110 INPUT #1,DISTBASE$,BLOCKBASE$,NAMEBASE$,SECT15BASE$
120 INPUT #1,DATADRVS,DISTDRV$,NAMEDRV$,BLOCKDRV$,OUTDRV$,SECT15DRV$
130 INPUT #1,DELTAL,DELTAT,DELTAS1,DELTAS2
140 INPUT #1,CLICK,AF,SEATS,CAP
150 CLOSE #1
160 '
170 '----------------------------program menu--------------------------------
180 '
190 KEY OFF
200 'CLS
210 'COLOR 2,0,0
220 LOCATE 6,1,0
230 PRINT TAB(20) SYST$, " APC-REPORT PROGRAM MENU" : PRINT
240 PRINT TAB(20) "Please choose one of the following: " :PRINT
250 PRINT TAB(20) "1 - GENERATE TIME POINT PROFILE REPORT" :PRINT
260 PRINT TAB(20) "2 - PRODUCE SECTION 15 DAILY RECORD FILE":PRINT
270 PRINT TAB(20) "3 - UPDATE SECTION 15 ANNUAL REPORT FILE":PRINT
275 PRINT TAB(20) "4 - GENERATE SUMMARY REPORT FILE":PRINT
280 PRINT TAB(20) "5 - QUIT (EXIT TO DOS)" :PRINT
290 LOCATE 19,1,1
300 INPUT " What is the number of your choice ":CHOICE
310 IF CHOICE < 1 OR CHOICE > 5 GOTO 320 ELSE 340

Appendix D. PROGRAM LISTINGS 211
320 PRINT "Invalid choice - try again " : GOTO 300

330'

340 ON CHOICE GOSUB 360,2200,4090,6240,7900

350'

360'<<<<<<< PART 1 - TIME POINT / IDLE TIME REPORT >>>>>>>>>>>>>>>>>>>>>>

370'

380'---------------retrieve output file-----------------------------------------

385 CLS

390 LINE (25,60) - (550,150),1,B

395 LINE (22,57) - (553,153),1,B

400 LOCATE 13,5,0

410 INPUT "For which output file do you want the time point file ? ",F$

420 LOCATE 15,5,0 : INPUT "Do you prefer a screen display (S) or a printed copy (P)? ",C$

430 IF C$ = "S" OR C$ = "S" THEN 460

440 IF C$ = "P" OR C$ = "P" THEN 460

450 LOCATE 17,5,0 :PRINT "INVALID INPUT - CHOOSE EITHER S OR P - TRY AGAIN ": GOTO 420

460'

470 F$ = OUTDRV$ + F$ + ".OUT"

475 CLS : LINE (70,70) - (500,120),1,B

476 LINE (67,67) - (503,123),1,B

480 LOCATE 13,25,0 :PRINT "Retrieving ";F$

490 OPEN F$ FOR INPUT AS #1

500 INPUT #1,ROUTENO,ROUTENAME$,NOSTOPS,TRIPDIST,LOOPDIST,TURNPOINT

510 INPUT #1,ACTNO,DATADATE$,8LOCK,DAYOFWK,BUSID,SEATS,CAP,CLICK

515 INPUT #1,TOTDIST,TOTTIME

520 DIM ACTTYPE(ACTNO),STOPNUM(ACTNO),ACTDIST(ACTNO),ACTTIME(ACTNO),

      SCHEDTIME(ACTNO),ACTONS(ACTNO),ACTOFFS(ACTNO),PASSLOAD(ACTNO),MATCH(ACTNO)

530 FOR I = 1 TO ACTNO

540   INPUT #1,ACTTYPE(I),STOPNUM(I),ACTDIST(I),ACTTIME(I),SCHEDTIME(I),

      ACTONS(I),ACTOFFS(I),PASSLOAD(I),MATCH(I)

550 NEXT I

560 CLOSE #1

570'

580'---------------retrieve names reference files-------------------------------
590 '  
600 NS = NAMEDRV$ + "NAMES" + RIGHT$(STR$(ROUTENO),LEN(STR$(ROUTENO))-1)+".REF"
603 CLS : LINE (60,80) - (550,170),1,B
605 LINE (57,77) - (553,173),1,B
610 LOCATE 13,15,0 : PRINT "The reference file will be ";N$
630 LOCATE 15,15,0 : INPUT "Is this the file you wish (Y or N) ? ",R$
640 IF R$ = "y" OR R$ = "Y" THEN 700
650 IF R$ = "n" OR R$ = "N" THEN 670
660 LOCATE 18,15,0 :PRINT "INVALID CHOICE - CHOOSE EITHER Y OR N - TRY AGAIN" :
670 GOTO 620
680 LOCATE 18,15,0 : PRINT "(Remember to indicate the drive and extension) ",N$
690 '  
700 '  
710 IF C$ = "p" OR C$ = "P" THEN 720 ELSE 750
720 CLS: LOCATE 13,15,0
725 LINE (60,60) - (550,100),1,B
726 LINE (57,57) - (553,103),1,B
728 PRINT "Ready printer - strike any key to begin printing ":
730 WS = ""
740 WS = INKEY$ : IF WS="" THEN 740
750 OPEN N$ FOR INPUT AS 83
760 INPUT #3,ROUTENO,ROUTENAMES$,NOSTOPS
770 DIM STOPNAME$(NOSTOPS)
780 FOR J = 1 TO NOSTOPS
790 INPUT #3,STOPNAME$(J)
800 NEXT J
810 CLOSE #3
820 '  
830 '----------accumulate statistics between the stops -----------------------
840 '  
850 I = 1 : N = 1
860 I = I + 1
870 IF I + 1 = ACTNO THEN 1970

Appendix D. PROGRAM LISTINGS

213
880 ' 
890 ONSBET = ONSBET + ACTONS(I) 
900 OFFSBET = OFFSBET + ACTOFFS(I) 
910 STOPSBET = STOPSBET + 1 
920 IF PASSLOAD(I) > MAXLOAD THEN 930 ELSE 940 
930 MAXLOAD = PASSLOAD(I) ; MAXSTOP = STOPNUM(I) 
940 IF SCHEDTIME(I) > 0 AND STOPNUM(I+1) <> STOPNUM(I) THEN 950 ELSE 860 
950 ' 
960 '----------compute and print timepoint information--------------------------' 
970 ' 
980 ' convert times to hours,minutes,seconds 
990 ATIME = ACTTIME(I) 
1000 GOSUB 2020 
1010 ARRIVE$ = TEMP$ 
1020 ATIME = SCHEDTIME(I) 
1030 GOSUB 2020 
1040 SCHEDTIME$ = TEMP$ 
1050 ATIME = SCHEDTIME(I) - ACTTIME(I) 
1060 GOSUB 2020 
1070 SCHEDDEV$ = TEMP$ 
1080 IF X = 0 THEN 1460 
1090 ATIME = ACTTIME(I) - LASTTIME : TIMEBET = ATIME 
1100 GOSUB 2020 
1110 TIMEBET$ = RIGHT$(TEMP$,5) 
1120 ' 
1130 DISTBET = (ACTDIST(I) - LASTDIST) * CLICK /5280 
1140 SPEED = DISTBET/(TIMEBET/240) 
1150 IF STOPNUM(I) = 0 THEN 1160 ELSE 1170 
1160 SNAME$ = " " : GOTO 1190 
1170 SNAME$ = STOPNAME$(STOPNUM(I)) 
1180 IF C$ = "p" OR C$ = "P" THEN 1330 
1190 ' 
1200 PRINT " " ; 
1210 PRINT USING "## landscaping or OFFSBET; 

Appendix D. PROGRAM LISTINGS
1220 PRINT USING "********";MAXLOAD;MAXSTOP; TIMEBET$;
1230 PRINT USING "***********.**";DISTBET; MAXTIHE; TIMEBET$;
1240 PRINT USING "***********.**";SPEED
1250 IF STOPNUM(I) = 0 THEN 1260 ELSE 1270
1260 PRINT " "; GOTO 1280
1270 PRINT STOPNUM(I);
1280 PRINT TAB(6) " ";PRINT USING " ";SNAME$; PRINT " ";
1290 PRINT USING "***";PASSLOAD(I) ";PRINT " ";
1300 PRINT USING " ";ARRIVE$;SCHEDTIME$;
1310 PRINT USING " ";SCHEDDEV$
1320 GOTO 1460
1330 LPRINT " ";
1340 LPRINT USING "*****";ONSBET;OFFSBET;
1350 LPRINT USING "*****";MAXLOAD;MAXSTOP; LPRINT USING "******";STOPSDET;
1360 LPRINT USING "***********.**";DISTBET; LPRINT TAB(62) TIMEBET$ ;
1370 LPRINT USING "***********.**";SPEED
1380 IF STOPNUM(I) = 0 THEN 1390 ELSE 1400
1390 LPRINT " "; GOTO 1410
1400 LPRINT STOPNUM(I);
1410 LPRINT TAB(6) " ";LPRINT USING " ";SNAME$; LPRINT " ";
1420 LPRINT USING "***";PASSLOAD(I) ";LPRINT " ";
1430 LPRINT USING " ";ARRIVE$;SCHEDTIME$;
1440 LPRINT USING " ";SCHEDDEV$
1450 ' 
1460 '-------reset values------------------------------------------
1470 ' 
1480 LASTDIST = ACTDIST(I) : LASTTIME = ACTTIME(I) : MAXLOAD = PASSLOAD(I)
1490 MAXSTOP = STOPNUM(I) : ONSBET = 0 : OFFSBET = 0 : STOPSBET = 0
1500 TIMEBET = 0 : DISTBET = 0 : SPEED = 0
1510 ' 
1520 IF X = 0 THEN 1620
1530 IF STOPNUM(I) = 1 THEN 1560 ELSE 860
1540 ' 
1550 ' 

Appendix D. PROGRAM LISTINGS
1560 IF X = 0 THEN 1630
1570 IF C$ = "P" OR C$ = "p" THEN 1620
1580 LOCATE 25,5,0 : PRINT "Strike Q to quit - strike any other key to continue." ;
1590 W$ = ""
1600 W$ = INKEY$ : IF W$ = "" THEN 1600
1610 IF W$ = "Q" OR W$ = "q" THEN 50
1620 X = X + 1
1630 IF C$ = "p" OR C$ = "P" THEN 1790
1640 CLS:PRINT
1650 PRINT TAB(2) SYST$;PRINT TAB(28) "TIME POINT PROFILE REPORT";
   PRINT TAB(60) "PREPARED "DATE$.
1660 PRINT " BLOCK #";BLOCK; " ROUTE " ROUTENO ; PRINT TAB(28) ROUTENAME$ ;
   PRINT TAB(59) DATADATE$ .
1670 PRINT: PRINT " STOP# STOP NAME LOAD DEPART SCHED DEV "
1680 LOCATE 6,2,0 : PRINT " ONS OFFS MAXLOAD MAXSTOP # STOPS DIST TIME SPEED" : PRINT
1690 IF STOPNUM(I) = 0 THEN 1700 ELSE 1720
1700 PRINT "; GOTO 1740
1710 SNAME$ = " 
1720 PRINT STOPNUM(I);
1730 SNAME$ = STOPNAME$(STOPNUM(I))
1740 PRINT TAB(6) " ";PRINT USING " ;SNAME$; PRINT " ";
1750 PRINT USING "####";PASSLOAD(I); PRINT " ";
1760 PRINT USING " " ;ARRIVE$;SCHEDTIME$;
1770 PRINT USING " " ;SCHEDDEV$
1780 GOTO 1940
1790 IF X > 1 THEN 1830
1800 LPRINT : LPRINT : LPRINT : LPRINT
1810 LPRINT TAB(2) SYST$;LPRINT TAB(28) "TIME POINT PROFILE REPORT";
   LPRINT TAB(60) "PREPARED "DATE$.
1820 LPRINT " BLOCK #";BLOCK; " ROUTE " ROUTENO ; LPRINT TAB(28) ROUTENAME$ ;
   LPRINT TAB(60) DATADATE$ .
1830 LPRINT:LPRINT:LPRINT: LPRINT " STOP# STOP NAME LOAD

Appendix D. PROGRAM LISTINGS
DEPART SCHED DEV

1840 LPRINT " ONS OFFS MAXLOAD MAXSTOP # STOPS DIST

TIME SPEED" : LPRINT

1850 IF STOPNUM(I) = 0 THEN 1860 ELSE 1870

1860 LPRINT " ;; GOTO 1880

1870 SNAME$ = " 

1880 LPRINT STOPNUM(I); 

1890 SNAME$ = STOPNAME$(STOPNUM(I))

1900 LPRINT TAB(6) " ;;LPRINT USING " ";SNAME$;

LPRINT " ";

1910 LPRINT USING "###";PASSLOAD(I) ;;LPRINT " ";

1920 LPRINT USING " ";ARRIVE$;SCHEDTIME$;

1930 LPRINT USING " ";SCHEDDEV$

1940 X = X + 1

1950 GOTO 860

1960 '

1970 LOCATE 25,15,0 : PRINT "Strike any key to return to menu."

1980 W$ = ""

1990 W$ = INKEY$ : IF W$ = "" THEN 1990

2000 GOTO 50

2010 '

2020 '--------TIME CONVERSION SUBROUTINE-----------------------------

2030 '

2040 ATIME = ATIME + .001

2050 SIGN = SGN(ATIME)

2060 ATIME = ABS(ATIME)

2070 HR = FIX(ATIME/240)

2080 MINIT = FIX((ATIME/240 - HR) * 60 )

2090 SEC = FIX((60 * (((ATIME/240 - HR) * 60) - MINIT)) + .2)

2100 IF SEC = 60 THEN 2110 ELSE 2140

2110 SEC = 0 : IF MINIT < 0 THEN MINIT = MINIT - 1 ELSE MINIT = MINIT + 1

2120 IF MINIT = 60 THEN 2130 ELSE 2140

2130 MINIT = 0 : IF HR < 0 THEN HR = HR - 1 ELSE HR = HR + 1

2140 IF HR = 0 THEN 2150 ELSE 2180
2150 TEMPS = RIGHT$(STR$(MINIT),2) + ":" + RIGHT$(STR$(SEC),2)
2160 IF SIGN = -1 THEN TEMPS = "-" + TEMPS
2170 GOTO 2190
2180 TEMPS = RIGHT$(STR$(HR),2) + ":" + RIGHT$(STR$(MINIT),2) + ":" + RIGHT$(STR$(SEC),2)
2190 RETURN

' <<<<<<<<< PART 2 - PRODUCE SECTION 15 DAILY REPORT FILE >>>>>>>>>>>>>>>>>>

'--------RETRIEVE SECTION 15 PARAMETERS-----------------------'

2220 OPEN "sect15.ext" FOR INPUT AS #1
2230 INPUT #1,TRANSITID$,MODE$,LEVEL$,FISCALYR$
2240 INPUT #1,WDYBEGIN,AMPEAK,MIDDAY,PMPEAK,NIGHT,WDYEND
2250 INPUT #1,SATBEGIN,SATEND,SUNBEGIN,SUNEND
2260 CLOSE #1

'---retrieve processed data (output) file---------------------'

2310 INPUT "Which output file would you like to process ":OUTPT$
2320 F$ = OUTDRVS + OUTPT$ + ".OUT"
2330 CLS : LINE (80,80) - (575,140),1,B
2340 LOCATE 15,25,0 : PRINT "Retrieving ":F$
2350 OPEN F$ FOR INPUT AS #2
2360 INPUT #2,ROUTENO,ROUTENAME$,NOSTOPS,TRIPTDIST,LOOPTDIST,TURNPOINT
2370 INPUT #2,ACTNO,DATADATE$,BLOCK,DAYOFW,BUSID,SEATS,CAP,CLICK
2380 INPUT #2,TOTDIST,TOTTIME
2390 INPUT #2,ACTTYPE(ACTNO),STOPNUM(ACTNO),ACTDIST(ACTNO),ACTTIME(ACTNO),

Appendix D. PROGRAM LISTINGS 218
SCHEDTIME(ACTNO),ACTONS(ACTNO),ACTOFFS(ACTNO),PASSLOAD(ACTNO),MATCH(ACTNO)

2420 FOR I = 1 TO ACTNO
2430 INPUT #2, ACTTYPE(I), STOPNUM(I), ACTDIST(I), ACTTIME(I), SCHEDTIME(I),
ACTONS(I), ACTOFFS(I), PASSLOAD(I), MATCH(I)
2440 NEXT I
2450 CLOSE #2
2455 CLS
2456 LINE (100, 100) • (590, 160), 1, B
2456 LINE (97, 97) • (593, 163), 1, B
2460 LOCATE 17, 15, 0: PRINT "Creating the report - Please wait. " ;
2470 '
2480 '------------------set limits according to the day of the week---------------------
2490 ON DAYOFWK GOTO 2500, 2500, 2500, 2500, 2500, 2590, 2650
2500 ' weekdays
2510 L = 1
2520 LIHIT(1) = AMPEAK
2530 LIMIT(2) = MIDDAY
2540 LIMIT(3) = PMPEAK
2550 LIMIT(4) = NIGHT
2560 LIMIT(5) = WKYEND
2570 ENDLIMIT = 5
2580 GOTO 2710
2590 ' saturdays
2600 L = 5
2610 LIMIT(5) = SATBEGIN
2620 LIMIT(6) = SATEND
2630 ENDLIMIT = 6
2640 GOTO 2710
2650 ' sundays
2660 L = 6
2670 LIMIT(6) = SUNBEGIN
2680 LIMIT(7) = SUNEND
2690 ENDLIMIT = 7
2700 GOTO 2710
2710 '

Appendix D. PROGRAM LISTINGS
2720 '-------locate the trips within the time period-------------------
2730 I = 0
2740 I = I + 1
2750 IF I > ACTNO THEN 3170
2760 IF ACTTIME(I) < LIMIT(L) THEN 2790 ELSE 2770
2770 L = L + 1
2780 IF L = ENDLIMIT THEN 3170 ELSE 2740
2790 IF STOPNUM(I) = 1 OR STOPNUM(I) = TURNPOINT THEN 2800 ELSE 2740
2800 IF STOPNUM(I) = 1 THEN 2810 ELSE 2820
2810 REFDIST = TRIPDIST : TRIPEND = TURNPOINT : GOTO 2840
2820 IF STOPNUM(I) = TURNPOINT THEN 2830 ELSE 2740
2830 REFDIST = LOOPDIST - TRIPDIST : TRIPEND = 1
2840 BEGLOOP = I
2850 I = I + 1
2860 IF I > ACTNO THEN 3170
2870 TRIPSUM = ACTDIST(I) - ACTDIST(BEGLOOP)
2880 IF ABS(TRIPSUM - REFDIST) < DELTAL THEN 2920
2890 IF TRIPSUM < REFDIST THEN 2850
2900 I = BEGLOOP : GOTO 2740
2910 '  
2920 IF STOPNUM(I) = TRIPEND THEN 2930 ELSE 2850
2930 ENDLOOP = I
2940 '  
2950 '-------determine variable values for section 15 daily report--------
2960 '  
2970 NOTRIPS(L) = NOTRIPS(L) + 1
2980 FOR I = BEGLOOP TO ENDLOOP
2990 PASSONS(L) = PASSONS(L) + ACTONS(I)
3000 PASS = PASSLOAD(I)
3010 IF I = ENDLOOP THEN 3140
3020 PASSBOARD(L) = PASSBOARD(L) + PASS
3030 DIST = (ACTDIST(I+1) - ACTDIST(I)) * CLICK / 5280
3040 TDIST(L) = TDIST(L) + DIST
3050 TIME = (ACTTIME(I+1) - ACTTIME(I)) / 4

Appendix D. PROGRAM LISTINGS 220
3060  TTIME(L) = TTIME(L) + TIME
3070  PASSMILE(L) = PASSMILE(L) + PASS * DIST
3080  PASSMIN(L) = PASSMIN(L) + PASS * TIME / 4
3090  CAPMILE(L) = CAPMILE(L) + CAP * DIST
3100  SEATMILE(L) = SEATMILE(L) + SEATS * DIST
3110  UNPASSTRIP(L) = PASSONS(L)/NOTRIPS(L)
3120  PASSMITRIP(L) = PASSMILE(L)/NOTRIPS(L)
3130  UNPASSTIME(L) = PASSMIN(L)/NOTRIPS(L)
3140 NEXT I
3150 I = ENOLOOP - 1
3160 GOTO 2740
3170 '-------store the daily report file-------------------------------------
3180 '
3190 S$ = SECT15ORV$ + OUTPTS + ".15"
3200 PRINT " Storing ";S$
3210 OPEN S$ FOR OUTPUT AS #3
3220 FOR L = 2 TO 7
3230 WRITE #3,PASSONS(L),PASS8OARD(L),TDIST(L),PASSMILE(L),TTIME(L),
PASSMIN(L),CAPMILE(L),SEATMILE(L),NOTRIPS(L)
3240 NEXT L
3250 CLOSE #3
3260 '
3270 '-------print the daily report file-------------------------------------
3280 CLS
3285 LINE (25,25) - (592,195),1,8
3286 LINE (22,22) - (595,198),1,8
3290 LOCATE 9,10,0 : PRINT "The report will be displayed on the screen. "
3300 LOCATE 11,10,0 : PRINT "For a hard copy of the report, ready the printer -
3310 IF C$ = "s" OR C$ = "S" THEN 3350
3320 LOCATE 12,10,0 : PRINT "Then press the right shift key along with the PrtSc key. "
3330 LOCATE 14,10,0 : PRINT "When you have finished printing or viewing the report, "
3340 LOCATE 15,10,0 : PRINT "strike any key to return to the menu. "
3350 LOCATE 17,10,0 : PRINT "Strike any key to begin."
3360 '

Appendix D. PROGRAM LISTINGS
3370 '-----------------print to screen---------------------------------------------
3380 W$ = ""
3390 ' 3400 W$ = INKEY$ : IF W$ = "" THEN 3400
3410 CLS
3420 PRINT : PRINT : PRINT TAB(5) SYST$ ; PRINT TAB(28) "SECTION 15 DAILY REPORT" ;:
 PRINT TAB(60) "PREPARED " ; DATE$ 
3430 LOCATE 4,5,0 : PRINT "ROUTE # " ; ROUTENO ; PRINT " "; ROUTENAME$ ; 
3440 ON DAYOFWK GOTO 3450, 3460, 3470, 3480, 3490, 3500, 3510, 3520 
3450 PRINT TAB(60) "MONDAY " ; GOTO 3520
3460 PRINT TAB(60) "TUESDAY " ; GOTO 3520
3470 PRINT TAB(60) "WEDNESDAY " ; GOTO 3520
3480 PRINT TAB(60) "THURSDAY " ; GOTO 3520
3490 PRINT TAB(60) "FRIDAY " ; GOTO 3520
3500 PRINT TAB(60) "SATURDAY " ; GOTO 3520
3510 PRINT TAB(60) "SUNDAY " ; GOTO 3520
3520 PRINT TAB(70) DATADATES$ 
3530 LOCATE 5,5,0 : PRINT "BUS # " ; BUSID ; PRINT TAB(28) "SEATED CAPACITY " ; SEATS ; PRINT TAB(60) "TOTAL CAPACITY " ; CAP ; PRINT  
3540 LOCATE 7,40,0 : PRINT "TIME PERIOD" ; PRINT TAB(27) "AM PEAK " ; PRINT "MIDDAY " ; PRINT "PM PEAK " ; PRINT "NIGHT " ; PRINT "SATURDAY" ; PRINT " SUNDAY" 
3550 PRINT ; PRINT " Passengers Boarded " ; PRINT TAB(25) " " ;  
3560 FOR L = 2 TO 7  
3570 PRINT USING "#####" ; PASSES(L) ; PRINT " " ;  
3580 NEXT L 
3590 PRINT " Passengers On Board " ; PRINT TAB(25) " " ;  
3600 FOR L = 2 TO 7  
3610 PRINT USING "#####" ; PASSBOAR(L) ; PRINT " " ;  
3620 NEXT L 
3630 PRINT " Bus Trip Distance " ; PRINT TAB(25) " " ;  
3640 FOR L = 2 TO 7  
3650 PRINT USING "#####" ; TDIST(L) ; PRINT " " ;  
3660 NEXT L 

Appendix D. PROGRAM LISTINGS 222
3680 PRINT " Passenger Miles ": PRINT TAB(25) ";
3690 FOR L = 2 TO 7
3700 PRINT USING "########";PASSMILE(L): PRINT " ";
3710 NEXT L
3720 PRINT " Bus Trip Time ": PRINT TAB(25) ";
3730 FOR L = 2 TO 7
3740 PRINT USING "########";TIME(L): PRINT " ";
3750 NEXT L
3760 PRINT " Passenger Minutes ": PRINT TAB(25) ";
3770 FOR L = 2 TO 7
3780 PRINT USING "########";PASSMIN(L): PRINT " ";
3790 NEXT L
3800 PRINT " Capacity Miles ": PRINT TAB(25) ";
3810 FOR L = 2 TO 7
3820 PRINT USING "8##8#3##";CAPMILE(L): PRINT " ";
3830 NEXT L
3840 PRINT " Seat Miles ": PRINT TAB(25) ";
3850 FOR L = 2 TO 7
3860 PRINT USING "8##8#3##";SEATMILE(L): PRINT " ";
3870 NEXT L
3880 PRINT " Trips in Sample ": PRINT TAB(25) ";
3890 FOR L = 2 TO 7
3900 PRINT USING "一共##";NOTrips(L): PRINT " ";
3910 NEXT L
3920 PRINT " : PRINT " : SAMPLE AVERAGES":
3930 PRINT " Unlinked Pass/Trip ": PRINT TAB(25) " ";
3940 FOR L = 2 TO 7
3950 PRINT USING "一共##";UNPASSTRIP(L): PRINT " ";
3960 NEXT L
3970 PRINT " Pass. Miles/Trip ": PRINT TAB(25) " ";
3980 FOR L = 2 TO 7
3990 PRINT USING "一共##";PASSMITRIP(L): PRINT " ";
4000 NEXT L
4010 PRINT: PRINT " Unl. Pass. Trip Time": PRINT TAB(25) " ";
4020 FOR L = 2 TO 7
4030 PRINT USING "########";UNPASSTIHE(L) ;: PRINT " ";
4040 NEXT L
4050 WS = ""
4060 WS = INKEY$: IF WS = "" THEN 4060
4070 GOTO 50
4080 '        '  
4090 '<<<<<< PART 3 - UPDATE SECTION 15 CUMULATIVE DATA FILE >>>>>>>>
4100 '        
4110 ' '  
4120 '------retrieve section 15 parameters-----------------------
4130 '        
4140 OPEN "sect15.ext" FOR INPUT AS #1
4150 INPUT #1,TRANSITIDS,MODES,LEVELS,FISCALYRS
4160 INPUT #1,WKDBEGIN,AMPEAK,MIDDAY,PMPEAK,NIGHT,WKDAYEND
4170 INPUT #1,SATBEGIN,SATEND,SUNBEGIN,SUNEND
4180 CLOSE #1
4190 '        
4200 '------retrieve the section 15 daily report -----------------
4210 CLS
4215 LINE (30,30) - (550,100),1,B
4216 LINE (27,27) - (553,103),1,B
4220 LOCATE 13,5,1
4230 INPUT "Which section 15 daily report file do you want to add? ",SS
4240 IF SS = "NONE" OR SS = "none" THEN 4320
4250 FS = SECT15DRVS + SS + ".15"
4260 OPEN FS FOR INPUT AS #1
4270 FOR L = 2 TO 7
4280 INPUT #1,PASSONS(L),PASSBOARD(L),TDIST(L),PASSMILE(L),TTIME(L),
   PASSMIN(L),CAPMILE(L),SEATMILE(L),NOTRIPS(L)
4290 NEXT L
4300 CLOSE #1
4310 '  
4320 '------retrieve the section 15 cumulative data file-----------------

Appendix D. PROGRAM LISTINGS 224
I

4330 ' 
4340 C$ = SECT15DRV$ + SECT15BASE$ + ".15A"
4350 OPEN C$ FOR INPUT AS #2
4360 FOR L = 2 TO 7
4370 INPUT #2,CPASSONS(L),CPASSBOARD(L),CTDIST(L),CPASSMILE(L),CTTIME(L), 
   CPASSMIN(L),CCAPMILE(L),CSEATMILE(L),CNOTRIPS(L)
4380 NEXT L
4390 CLOSE #2
4400 '
4410 '·········compute and store the new values in the cumulative file·········
4420 '
4430 OPEN C$ FOR OUTPUT AS #3
4440 FOR L = 2 TO 7
4450 CPASSONS(L) = PASSONS(L) + CPASSONS(L)
4460 CPASSBOARD(L) = PASSBOARD(L) + CPASSBOARD(L)
4470 CTDIST(L) = TDIST(L) + CTDIST(L)
4480 CPASSMILE(L) = PASSMILE(L) + CPASSMILE(L)
4490 CTTIME(L) = TTIME(L) + CTTIME(L)
4500 CPASSMIN(L) = PASSMIN(L) + CPASSMIN(L)
4510 CCAPMILE(L) = CAPMILE(L) + CCAPMILE(L)
4520 CSEATMILE(L) = SEATMILE(L) + CSEATMILE(L)
4530 CNOTRIPS(L) = NOTRIPS(L) + CNOTRIPS(L)
4540 WRITE #3,CPASSONS(L),CPASSBOARD(L),CTDIST(L),CPASSMILE(L),CTTIME(L), 
   CPASSMIN(L),CCAPMILE(L),CSEATMILE(L),CNOTRIPS(L)
4550 NEXT L
4560 CLS
4565 LINE (60,60) - (595,175),1,8
4566 LINE (57,57) - (598,178),1,8
4570 LOCATE 15,15,0
4580 INPUT "Do you want to print the cumulative file (Y or N) ? ",P$
4590 IF P$ = "Y" OR P$ = "y" THEN 4620
4600 IF P$ = "n" OR P$ = "N" THEN 50
4610 LOCATE 15,15,0 : PRINT "INVALID INPUT - ENTER EITHER Y OR N - TRY AGAIN" :
GOTO 4570

4620 '  
4630 '----------calculate averages and annual totals-----------------------------
4640 '  
4650 LOCATE 17,15,1 : INPUT "Do you want to include the annual totals (Y or N) ? ", ANTOT$
4660 IF ANTOT$ = "Y" OR ANTOT$ = "y" THEN 4690
4670 IF ANTOT$ = "N" OR ANTOT$ = "n" THEN 4760
4680 LOCATE 15,15,0 : PRINT "INVALID INPUT - ENTER EITHER Y OR N - TRY AGAIN" : GOTO 4650
4690 CLS :
4695 LINE (30,30) - (590,180),1,B
4696 LINE (27,27) - (593,183),1,B
4700 LOCATE 6,10,1 : INPUT "What is the total number of trips in the a.m. peak ? ", TOTTRIPS(2)
4710 LOCATE 8,10,1 : INPUT "What is the total number of trips in the midday ? ", TOTTRIPS(3)
4720 LOCATE 10,10,1 : INPUT "What is the total number of trips in the p.m. peak ? ", TOTTRIPS(4)
4730 LOCATE 12,10,1 : INPUT "What is the total number of trips in the night ? ", TOTTRIPS(5)
4740 LOCATE 14,10,1 : INPUT "What is the total number of trips on Saturday ? ", TOTTRIPS(6)
4750 LOCATE 16,10,1 : INPUT "What is the total number of trips on Sunday ? ", TOTTRIPS(7)
4760 CLS
4770 FOR L = 2 TO 7
4780 IF CNOTRIPS(L) = 0 THEN 4790 ELSE 4800
4790 CUNPASS(L) = 0 : CPASSMITRIP(L) = 0 : CUNPASSTIME(L) = 0 : GOTO 4830
4800 CUNPASS(L) = CPASSONS(L)/CNOTRIPS(L)
4810 CPASSMITRIP(L) = CPASSMILE(L)/CNOTRIPS(L)
4820 CUNPASSTIME(L) = CPASSMIN(L)/CNOTRIPS(L)
4830 TOTUNPASS(L) = TOTTRIPS(L) * CUNPASS(L)
4840 TOTPASSMILE(L) = TOTTRIPS(L) * CPASSMITRIP(L)
4850 NEXT L
4860 ' 
4870 '--------print a hardcopy of the cumulative section 15 report----------
4880 ' 
4890 CLS
4900 LOCATE 13,15,0 : PRINT "Ready printer - strike any key when ready."
4910 W$ = ""
4920 W$ = INKEY$ : IF W$ = "" THEN 4920
4940 LPRINT TAB(5) SYST$ ; : LPRINT TAB(30) "SECTION 15 ANNUAL REPORT" ; : LPRINT
TAB(60) "PREPARED " ; DATE$ 
4950 LPRINT TAB(5) "TRANSIT ID " ; TRANSITID$ ; : LPRINT TAB(30) "NON-RAIL MODES" ; 
: LPRINT TAB(60) "MODE " ; MODE$ 
4960 LPRINT TAB(5) "FISCAL YEAR END " ; FISCALYR$ ; : LPRINT TAB(30) "FORM 406A" ; 
: LPRINT TAB(60) "LEVEL " ; LEVEL$ 
4970 : LPRINT : LPRINT
4980 LPRINT " TIME PERIOD " ; : LPRINT TAB(23) "AM PEAK " ; : LPRINT "MIDDAY " ; 
: LPRINT "PM PEAK " ; : LPRINT "NIGHT " ; : LPRINT "SATDAY " ; : LPRINT " SUNDAY " ; 
: LPRINT " TOTAL"
4990 LPRINT : LPRINT : LPRINT " Passengers Boarded " ; : LPRINT TAB(21) ""
5000 FOR L = 2 TO 7
5010 IF L < 5 THEN 5020 ELSE 5030
5020 LPRINT USING "###8###8#" ; CPASSONS(L) ; : GOTO 5040
5030 LPRINT USING "###8###8#" ; CPASSONS(L) ; 
5040 SUM = SUM + CPASSONS(L)
5050 NEXT L
5060 LPRINT USING "#########8" ; SUN$ ; : SUM = 0
5070 LPRINT : LPRINT " Pass. On Board" ; : LPRINT TAB(21) ""
5080 FOR L = 2 TO 7
5090 IF L < 5 THEN 5100 ELSE 5110
5100 LPRINT USING "#########8" ; CPASSBOARD(L) ; : GOTO 5120
5110 LPRINT USING "#########8" ; CPASSBOARD(L) ; 
5120 SUM = SUM + CPASSBOARD(L)
5130 NEXT L

Appendix D. PROGRAM LISTINGS
5140 LPRINT USING "##########";SUM; : SUM = 0
5150 LPRINT : LPRINT " Bus Trip Distance "; LPRINT TAB(21) " ";
5160 FOR L = 2 TO 7
5170 IF L < 5 THEN 5180 ELSE 5190
5180 LPRINT USING "##########";CTDIST(L) ; GOTO 5200
5190 LPRINT USING "##########";CTDIST(L) ;
5200 SUM = SUM + CTDIST(L)
5210 NEXT L
5220 LPRINT USING "##########";SUM; : SUM = 0
5230 LPRINT : LPRINT " Passenger Miles "; LPRINT TAB(21) " ";
5240 FOR L = 2 TO 7
5250 IF L < 5 THEN 5260 ELSE 5270
5260 LPRINT USING "##########";CPASSMILE(L) ; GOTO 5280
5270 LPRINT USING "##########";CPASSMILE(L) ;
5280 SUM = SUM + CPASSMILE(L)
5290 NEXT L
5300 LPRINT USING "##########";SUM; : SUM = 0
5310 LPRINT : LPRINT " Bus Trip Time "; LPRINT TAB(21) " ";
5320 FOR L = 2 TO 7
5330 IF L < 5 THEN 5340 ELSE 5350
5340 LPRINT USING "##########";CTTIME(L) ; GOTO 5360
5350 LPRINT USING "##########";CTTIME(L) ;
5360 SUM = SUM + CTTIME(L)
5370 NEXT L
5380 LPRINT USING "##########";SUM; : SUM = 0
5390 LPRINT : LPRINT " Passenger Minutes "; LPRINT TAB(21) " ";
5400 FOR L = 2 TO 7
5410 IF L < 5 THEN 5420 ELSE 5430
5420 LPRINT USING "##########";CPASSMIN(L) ; GOTO 5440
5430 LPRINT USING "##########";CPASSMIN(L) ;
5440 SUM = SUM + CPASSMIN(L)
5450 NEXT L
5460 LPRINT USING "##########";SUM; : SUM = 0
5470 LPRINT : LPRINT " Capacity Miles "; LPRINT TAB(21) " ";
5480 FOR L = 2 TO 7
5490 IF L < 5 THEN 5500 ELSE 5510
5500 LPRINT USING "#########";CCAPMILE(L) ; GOTO 5520
5510 LPRINT USING "#######";CCAPMILE(L) ;
5520 SUM = SUM + CCAPMILE(L)
5530 NEXT L
5540 LPRINT USING "#########";SUM; : SUM = 0
5550 LPRINT :LPRINT " Seat Miles "; LPRINT TAB(21) ""
5560 FOR L = 2 TO 7
5570 IF L < 5 THEN 5580 ELSE 5590
5580 LPRINT USING "$####8###";CSEATMILE(L) ; SGOTO 5600
5590 LPRINT USING "¢#88#8#";CSEATMILE(L) ;
5600 SUN = SUN + CSEATMILE(L)
5610 NEXT L
5620 LPRINT USING "#########";SUM; : SUM = 0
5630 LPRINT :LPRINT " Trips in Sample "; LPRINT TAB(21) ""
5640 FOR L = 2 TO 7
5650 IF L < 5 THEN 5660 ELSE 5670
5660 LPRINT USING "#########";CNOTRIPS(L) ; GOTO 5680
5670 LPRINT USING "#######";CNOTRIPS(L) ;
5680 SUM = SUM + CNOTRIPS(L)
5690 NEXT L
5700 LPRINT USING "#########";SUM; : SUM = 0
5710 IF ANTOT$ = "n" OR ANTOT$ = "N" THEN 5800
5720 LPRINT :LPRINT " Total Trips "; LPRINT TAB(21) ""
5730 FOR L = 2 TO 7
5740 IF L < 5 THEN 5750 ELSE 5760
5750 LPRINT USING "#########";TOTTRIPS(L) ; GOTO 5770
5760 LPRINT USING "#######";TOTTRIPS(L) ;
5770 SUM = SUM + TOTTRIPS(L)
5780 NEXT L
5790 LPRINT USING "#########";SUM; : SUM = 0
5800 LPRINT :LPRINT : LPRINT " SAMPLE AVERAGES" ;
5810 LPRINT :LPRINT " Unl. Pass./Trip" ; LPRINT TAB(21) "";
5820 FOR L = 2 TO 7
5830 IF L < 5 THEN 5840 ELSE 5850
5840 LPRINT USING "#########";CUNPASS(L);:GOTO 5860
5850 LPRINT USING "#######";CUNPASS(L);
5860 SUM = SUM + CUNPASS(L)
5870 NEXT L
5880 LPRINT USING "#########";SUM;
5890 LPRINT " Pass. Miles/Trip";: LPRINT TAB(21) "";
5900 FOR L = 2 TO 7
5910 IF L < 5 THEN 5920 ELSE 5930
5920 LPRINT USING "#######";CPASSMITRIP(L);:GOTO 5940
5930 LPRINT USING "#######";CPASSMITRIP(L);
5940 SUM = SUM + CPASSMITRIP(L)
5950 NEXT L
5960 LPRINT USING "#######";SUM;
5970 LPRINT " Unl. Pass. Trip";
5980 FOR L = 2 TO 7
5990 IF L < 5 THEN 6000 ELSE 6010
6000 LPRINT USING "#######";CUNPASSTIME(L);:GOTO 6020
6010 LPRINT USING "#######";CUNPASSTIME(L);
6020 SUM = SUM + CUNPASSTIME(L)
6030 NEXT L
6040 LPRINT USING "#######";SUM;
6050 IF ANTOT$ = "n" OR ANTOT$ = "N" THEN 6230
6060 LPRINT " ANNUAL TOTALS";
6070 LPRINT " Unl. Pass. Trips";: LPRINT TAB(21) "";
6080 FOR L = 2 TO 7
6090 IF L < 5 THEN 6100 ELSE 6110
6100 LPRINT USING "#######";TOTUNPASS(L);:GOTO 6120
6110 LPRINT USING "#######";TOTUNPASS(L);
6120 SUM = SUM + TOTUNPASS(L)
6130 NEXT L
6140 LPRINT USING "#######";SUM;
6150 LPRINT " Passenger Miles";: LPRINT TAB(21) "";

Appendix D. PROGRAM LISTINGS 230
FOR L = 2 TO 7
   IF L < 5 THEN "#" : GOTO 6200
   LPRINT USING "###";TOTPA$SNILE(L) : GOTO 6200
6190 LPRINT USING "###";TOTPA$SNILE(L)
6200 SUM = SUM + TOTPA$SNILE(L)
6210 NEXT L
6220 LPRINT USING "###";SUM; SUM = 0
6230 LPRINT: PRINT: PRINT: GOTO 50
6240 '-------------------PART 4 - GENERATE SUMMARY REPORT FILE-------------------
6250 '-------------------WRITTEN BY SRINATH RAJU-----------------------------
6260 CLS
6270 '-------------------RETRIEVE PROCESSED DATA FILE-----------------------
6280 '
6290 LOCATE 13,3,1
6295 INPUT "FOR WHICH OUTPUT FILE WOULD YOU LIKE THE SUMMARY REPORT FILE ?",F$
6296 LOCATE 17,3,1 : INPUT "DO YOU PREFER A SCREEN (S) DISPLAY OR A PRINTED (P) COPY ?",C$
6297 IF C$ = "S" OR C$ = "s" THEN 6300
6298 IF C$ = "P" OR C$ = "p" THEN 6300
6299 LOCATE 17,5,0 :PRINT "INVALID INPUT - CHOOSE EITHER S OR P - TRY AGAIN: GO TO 6296
6300 F$ = OUTDRV$ + F$ + ".OUT" : REN ADD DRIVE AND EXTENSION.
6310 OPEN F$ FOR INPUT AS 8
6320 INPUT #1,ROUTENO,ROUTENANE$,NOSTOPS,TRIPDIST,LOOPDIST,TURNPOINT
6330 INPUT #1,ACTION,DATAFATE$,BLOCK,DAYOFWK,BUSID,SEATS,CAP,CLICK
6340 INPUT 81,TOTDIST,TOTTINE
6345 DIM ACTTYPE(ACTION),STOPNUM(ACTION),ACTDIST(ACTION),ACTTIME(ACTION),
   SCHEDTIME(ACTION),ACTONS(ACTION),ACTOFFS(ACTION),PASSLOAD(ACTION),MATCH(ACTION)
6350 FOR I = 1 TO ACTION
6360 INPUT #1,ACTTYPE(I),STOPNUM(I),ACTDIST(I),ACTTIME(I),SCHEDTIME(I),
   ACTONS(I),ACTOFFS(I),PASSLOAD(I),MATCH(I)
6370 NEXT I
6380 CLOSE #1
6390 '
6400 '-------------------RETRIEVE NAMES REFERENCE FILE-----------------------
6410 'ADD DRIVE AND EXTENSION

Appendix D. PROGRAM LISTINGS
N$ = NAMEDRV$ + "NAMES" + RIGHTS$(STR$(ROUTENO),LEN(STR$(ROUTENO))-1)+".REF"

DRAW WINDOWS.

CLS : LINE (60,80) - (550,170),1,B
LINE (57,77) - (553,173),1,B
LOCATE 13,15,0 : PRINT "THE REFERENCE FILE WILL BE ";N$
LOCATE 18,1,0 : PRINT 
LOCATE 15,15,0 : INPUT "IS THIS THE FILE YOU WISH (Y OR N)?",R$
IF R$ = "y" OR R$ = "Y" THEN 6520
IF R$ = "n" OR R$ = "N" THEN 6490
LOCATE 18,15,0 : PRINT "INVALID CHOICE - CHOOSE EITHER Y OR N - TRY AGAIN"
: GO TO 6440
LOCATE 18,15,0 : PRINT "(REMEMBER TO INDICATE THE DRIVE AND EXTENSION)",N$
LOCATE 17,15,1 : INPUT "WHICH FILE DO YOU PREFER?",N$

FOR A PRINTED COPY
IF C$ = "P" OR C$ = "p" THEN 6540 ELSE 6570
CLS : LOCATE 13,15,0
DRAW WINDOWS
LINE (60,60) - (550,100),1,B
LINE (57,57) - (553,103),1,B
PRINT "READY PRINTER - STRIKE ANY KEY TO BEGIN PRINTING"
WS = ""
WS = INKEY$ : IF WS ="" THEN 6560
OPEN N$ FOR INPUT AS #3
INPUT #3,ROUTENO,ROUTENAME$,NOSTOPS
DIM STOPNAME$(NOSTOPS)
FOR J = 1 TO NOSTOPS
INPUT #3,STOPNAME$(J)
NEXT J
CLOSE #3

----------ACCUMULATE STATISTICS BETWEEN THE Stops----------
DIM MAXSTOP(ACTNO), MAXTIME(ACTNO), DISTANS(ACTNO), MAXLOAD(ACTNO)

I = 1: N = 1: J = 1

I = I + 1

IF I > ACTNO THEN 6780: REM ONCE END OF PROCESSED FILE

IF STOPNUM(I) > 0 AND STOPNUM(I-1) <> STOPNUM(I) THEN 6745 ELSE 6710

IF PASSLOAD(I) = MAXLOAD THEN 6755

CHECK FOR ACTIVITY RECORD WITH MAX PASSENGER LOAD.

IF PASSLOAD(I) > MAXLOAD THEN 6753 ELSE 6710

MAXLOAD = PASSLOAD(I): IF J = 1 THEN GOTO 6769 ELSE 6760

FOR K = 1 TO J

MAXSTOP(K) = 0: MAXTIME(K) = 0: DISTANS(K) = 0: MAXLOAD(J) = 0

NEXT K

J = 1: MAXSTOP(J) = STOPNUM(I): MAXTIME(J) = ACTTIME(I):

DISTANS(J) = ACTDIST(I): MAXLOAD(J) = MAXLOAD: GO TO 6710

NOTE STOP NUMBER, TIME AND DIST OF STOP WITH MAX PASS LOAD ABOVE.

'ERASE ALL PREVIOUS MAX LOAD STOPS DETAILS

FOR K = 1 TO J

MAXSTOP(K) = 0: MAXTIME(K) = 0: DISTANS(K) = 0: MAXLOAD(J) = 0

NEXT K

J = 1: MAXSTOP(J) = STOPNUM(I): MAXTIME(J) = ACTTIME(I):

DISTANS(J) = ACTDIST(I): MAXLOAD(J) = PASSLOAD(I)

GOTO 6710

COMPUTE AND PRINT TIMEPOINT INFORMATION

CLS

PRINT TAB(2) SYST$: PRINT TAB(28) "$SUMMARY REPORT FILE": PRINT TAB(60) "$PREPARED

DATE$

PRINT "BLOCK "; BLOCK; " ROUTE "; ROUTENO; PRINT TAB(28) ROUTENAMES$; PRINT

TAB(59) DATADATES$"MAXIMUM PASSENGER LOAD DETAILS" :

PRINT "STOP "; STOP NAME; MAXIMUM LOAD; DISTANCE": PRINT

TAB(68) "TIME":

FOR K = 1 TO J

ATIME = MAXTIME(K)

GOSUB 2020: REM TIME CONVERSION SUBROUTINE

STIMES$ = TEMPS
6820 'DISTANS = MAXDIST(K)
6825 DISTAN = DISTANS(K)*CLICK/5280
6828 'NOTE STOP NAME OF STOP WITH MAX LOAD
6830 SNAME$ = STOPNAME$(MAXSTOP(K))
6840 IF C$ = "p" OR C$ = "P" THEN 6880
6841 'LOCATE 4,20;O:PRINT "MAXIMUM PASSENGER LOAD DETAILS"; PRINT:
6842 'PRINT "STOP # STOP NAME MAXIMUM LOAD DISTANCE TIME";
6845 PRINT USING "88888";MAXSTOP(K); PRINT;
6850 PRINT USING "88888888"; SNAME$; PRINT " ";
6855 PRINT USING "888888888888.88";DISTAN;
6860 PRINT TAB (64) STIME$; PRINT
6870 GO TO 6950
6880 LPRINT:LPRINT:LPRINT:LPRINT
6882 LPRINT TAB(2) SYST$;LPRINT TAB(28)"SUMMARY REPORT FILE";LPRINT TAB(60)
   "PREPARED "
6883 LPRINT "BLOCK ";BLOCK;"ROUTE ";ROUTENO;LPRINT TAB(28)ROUTENAME$; LPRINT TAB(59)
   DATADATES:
6884 LPRINT TAB(25) "MAXIMUM PASSENGER LOAD DETAILS"
6885 LPRINT "STOP # STOP NAME MAXIMUM LOAD DISTANCE TIME";
6890 LPRINT USING "88888";MAXSTOP(K); PRINT;
6900 LPRINT USING "888888888888.88";DISTAN;
6910 LPRINT TAB(64) STIME$;
6920 LPRINT TAB (64) STIME$ : LPRINT
6930 ' 6940 ' 6950 NEXT K 6960 'RESET ALL ARRAYS------- 6964 FOR K = 1 TO J
6965 MAXSTOP(K) = 0 : MAXTIME(K) = 0 : DISTANS(K) = 0 6966 NEXT K 6970 '---------------- BUSY STOP DETAILS-------------------------------------
6975 PRINT TAB(25) "BUSY STOPS DETAILS":
6977 PRINT "STOP # STOP NAME ONS OFFS DISTANCE"
TIME":PRINT
6980 ' 6990 DIM SNUMBER(ACTNO), NUMONS(ACTNO), NUMOFFS(ACTNO), NUMLOD(ACTNO), NUMTIME(ACTNO)
6995 'SETTING ACTIVITY FACTOR = 5. THIS CAN BE SET TO ANY VALUE TO WATCH ALL
6997 'STOPS WITH PASSENGER ACTIVITIES GREATER THAN ACTIVITY FACTOR.
7000 AFACT = 5
7010 I = 1 : N = 1 : J = 1
7020 I = I+1
7030 IF I > ACTNO THEN 7330
7040 ' 7050 IF STOPNUM(I) > 0 AND STOPNUM(I-1) <> STOPNUM(I) THEN 7060 ELSE 7020
7060 IF ACTONS(I) >= AFACT OR ACTOFFS(I) >= AFACT THEN 7070 ELSE 7020
7070 'SNUMBER(J) = STOPNUM(I) : NUMONS(J) = ACTONS(I) : NUMOFFS(J) = ACTOFFS(I) :
' NUMLOD(I) = PASSLOAD(I) : NUMTIME(J) = ACTTIME(I)
7080 'PRINT "HIT ANY KEY TO RETURN TO MENU"
7085 'W$ = "" 7086 'W$ = INKEY$ : IF W$ = "" THEN 7086
7087 'GO TO 50 : REM GO BACK TO MENU
7090 '-----------------COMPUTE AND PRINT INFORMATION---------------------------
7110 'FOR L = 1 TO ACTNO
7120 ATIME = ACTTIME(I)
7130 GOSUB 2020 : REM TIME CONVERSION SUBROUTINE.
7140 TATIMES = TEMPS
7150 'TDISTANS = ACTDIST(I)
7160 TDISTAN = ACTDIST(I)*CLICK/5280
7170 SNAME$ = STOPNAME$(STOPNUM(I))
7180 IF C$ = "P" OR C$ = "p" THEN 7230
7185 PRINT USING "####";STOPNUM(I); : PRINT "  ";
7190 PRINT USING "$###$";SNAME$;PRINT "  ";
7200 PRINT USING "#####";ACTONS(I);ACTOFFS(I);: PRINT USING "#####";TDISTAN;
7220 PRINT TAB (68) TATIMES; PRINT
7230 GO TO 7290
7240 LPRINT;LPRINT;LPRINT;LPRINT
7245 LPRINT USING "#####";STOPNUM(I);:PRINT "  ";
LPRINT USING " "§SNAME$§2LPRINT " "§;
LPRINT USING "#######";ACTONS(I);ACTOFFS(I); LPRINT USING "############.##";
TDISTAN§;
LPRINT TAB (64) TATIME$2 LPRINT
NEXT L
---RESET ALL ARRAYS--------------------------
FOR A = 1 TO ACTNO
  SNUMBER (A) = 0 ; NUMONS(A) = 0 ; NUMOFFS(A) = 0 ; NUMLOD(A) = 0 ; NUMTIME(A) = 0
NEXT A
GOTO 7020

MAXIMUM DEVIATION DETAILS--------------------------
PRINT "DEVIATION DETAILS"
DIM DMAXSTOP(ACTNO), DMAXTIME(ACTNO), DDISTANS(ACTNO), ATIME(ACTNO)
I = 1 ; N = 1 ; J = 1
I = I + 1
IF I > ACTNO THEN 7510

IF STOPNUM(I) > 0 AND STOPNUM(I-1) <> STOPNUM(I) THEN 7410 ELSE 7370
  IF SCHEDTIME(I) > 0 THEN 7420 ELSE 7370
    COMPUTE INFORMATION ABOUT TIMEPOINTS--------------------------
    FIND DIFF OF SCHED TIME AND ACTIVITY TIME TO GET SCHED DEVIATION
    ATIME(I) = ABS(SCHEDTIME(I) - ACTTIME(I))
    IF ATIME(I) = MAXDEV THEN 7468
    IF ATIME(I) > MAXDEV THEN 7465 ELSE 7370
    J = J+1 ; DMAXSTOP(J) = STOPNUM(I) ; DMAXTIME(J) = ACTTIME(I);
    DDISTANS(J) = ACTDIST(I) ; GO TO 7370
MAXDEV = ATIME(I) ; IF J = 1 THEN 7490 ELSE 7470
IF MAXDEV THEN 7465 ELSE 7370
FOR B = 1 TO J
  DMAXSTOP(B) = 0 ; DMAXTIME(B) = 0 ; DDISTANS(B) = 0 ;
Appendix D. PROGRAM LISTINGS
7485 NEXT B
7487 ' MAX DEV DETAILS
7490 J = 1 : DMAXSTOP(J) = STOPNUM(I) : DMAXTIME(J) = ACTTIME(I) :
DDISTANS(J) = ACTDIST(I) : GOTO 7370
7500 '
7510 '------------------------PRINT THE INFORMATION ------------------------
7520 PRINT TAB(2) SYST$: PRINT TAB(28) "SUMMARY REPORT FILE"; PRINT TAB(60)
"PREPARED "; DATE$
7522 PRINT " BLOCK #"; BLOCK; " ROUTE "; ROUTENO ; PRINT TAB(28) ROUTENAME$;:
PRINT TAB(59) DATADATE$
7523 PRINT TAB(25) "MAXIMUM DEVIATION DETAILS":
7524 PRINT " STOP 	 STOP NAME 	 DISTANCE 	 SCHED DEVIATION 	 TIME":
7525 FOR C = 1 TO J
7530 ATIME = DMAXTIME (C)
7540 GOSUB 2020
7550 DSTIMES$ = TEMP$
7565 DDISTAN = DDISTANS(C)*CLICK/5280
7570 DSNAME$ = STOPNAME$(DMAXSTOP(C))
7575 ATIME = MAXDEV
7576 GOSUB 2020 : REM TIME CONVERSION SUBROUTINE
7577 DSDEV$ = TEMP$
7580 IF C$ = "P" OR C$ = "P" THEN 7620
7585 PRINT USING ";DMAXSTOP(C):PRINT "
7590 PRINT PRINT " ";DSNAME$: PRINT " 
7610 PRINT USING "DDISTAN: PRINT TAB (56)DSDEV$: PRINT TAB 
7620 IF C$ = "P" OR C$ = "P" THEN 7620
7630 PRINT USING "DMAXSTOP(C):PRINT "
7640 PRINT PRINT " ";DSNAME$: LPRINT " 
7650 PRINT USING "DDISTAN: LPRINT TAB(47) DSDEV$: LPRINT 
TAB(64) DSTIMES$: LPRINT
7670 '
7680 '

Appendix D. PROGRAM LISTINGS
1 NEXT C
7695 $ = ""
7696 $ = INKEY$ : IF $ = "" THEN 7696
7700 '-------------------------RESET ALL ARRAYS---------------------'
7710 FOR M = 1 TO J
7715 DMAXSTOP(M) = 0 : DMAXTIME(M) = 0 : ATIME(M) = 0 : DDISTANS(M) = 0
7717 NEXT M
7720 '--------------------------TOTAL DISTANCE ON THE ROUTE----------'
7730 TOTALD = TOTDIST*CLICK/5280 : REM TOTAL DIST IN MILES
7735 IF CS = "P" OR CS = "p" THEN 7750
7740 PRINT "THE TOTAL DISTANCE COVERED FOR THE DAY IS ";TOTALD " MILES"
7745 GO TO 7760
7750 LPRINT "THE TOTAL DISTANCE COVERED FOR THE DAY IS ";TOTALD " MILES"
7760 '-------------------------TOTAL NUMBER OF PASSENGERS CARRIED PER MILE--------'
7770 FOR I = 1 TO ACTNO
7780 AONS = AONS + ACTONS(I)
7790 NEXT I
7800 PPM = AONS / TOTALD
7805 IF CS = "P" OR CS = "p" THEN 7820
7810 PRINT "TOTAL NUMBER OF PASSENGERS CARRIED PER MILE IS";PPM
7815 GO TO 7830
7820 LPRINT "TOTAL NUMBER OF PASSENGERS CARRIED PER MILE IS";PPM
7830 ' 7840 '--------------------------AVERAGE SPEED FOR THE DAY---------------------'
7850 TOTALT = TOTTIME*15/3600 : REM TIME IN HOURS
7855 ASPEED = TOTALD / TOTALT
7860 IF CS = "P" OR CS = "p" THEN 7880
7870 PRINT "THE AVERAGE  SPEED FOR THE DAY IS";ASPEED " MPH"
7880 GO TO 7895
7890 LPRINT "THE AVERAGE  SPEED FOR THE DAY IS";ASPEED " MPH"
7893 GO TO 7900
7895 $ = ""
7896 $ = INKEY$ : IF $ = "" THEN 7896

Appendix D. PROGRAM LISTINGS 238
7897 CLS : GO TO 200 : REM GO BACK TO MENU
7900 CLS : END
10 ' << PLOTS >>
20 ' WRITTEN BY PAUL ANDERSON & SRINATH RAJU
30 ' 60 CLS :SCREEN 0:CLEAR
40 ' 80 '------retrieve system parameters-----------------------------
90 ' 100 OPTION BASE 1
110 OPEN "SYSTINFO.EXT" FOR INPUT AS 1
120 INPUT #1,SYST$
130 INPUT #1,DISTBASE$,BLOCKBASE$,NAMEBASE$,SECT15BASE$
140 INPUT #1,DATADR$,DISTDR$,BLOCKDR$,NAMEDR$,OUTDR$,SECT15DR$
150 PASS = -1
160 CLOSE #1
170 ' 180 '-----------------present menu-----------------------------
190 ' 200 KEY OFF
210 CLS
220 COLOR 4,0,0
230 LOCATE 6,1,0
240 PRINT TAB(20) SYST$, " APC-PLots PROGRAM MENU": PRINT
250 PRINT TAB(20) "Please choose one of the following: ": PRINT
260 PRINT TAB(25) "1 - GENERATE PASSENGER LOAD PLOT": PRINT
270 PRINT TAB(25) "2 - GENERATE ROUTE DEMAND PLOT": PRINT
280 PRINT TAB(25) "3 - QUIT (EXIT TO DOS) ": PRINT
290 INPUT " What is the number of your choice "; CHOICE
300 IF CHOICE < 1 OR CHOICE > 3 GOTO 310 ELSE 320
310 PRINT "Invalid choice - try again ":GOTO 290
320 ON CHOICE GOSUB 340,1210,2240
330 ' 340 '<<<<<<<<<<< PART 1 - PASSENGER LOAD PLOT >>>>>>>>>>>>>>>>>>>>>>>

Appendix D. PROGRAM LISTINGS 240
350 '  
360 '---------------- retrieve output file ----------------------------  
370 '  
380 CLS:LOCATE 5,27,0 : PRINT "PASSENGER LOAD PLOT"  
390 LOCATE 13,10,1  
400 INPUT " For which output file would you like the plot ? ",OUTFILE$  
410 F$ = OUTDRV$ + OUTFILE$ + ".OUT"  
420 LOCATE 15,25,0 :PRINT "Retrieving ";F$  
430 OPEN F$ FOR INPUT AS #1  
440 INPUT # 1,ROUTENO,ROUTENAME$,NOSTOPS,TRIPDIST,LOOPDIST,TURNPOINT  
450 INPUT # 1,ACTNO,DATADATE$,BLOCK,DAYOFWK,BUSID,SEATS,CAP,CLICK  
460 Z = ACTNO  
470 DIM X(Z),Y(Z),ACTTIME(Z),PASSLOAD(Z)  
480 FOR I = 1 TO Z  
490 INPUT #1,A,B,C,ACTTIHE(I),D,E,F,PASSLOAD(I),G  
500 NEXT I  
510 '  
520 CLS  
530 '  
540 '-------- display instructions for obtaining a hardcopy -------  
550 '  
560 LOCATE 10,15,0 :  
570 PRINT "The plot will be displayed on the screen."
580 LOCATE 12,15,0:PRINT "For a printed copy, wait until the plot is complete -"  
590 LOCATE 13,15,0:PRINT "Then press the left shift key along with the Prtsc key."
600 LOCATE 15,15,0:PRINT"When you have finished viewing or printing the plot, "  
610 LOCATE 16,15,0:PRINT"strike any key to return to the menu. "  
620 LOCATE 18,15,0:PRINT"Strike any key to begin plotting. "  
630 W$ =""  
640 W$ = INKEY$:IF W$ ="" THEN 640  
650 '  
660 '---------------- set the axes and print labels ---------------------  
670 '  
680 SCREEN 2
690 LOCATE 1,16,0 : PRINT SYST$; " PASSENGER LOAD PLOT " ;DADATADATE$
700 LOCATE 2,16,0 : PRINT "ROUTE ";ROUTENO$; " ROUTENAM\$; " BLOCK ";BLOCK
710 LOCATE 2,4,0 :PRINT "40"
720 LOCATE 7,4,0 :PRINT "30"
730 LOCATE 12,4,0 :PRINT "20"
740 LOCATE 17,4,0 :PRINT "10"
750 LOCATE 22,4,0 :PRINT "0"
760 '
770 LOCATE 2,1,0 :PRINT "P"
780 LOCATE 3,1,0 :PRINT "A"
790 LOCATE 4,1,0 :PRINT "S"
800 LOCATE 5,1,0 :PRINT "S"
810 LOCATE 6,1,0 :PRINT "E"
820 LOCATE 7,1,0 :PRINT "M"
830 LOCATE 8,1,0 :PRINT "G"
840 LOCATE 9,1,0 :PRINT "E"
850 LOCATE 10,1,0 :PRINT "R"
860 LOCATE 11,1,0 :PRINT "S"
870 '
880 ' y - axis
890 LINE(60,10)-(60,170)
900 FOR M = 10 TO 130 STEP 40
910 LINE (50,M)-(70,M)
920 LINE (57,M+4)-(63,M+4)
930 LINE (57,M+8)-(63,M+8)
940 LINE (57,M+12)-(63,M+12)
950 LINE (57,M+16)-(63,M+16)
960 LINE (54,M+20)-(66,M+20)
970 LINE (57,M+24)-(63,M+24)
980 LINE (57,M+28)-(63,M+28)
990 LINE (57,M+32)-(63,M+32)
1000 LINE (57,M+36)-(63,M+36)
1010 NEXT M
1020 ' x - axis

Appendix D. PROGRAM LISTINGS
1030 LINE (50,170)-(636,170)
1040 FOR N = 28 TO 572 STEP 64
1050 LINE(N+32,168)-(N+32,175)
1060 LINE(N+64,168)-(N+64,173)
1070 NEXT N
1080 LOCATE 23,8,0
1090 PRINT "4 6 8 10 12 14 16 18 20 "
1100 PRINT " TIME OF DAY (HOURS)"
1110 Y(1) = 170 : X(1) = 64
1120 FOR I = 2 TO ACTNO
1130 Y(I) = 170 - 4*PASSLOAD(I)
1140 X(I) = 32*ACTTIME(I)/240 - 68
1150 LINE(X(I),Y(I))-(X(I-1),Y(I-1))
1160 NEXT I
1170 WS = ""
1180 WS = INKEY$ : IF WS = "" THEN 1180
1190 SCREEN 0 : CLS : GOTO 60
1200 '
1210 '<<<<<<<<<<<<<< PART 2 - ROUTE DEMAND PLOT >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
1220 '
1230 '---------------- retrieve output file -------------------------------------------
1240 CLS: LOCATE 5,28,0 : PRINT "ROUTE DEMAND PLOT"
1250 LOCATE 13,10,1
1260 INPUT " For which output file would you like the plot ? ",OUTFILE$
1270 F$ = OUTDRV$ + OUTFILE$ + ".OUT"
1280 LOCATE 15,25,0 :PRINT "Retrieving ";F$
1290 OPEN F$ FOR INPUT AS # 1
1300 INPUT # 1,ROUTENO,ROUTENAMES$,NOSTOPS,TRIPTDIST,LOOPDIST,TURNPOINT
1310 INPUT # 1,ACTNO,DATADATE$,BLOCK,DAYOFWK,BUSID,SEATS,CAP,CLICK
1320 Z = ACTNO
1330 DIM ACTTIME(Z),ACTONS(Z)
1340 FOR I = 1 TO Z
1350 INPUT #1,A,B,C,ACTTIME(I),D,ACTONS(I),E,F,G
1360 NEXT I
CLS
'---- display instructions for obtaining a hardcopy ---------
LOCATE 10,10,0:
PRINT "The plot will be displayed on the screen."
LOCATE 12,10,0:PRINT "For a printed copy, wait until the plot is complete -"
LOCATE 13,10,0:PRINT "Then press the left Shift key together with the Prtsc key."
LOCATE 15,10,0:PRINT "When you have finished viewing or printing the plot,"
LOCATE 16,10,0:PRINT "strike any key to return to the menu."
LOCATE 18,10,0:PRINT "Strike any key to begin plotting."
W$ ="
W$ = INKEY$:IF W$="" THEN 1500
'---- SET THE AXES AND PRINT LABELS ---------------------
SCREEN 2
LOCATE 1,16,0:PRINT SYST$;" ROUTE DEMAND PLOT ";DATADATES
LOCATE 2,16,0:PRINT "ROUTE #";ROUTENO$; "ROUTENAMES$; " BLOCK #";BLOCK
LOCATE 2,1,0:PRINT "P"
LOCATE 3,1,0:PRINT "A"
LOCATE 4,1,0:PRINT "S"
LOCATE 5,1,0:PRINT "S"
LOCATE 6,1,0:PRINT "E"
LOCATE 7,1,0:PRINT "N"
LOCATE 8,1,0:PRINT "G"
LOCATE 9,1,0:PRINT "E"
LOCATE 10,1,0:PRINT "R"
LOCATE 11,1,0:PRINT "S"
LOCATE 14,1,0:PRINT "P"
LOCATE 15,1,0:PRINT "E"
LOCATE 16,1,0:PRINT "R"
LOCATE 19,1,0:PRINT "H"
1710 LOCATE 20,1,0 :PRINT "O"
1720 LOCATE 21,1,0 :PRINT "U"
1730 LOCATE 22,1,0 :PRINT "R"
1740 LOCATE 3,4,0 :PRINT "50"
1750 LOCATE 7,4,0 :PRINT "40"
1760 LOCATE 10,4,0 :PRINT "30"
1770 LOCATE 14,4,0 :PRINT "20"
1780 LOCATE 18,4,0 :PRINT "10"
1790 LOCATE 22,4,0 :PRINT "0"
1800 LINE(60,5)-(60,170)
1810 M = 5
1820 LINE (54,M)-(66,M)
1830 LINE (57,M+3)-(63,M+3)
1840 LINE (57,M+6)-(63,M+6)
1850 LINE (57,M+9)-(63,M+9)
1860 LINE (57,M+12)-(63,M+12)
1870 FOR M = 20 TO 150 STEP 30
1880 LINE (50,M)-(70,M)
1890 LINE (57,M+3)-(63,M+3)
1900 LINE (57,M+6)-(63,M+6)
1910 LINE (57,M+9)-(63,M+9)
1920 LINE (57,M+12)-(63,M+12)
1930 LINE (54,M+15)-(66,M+15)
1940 LINE (57,M+18)-(63,M+18)
1950 LINE (57,M+21)-(63,M+21)
1960 LINE (57,M+24)-(63,M+24)
1970 LINE (57,M+27)-(63,M+27)
1980 NEXT M
1990 ' x - axis
2000 LINE (50,170)-(636,170)
2010 FOR N = 28 TO 572 STEP 64
2020 LINE(N+32,168)-(N+32,174)
2030 LINE(N+64,168)-(N+64,172)
2040 NEXT N

Appendix D. PROGRAM LISTINGS
2050    LINE(636,168)-(636,174)
2060   LOCATE 23,8,0
2070 PRINT "4  6  8  10  12  14  16  18  20"
2080 PRINT "TIME OF DAY (HOURS)"
2090   
2100   '------------------- calculate and print the hourly demand -------------------
2110   
2120 DIM HOURDEM(24)
2130 I = 1
2140 FOR T = 4 TO 20
2150 IF ACTTIME(I)/240 < T + 1 THEN HOURDEM(T) = HOURDEM(T) + ACTONS(I)
2160   ELSE 2170
2170   I = I + 1 : IF I > ACTNO THEN 2180 ELSE 2150
2180 NEXT T
2190 FOR T = 4 TO 21
2200 LINE(32 * T * 65,170) - (32 * (T + 1) - 71, 170 - 3*HOURDEM(T)), 8
2210 NEXT T
2220 W$ = ""
2230 W$ = INKEY$ : IF W$ = "" THEN 2220
2240 SCREEN 0 :KEY ON :COLOR 7,0,0 :CLS :END
RETRIEVE THE REQUIRED SYSTEM PARAMETERS

OPTION BASE 1

OPEN "SYSTINFO.EXT" FOR INPUT AS #1

INPUT #1, SYST$

INPUT #1, DISTBASE$, BLOCKBASE$, NAMEBASE$, SECT15BASE$

INPUT #1, DATADR$, DISTDR$, BLOCKDR$, NAMEDR$, OUTDR$, SECT15DR$

PASS = -1

CLOSE #1

PRESENT PROGRAM MENU

PLOTS OPTIONS IN PROGRAM MENU

KEY OFF

CLS

COLOR 4,0,0

LOCATE 6,1,0

PRINT TAB(20) SYST$, "ADAS PLOTS PROGRAM MENU"

PRINT TAB(20) "Please choose one of the following - Type 1, 2 or 3" : PRINT

PRINT TAB(25) "1 - GENERATE ROUTE EVALUATION PLOT"

PRINT TAB(25) "2 - GENERATE SCHEDULE EVALUATION PLOT"

PRINT TAB(25) "3 - QUIT - EXIT TO DOS"

LOCATE 15,20,0

INPUT "What is the number of your choice"; CHOICE

IF CHOICE < 1 OR CHOICE > 3 THEN 310 ELSE 320

PRINT "Invalid Choice - Try Again" : GO TO 290

BRANCH OFF TO SUBROUTINES

ON CHOICE GOSUB 340, 1300, 2100

Appendix D. PROGRAM LISTINGS
PART 1 - ROUTE EVALUATION PLOT

THIS PART GIVES A PLOT OF PASSENGER LOAD VS TIMEPOINTS (18 AT A TIME)

retrieve output file

CLS : LOCATE 5,27,0 : PRINT "ROUTE EVALUATION PLOT"
LOCATE 13,10,1
INPUT "For which output file would you like the plot?",OUTFILE$ 
FS = OUTDRVS + OUTFILE$ + ".OUT"
LOCATE 15,25,0 : PRINT "Retrieving ";FS
OPEN FS FOR INPUT AS # 1
INPUT # 1,ROUTENO,ROUTENAME$,NOSTOPS,TRIPDIST,LOOPDIST,TURNPOINT
INPUT # 1,ACTNO,DATADATE$,BLOCK,DAYOFWK,BUSID,SEATS,CAP,CLICK
INPUT # 1,TOTDIST,TOTTIME
Z = ACTNO
DIM X(Z),Y(Z),ACTTIME(Z),PASSLOAD(Z),STOPNUM(Z),STPTIME(Z),SCHEDTIME(Z)
FOR I = 1 TO Z
INPUT # 1, A, STOPNUM(I), B, ACTTIME(I), SCHEDTIME(I), D, E, PASSLOAD(I), F
PRINT STOPNUM(I) : PRINT SCHEDTIME(I)
NEXT I
CLOSE # 1
CLS

Display instructions for obtaining a hardcopy

The plot will be displayed on the screen.
For a printed copy, wait until the plot is complete -
Then press the left shift key along with the Prtsc key.
When you have finished viewing or printing the plot.
Strike any key to return to the menu.
Strike any key now to begin plotting.
640 W$ = INKEY$ : IF W$ = "" THEN 640

650 '  
660 '----------------------Set the axes and print labels----------------------
670 GOSUB 680
675 GO TO 1095
680 SCREEN 2 : REM SET MODE TO STD RESOLUTION GRAPHICS MODE
720 LOCATE 1,16,0:PRINT SYST$;"ROUTE EVALUATION PLOT";DATADATE$
725 'PRINT HEADER INFORMATION
730 LOCATE 2,16,0:PRINT "ROUTE ";ROUTENO$; "ROUTENAME$"; BLOCK ";BLOCK
735 'PRINT GRADUATIONS FOR Y AXIS
740 LOCATE 1,4,0 : PRINT "40"
750 LOCATE 6,4,0 : PRINT "30"
760 LOCATE 11,4,0 : PRINT "20"
770 LOCATE 16,4,0 : PRINT "10"
780 LOCATE 21,4,0 : PRINT "0"
790 'PRINT LABELS FOR Y AXIS
800 LOCATE 2,1,0 : PRINT "P"
810 LOCATE 3,1,0 : PRINT "A"
820 LOCATE 4,1,0 : PRINT "S"
830 LOCATE 5,1,0 : PRINT "E"
840 LOCATE 6,1,0 : PRINT "N"
850 LOCATE 7,1,0 : PRINT "G"
860 LOCATE 8,1,0 : PRINT "R"
870 LOCATE 9,1,0 : PRINT "S"
880 LOCATE 10,1,0 : PRINT "T"
890 LOCATE 11,1,0 : PRINT "S"
900 ' -------------------Y AXIS---------------------------------------------
910 LINE (60,02) · (60,162) : REM DRAW VERT LINE
920 FOR M = 02 TO 122 STEP 40 : REM DRAW HORZ MARKINGS ON THE VERT LINE
930 LINE (50,M) · (70,M)
940 LINE (57,M+4) · (63,M+4)
950 LINE (57,M+8) · (63,M+8)
960 LINE (57,M+12) · (63,M+12)
970 LINE (57,M+16) · (63,M+16)

Appendix D. PROGRAM LISTINGS 249
980 LINE (54,M+20) - (66,M+20)
990 LINE (57,M+24) - (63,M+24)
1000 LINE (57,M+28) - (63,M+28)
1010 LINE (57,M+32) - (63,M+32)
1020 LINE (57,M+36) - (63,M+36)
1030 NEXT M
1040 '-----------------------------------X AXIS-----------------------------------'
1050 LINE (50,162) - (636,162) : REM DRAW HORZ LINE
1060 FOR N = 28 TO 572 STEP 32 : REM DRAW VERT MARKINGS
1070 LINE (N+32,160) - (N+32,167)
1080 'LINE (N+64,160) - (N+64,167)
1090 NEXT N
1093 'LOCATE 24,24,0 : PRINT "TIMEPOINTS (TIME IN HOURS)" : REM DELETED SINCE SCREEN GOES UP
1094 RETURN
1095 '
1096 PREV = 1
1100 Y(I) = 162 : X(I) = 64 : P = 0
1102 FOR I = 2 TO ACTNO
1106 IF STDPNUM(I) > 0 AND STDPNUM(I) <> STOPNUM(I-1) THEN 1107 ELSE 1230
1107 IF SCHEDTIME(I) = 0 THEN 1230 ELSE 1110
1110 'LOCATE ONLY TIMEPOINTS AS ABOVE
1111 '----------------print 18 timepoint information at a time-------------------X
1112 P = P + 1
1113 Y(I) = 162 - 4*PASSLOAD(I) : REM CALC Y COORDINATE
1114 X(I) = X(prev) + 32 : REM CALC X COORDINATE
1115 LINE (X(PREV),Y(PREV)) - (X(I),Y(I)) : REM JOIN THE 2 POINTS
1115 PREV = I
1116 FN = 7 + 4*P : REM CALC POSNS FOR PRINTING STOP NAS AT TIMEPOINTS.
1117 LOCATE 22,FN,0 : PRINT STOPNUM(I)
1118 STPTIME(I) = ACTTIME(I) / 240 : REM CALC TIME IN HOURS
1119 LOCATE 23,FN,0 : PRINT USING "##.#": STPTIME(I)
1120 IF P = 17 THEN 1190 ELSE 1230
1121 END
1210 IF $ = "Q" OR $ = "q" THEN 50
1220 CLS : GOSUB 680 : REM SUBROUTINE FOR SETTING AXES FOR NEXT PAGE
1225 X(1) = 64 : Y(1) = 162 : P = 0 : PREV = 1
1230 NEXT I
1240 $ = ""
1250 $ = INKEY$ : IF $ = "" THEN 1250
1260 SCREEN 0 : CLS : GO TO 160 : REM BACK TO MENU
1270 ' THIS PROGRAM ALLOWS PASSENGER LOADS TO BE ASSESSED
1280 ' FOR EVERY TIMEPOINT ALONG THE ROUTE ALLOWING THE MANAGERS TO PLAN AND ASSESS THEIR ACTIONS
1290 ° SERVING AS A DECISION AID.
1295 ' PART 2 - SCHEDULE EVALUATION PLOT
1300 ' BACK TO MENU
1310 CLS : SCREEN 0
1320 CLS : LOCATE 5,27,0 : PRINT "Schedule Evaluation Plot"
1330 LOCATE 13,10,1
1340 INPUT "For which output file would you like the plot?", OUTFI
1345 ' ADD DRIVE AND EXTENSION
1350 F$ = OUTDRVS + OUTFI$ + ".OUT"
1360 LOCATE 15,25,0 : PRINT "Retrieving ":F$ /r
1370 OPEN F$ FOR INPUT AS # 1
1380 INPUT # 1, ROUTENO, ROUTENAME$, NOSTOPS, TRIPDIST, LOOPDIST, TURNPOINT
1390 INPUT # 1, ACTNO, DATADATES$, BLOCK, DAYOFWK, BUSID, SEATS, CAP, CLICK
1395 INPUT # 1, TOTDIST, TOTTIME
1400 Z = ACTNO
1410 DIM X1(Z), Y1(Z), ACTTIME1(Z), STOPNUMI(Z), STPTIME1(Z), SCHEDTIME1(Z), SCHEDDEV(Z), SDEV(Z)
1420 FOR I = 1 TO Z
1430 INPUT # 1, A, STOPNUMI(I), B, ACTTIME1(I), SCHEDTIME1(I), C, D, E, F
1440 NEXT I
1450 CLOSE # 1
1460 CLS
1470 '  
1480 ' Display Instructions  
1500 LOCATE 10,15,0

Appendix D. PROGRAM LISTINGS
1510 PRINT "The plot will be displayed on the screen."
1520 LOCATE 12,15,0:PRINT "For a printed copy, wait until the plot is complete."
1530 LOCATE 13,15,0:PRINT "Then press the left shift key along with the Prtsc key."
1540 LOCATE 15,15,0:PRINT "When you have finished viewing or printing the plot,"
1550 LOCATE 16,15,0:PRINT "strike any key to return to the menu."
1560 LOCATE 18,15,0:PRINT "Strike any key to begin plotting."
1570 W$ = ""
1580 W$ = INKEY$: IF W$ = "" THEN 1580
1590 '  
1600 '------------------------set the axes and print labels--------------------------
1605 'THIS PLOT GIVES SCHEDULE DEVIATION VS TIMEPOINTS
1610 GOSUB 1620
1615 GOTO 1956
1620 SCREEN 2 : REM SET MODE TO STD RESOLUTION GRAPHICS MODE
1630 '  
1640 '  
1650 'PRINT HEADER INFORMATION
1660 '  
1670 LOCATE 1,16,0 : PRINT SYST$;"SCHEDULE EVALUATION PLOT ";DATADATE$
1680 LOCATE 2,16,0 : PRINT "ROUTE ";ROUTENO$;"ROUTENAME$" BLOCK ";BLOCK
1685 'PRINT AXIS GRADUATIONS
1690 LOCATE 1,4,0 : PRINT "20"
1700 LOCATE 6,4,0 : PRINT "15"
1710 LOCATE 11,4,0 : PRINT "10"
1720 LOCATE 16,4,0 : PRINT "5"
1730 LOCATE 21,4,0 : PRINT "0"
1740 'PRINT LABELS
1750 LOCATE 2,1,0 : PRINT "S"
1760 LOCATE 3,1,0 : PRINT "C"
1765 LOCATE 4,1,0 : PRINT "H"
1770 LOCATE 5,1,0 : PRINT "E"
1780 LOCATE 6,1,0 : PRINT "D"
1790 LOCATE 7,1,0 : PRINT "U"
1800 LOCATE 8,1,0 : PRINT "L"
1810 LOCATE 9,1,0 : PRINT "E"
1820 LOCATE 11,1,0 : PRINT "D"
1830 LOCATE 12,1,0 : PRINT "E"
1840 LOCATE 13,1,0 : PRINT "V"
1850 LOCATE 14,1,0 : PRINT "I"
1860 LOCATE 15,1,0 : PRINT "A"
1870 LOCATE 16,1,0 : PRINT "T"
1880 LOCATE 17,1,0 : PRINT "I"
1890 LOCATE 18,1,0 : PRINT "O"
1900 LOCATE 19,1,0 : PRINT "N"
1905 ' "--------------Y AXIS-----------------------" """"""""""
1910 LINE (60,02) · (60,162) : REM DRAW VERTICAL LINE
1915 FOR M = 02 TO 122 STEP 40 : REM DRAW HORZ MARKINGS
1920 LINE (50,M) · (70,M)
1925 LINE (57,M+4) · (63,M+4)
1930 LINE (57,M+8) · (63,M+8)
1935 LINE (57,M+12) · (63,M+12)
1936 LINE (57,M+16) · (63,M+16)
1937 LINE (54,M+20) · (66,M+20)
1938 LINE (57,M+24) · (63,M+24)
1939 LINE (57,M+28) · (63,M+28)
1940 LINE (57,M+32) · (63,M+32)
1941 LINE (57,M+36) · (63,M+36)
1942 NEXT M
1945 ' "----------X AXIS--------------------------" """"""""""
1950 LINE (50,162) · (636,162) : REM DRAW HORZ LINE
1951 FOR N = 28 TO 572 STEP 32 : REM DRAW VERT GRADUATIONS
1952 LINE (N+32,160) · (N+32,167)
1953 LINE (N+64,160) · (N+64,167)
1954 NEXT N
1955 RETURN
1956 PREV = 1
1957 Y1(1) = 162 : X1(1) = 64 : P = 0
1958 FOR I = 2 TO ACTNO

Appendix D. PROGRAM LISTINGS 253
1959 IF STOPNUM1(I) > 0 AND STOPNUM1(I) <> STOPNUM1(I-1) THEN 1960 ELSE 2050
1960 IF SCHEDTIME1(I) = 0 THEN 2050 ELSE 1965
1965 'SELECT ONLY TIMEPOINTS AS DONE ABOVE
1966 '----------------------print 18 timepoint information at a time---------------------
1980 P = P + 1
1990 SCHEDEV(I) = ABS(ACTTIME1(I) - SCHEDTIME1(I)) : REM CALC SCHED DEV
2000 SDEV(I) = SCHEDEV(I)/4
2010 Y1(I) = 162 - 8*SDEV(I) : REM FIND Y COORDINATE
2020 X1(I) = X1(PREV) + 32 : REM FIND X COORDINATE
2030 LINE (X1(PREV),Y1(PREV)) - (X1(I),Y1(I)) : REM PLOT THE POINTS AND JOIN BY A LINE
2035 PREV = I
2036 FN = 6 + 4*P : REM CALC POSITION FOR PRINTING STOP NUMBERS
2037 LOCATE 22,FN,0 : PRINT STOPNUM1(I)
2038 STPTIME1(I) = ACTTIME1(I) / 240 : REM CALC STOP TIME IN HRS.
2039 LOCATE 23,FN,0 : PRINT USING "##.8"; STPTIME1(I)
2040 IF P = 17 THEN 2041 ELSE 2050 : REM PLOT ONLY 17 POINTS
2041 WS = ""
2042 WS = INKEY$: IF WS = "" THEN 2042
2043 IF WS = "Q" OR WS = "q" THEN 50
2044 CLS : P = 0 : X1(1) = 64 : Y1(1) = 162 : PREV = 1
2045 GOSUB 1620 : REM SET AXES FOR NEXT PAGE.
2050 NEXT I
2060 WS = ""
2070 WS = INKEY$: IF WS = "" THEN 2070
2080 SCREEN 0 : CLS : GO TO 60 : REM RETURN TO MENU
2090"
2100 SCREEN 0 : CLS : KEY ON : COLOR 7,0,0 : CLS :END
Appendix E. RAW DATA FILES

E.1 B63OCT21.DAT AT ROANOKE VALLEY METRO

1 0 0 0 0
3 4 0 0 0
6 211 1 0 0
3 9 77 0 0
6 8 1 0 0
5 0 0 1 1
3 5 4 0 0
6 9 2 0 0
5 0 0 4 2
3 9 20 0 0
5 3 2 2 2
Appendix E. RAW DATA FILES
4 5 3 0 0
5 3 108 1 0
5 6 183 2 0
4 5 2 0 0
5 2 71 1 0
4 6 3 0 0
5 5 135 0 1
4 5 2 0 0
5 3 94 2 0
4 5 1 0 0
5 1 26 1 0
4 5 2 0 0
5 2 17 1 0
5 5 165 1 0
4 5 2 0 0
5 6 212 0 0
5 7 81 0 1
4 10 1 0 0
4 6 2 0 0
4 4 1 0 0
4 5 0 0 0
4 4 1 0 0
5 3 89 0 1
4 4 5 0 0
4 9 1 0 0
5 1 43 1 0
4 4 3 0 0
4 3 4 0 0
<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5</td>
<td>189</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>88</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>181</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>59</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>140</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>63</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>156</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>37</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>140</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>160</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>72</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>85</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>62</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
4 9 3 0 0
11 4 255 0 1
4 6 0 0 0
4 7 2 0 0
5 3 110 1 0
5 3 76 1 0
5 7 252 1 0
4 6 4 0 0
5 5 233 1 0
4 5 2 0 0
4 4 3 0 0
4 5 0 0 0
5 5 149 1 0
5 9 216 0 1
5 5 213 0 1
5 3 36 0 1
3 7 83 0 0
5 1 1 5 1
4 7 1 0 0
5 2 18 0 1
4 9 5 0 0
4 4 3 0 0
5 6 239 1 0
4 4 2 0 0
4 4 6 0 0
4 7 0 0 0
5 5 124 1 0
5 4 215 0 1
<table>
<thead>
<tr>
<th>5</th>
<th>2</th>
<th>69</th>
<th>1</th>
<th>0</th>
<th>4</th>
<th>3</th>
<th>6</th>
<th>0</th>
<th>0</th>
<th>4</th>
<th>5</th>
<th>4</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>4</td>
<td>106</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>63</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>8</td>
<td>74</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>12</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>2</td>
<td>95</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>87</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>48</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>251</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>154</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>225</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>9</td>
<td>141</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>25</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>9</td>
<td>141</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>3</td>
<td>8</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>7</td>
<td>225</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

Appendix E. RAW DATA FILES
67200
50024
411200
48100
45200
45100
5414210
47000
5310210
538801
5415701
513610
549810
46000
1148501
32000
614200
45400
535220
523610
45000
524110
513610
47000
45100
1
45400
5513810
46000
1148501
32000
614200
45400
535220
523610
45000
524110
513610
47000
45100
1
45400
5513810
46000
I

4 7 0 0 0
5 5 85 0 1
5 2 85 0 3
5 8 154 0 1
5 3 28 2 1
5 5 86 4 2
5 9 68 9 0
4 9 0 0 0
4 6 1 0 0
5 5 167 1 0
4 6 1 0 0
5 6 233 1 0
3 9 217 0 0
5 0 1 0 1
4 5 2 0 0
5 1 5 1 0
4 5 3 0 0
4 8 1 0 0
4 4 2 0 0
5 2 90 0 1
5 3 133 1 0
4 8 1 0 0
5 3 140 0 3
4 6 1 0 0
4 2 5 0 0
11 5 158 0 1
3 3 0 0 0
6 6 1 0 0

Appendix E. RAW DATA FILES
5 0 0 0 15
4 5 3 0 0
4 2 2 0 0
3 8 151 0 0
6 8 2 0 0
5 0 0 2 1
4 6 1 0 0
5 4 185 3 0
4 4 3 0 0
5 4 232 0 1
5 2 73 1 0
4 5 0 0 0
5 3 148 1 0
4 3 0 0 0
5 1 4 2 0
5 3 54 1 0
5 2 53 1 0
5 2 54 1 0
4 6 0 0 0
4 4 1 0 0
5 4 98 0 1
4 7 1 0 0
5 4 109 1 5
3 9 81 0 0
5 2 1 1 3
4 14 3 0 0
5 4 210 0 1
5 3 168 1 0

Appendix E. RAW DATA FILES
4 5 2 0 0
4 4 0 0 0
4 5 2 0 0
4 6 0 0 0
5 2 3 8 0 1
4 5 1 0 0
4 5 0 0 0
11 3 3 9 0 1
3 4 0 0 0
6 1 5 1 0 0
4 4 2 0 0
5 3 5 4 1 0
5 2 3 8 1 0
4 5 2 0 0
5 2 3 9 1 0
5 7 2 4 8 1 1
5 3 5 1 3 0
4 8 0 0 0
4 3 5 0 0
4 7 2 0 0
4 5 1 0 0
4 4 0 0 0
5 9 2 0 7 0 6
5 3 7 7 1 1
4 1 0 5 0 0
4 3 3 0 0
3 7 1 4 5 0 0
7 12 0 2 5 2 2 8

Appendix E. RAW DATA FILES
<table>
<thead>
<tr>
<th>46000</th>
<th>45000</th>
<th>45400</th>
<th>45300</th>
<th>44100</th>
<th>43000</th>
<th>354900</th>
<th>616100</th>
<th>46100</th>
<th>5615410</th>
<th>44500</th>
<th>43100</th>
<th>44000</th>
<th>410400</th>
<th>512401</th>
</tr>
</thead>
</table>

Appendix E. RAW DATA FILES
4 4 5 0 0
11 7 211 0 1
3 1 0 0 0
5 3 1 3 2
3 6 20 0 0
4 4 2 0 0
4 3 4 0 0
3 7 130 0 0
6 13 2 0 0
5 0 1 0 1
4 7 3 0 0
5 4 182 1 0
4 4 3 0 0
4 6 4 0 0
5 2 49 1 0
3 8 167 0 0
4 5 0 0 0
4 3 0 0 0
4 7 0 0 0
4 5 4 0 0
4 5 0 0 0
4 8 1 0 0
5 5 81 0 1
3 5 8 0 0
5 3 1 0 1
3 10 80 0 0
6 6 2 0 0
5 0 1 4 2

Appendix E. RAW DATA FILES
<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5 8</td>
<td>92</td>
<td>2 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 9</td>
<td>147</td>
<td>1 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 9</td>
<td>2 0</td>
<td>0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 3</td>
<td>2 0</td>
<td>0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 5</td>
<td>0 0</td>
<td>0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 3</td>
<td>58</td>
<td>0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 6</td>
<td>2 0</td>
<td>0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 3</td>
<td>5 0</td>
<td>0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 5</td>
<td>1 0</td>
<td>0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 5</td>
<td>3 0</td>
<td>0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 4</td>
<td>135</td>
<td>0 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 2</td>
<td>69</td>
<td>0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 4</td>
<td>233</td>
<td>0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 4</td>
<td>0 0</td>
<td>0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 3</td>
<td>140</td>
<td>0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 4</td>
<td>140</td>
<td>0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 3</td>
<td>0 0</td>
<td>0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 31</td>
<td>1 0</td>
<td>0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 0 0</td>
<td>3 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 7 1</td>
<td>0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 3 0</td>
<td>0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 4 4</td>
<td>0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 3</td>
<td>155</td>
<td>1 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 7 0</td>
<td>0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 3 5</td>
<td>0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 3 1</td>
<td>0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 5 0</td>
<td>0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 5</td>
<td>246</td>
<td>1 0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Appendix E. RAW DATA FILES
<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>46000</td>
<td>41100</td>
<td>36780</td>
<td>64200</td>
<td>50123</td>
</tr>
<tr>
<td>34400</td>
<td>61510</td>
<td>50020</td>
<td>58762</td>
<td>52101</td>
</tr>
<tr>
<td>52580</td>
<td>53840</td>
<td>44100</td>
<td>44400</td>
<td>53108</td>
</tr>
<tr>
<td>37194</td>
<td>51116</td>
<td>44200</td>
<td>45200</td>
<td>54140</td>
</tr>
<tr>
<td>51310</td>
<td>11229</td>
<td>49400</td>
<td>52540</td>
<td>52350</td>
</tr>
</tbody>
</table>

Appendix E. RAW DATA FILES
45000
512910
5823801
45300
44200
44200
47200
411000
351700
51271
3108000
65200
50080
51422810
49000
51910
46500
525701
5516103
515201
5310401
43200
527003
51025301
526101
45500
5219501
Appendix E. RAW DATA FILES
Appendix E. RAW DATA FILES

5 6 189 0 2
5 4 143 0 1
5 4 76 0 1
4 3 1 0 0
4 3 6 0 0
4 4 0 0 0
4 3 0 0 0
5 6 55 1 0
11 1 1 4 0 1
5 2 2 1 1
4 4 0 0 0
4 4 4 0 0
4 3 4 0 0
4 3 0 0 0
3 6 198 0 0
5 7 136 3 0
4 7 0 0 0
5 4 247 1 0
4 6 1 0 0
4 4 2 0 0
5 2 50 0 1
4 8 4 0 0
4 4 0 0 0
4 3 0 0 0
4 5 0 0 0
5 3 108 0 1
4 5 3 0 0
4 8 2 0 0
5 5 83 1 2
5 10 81 4 2
5 11 222 0 1
5 6 238 0 1
4 4 0 0 0
5 5 217 0 1
4 4 1 0 0
4 8 2 0 0
5 3 81 0 1
4 4 5 0 0
4 5 3 0 0
11 5 187 0 1
4 6 1 0 0
5 2 66 0 1
4 4 1 0 0
4 5 4 0 0
5 2 60 1 0
4 6 0 0 0
5 3 107 0 1
4 4 1 0 0
4 6 2 0 0
4 5 1 0 0
4 6 3 0 0
3 10 151 0 0
6 11 1 0 0
3 8 71 0 0
5 6 95 1 0
4 10 0 0 0
273

Appendix E. RAW DATA FILES
4 4 2 0 0
3 7 172 0 0
7 29 1 114 165
8 0 1 67 85
12 10 21 86 5
12 2 63 18 55
12 194 0 0 0
### E.2 ACT3.DAT AT TRT NORFOLK

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>138</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>92</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>19</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>58</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>78</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>161</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>23</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>199</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1</td>
<td>12</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>138</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>143</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>67</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>44</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>244</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>83</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>1</td>
<td>8</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>66</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>70</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>220</td>
<td>3</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>186</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>155</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>41</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>44</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>67</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>180</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>55</td>
<td>0</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>72</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>5</td>
<td>246</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>112</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>-----</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>218</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>139</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>95</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>141</td>
<td>1</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>137</td>
<td>0</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>109</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>176</td>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>135</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>254</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>177</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>1</td>
<td>8</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>168</td>
<td>8</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>199</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>34</td>
<td>5</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>174</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>92</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>100</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>13</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
4 5 0 0 0
5 1 72 0 2
5 3 184 1 3
5 5 90 1 0
5 3 156 0 1
5 2 174 0 3
4 3 0 0 0
4 2 0 0 0
11 2 239 0 1
4 2 0 0 0
3 4 78 0 0
5 5 2 3 0
6 0 0 0 0
5 2 168 1 0
4 2 0 0 0
5 2 162 1 2
5 4 105 1 0
5 3 247 1 1
11 1 63 0 1
5 2 100 0 1
5 2 75 0 2
11 1 45 0 1
5 6 58 3 12
3 4 1 0 0
5 8 141 7 3
3 5 88 0 0
4 8 1 0 0
4 5 0 0 0

Appendix E. RAW DATA FILES 278
<p>| 4 | 5 | 2 | 78 | 1 | 0 |
| 4 | 3 | 0 | 0 | 0 | 0 |
| 4 | 2 | 0 | 0 | 0 | 0 |
| 5 | 2 | 175 | 1 | 0 | 0 |
| 5 | 2 | 116 | 1 | 0 | 0 |
| 5 | 1 | 2 | 0 | 1 | 1 |
| 5 | 2 | 142 | 1 | 0 | 0 |
| 4 | 3 | 0 | 0 | 0 | 0 |
| 5 | 2 | 63 | 0 | 1 | 1 |
| 5 | 5 | 203 | 0 | 1 | 1 |
| 4 | 4 | 0 | 0 | 0 | 0 |
| 4 | 1 | 0 | 0 | 0 | 0 |
| 5 | 3 | 150 | 0 | 3 | 3 |
| 5 | 3 | 146 | 3 | 0 | 0 |
| 4 | 5 | 0 | 0 | 0 | 0 |
| 5 | 3 | 231 | 1 | 2 | 2 |
| 4 | 3 | 0 | 0 | 0 | 0 |
| 4 | 2 | 0 | 0 | 0 | 0 |
| 5 | 2 | 129 | 0 | 3 | 3 |
| 5 | 3 | 206 | 0 | 2 | 2 |
| 5 | 3 | 249 | 1 | 0 | 0 |
| 5 | 3 | 118 | 0 | 2 | 2 |
| 5 | 3 | 200 | 0 | 3 | 3 |
| 5 | 4 | 222 | 0 | 1 | 1 |
| 4 | 2 | 0 | 0 | 0 | 0 |
| 5 | 2 | 59 | 0 | 1 | 1 |
| 4 | 3 | 0 | 0 | 0 | 0 |
| 5 | 1 | 9 | 0 | 2 | 2 |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>9</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>18</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>76</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>46</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>105</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>216</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>39</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>95</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>207</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>51</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>182</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>164</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>195</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Tag</td>
<td>Value1</td>
<td>Value2</td>
<td>Value3</td>
<td>Value4</td>
<td>Value5</td>
</tr>
<tr>
<td>-----</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>172</td>
<td>3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>151</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>234</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>138</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>93</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>87</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>166</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>187</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>141</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>216</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>31</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>98</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>191</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>41</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>215</td>
<td>0</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>241</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Appendix E. RAW DATA FILES
<table>
<thead>
<tr>
<th>4</th>
<th>2</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>3</td>
<td>162</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>167</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>11</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>196</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>84</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>25</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>235</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>32</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>82</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>95</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>187</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>------</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>149</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>91</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>63</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>179</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>179</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>128</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>110</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>208</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>71</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>158</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>80</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>144</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>174</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>52</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>73</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>196</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>15</td>
<td>10</td>
<td>09</td>
<td>87</td>
<td>143</td>
</tr>
<tr>
<td>15</td>
<td>5</td>
<td>2</td>
<td>18</td>
<td>20</td>
</tr>
</tbody>
</table>

Appendix E. RAW DATA FILES
## Appendix F. Table showing mapping of VPI Reference Files with that of TRT Files.

<table>
<thead>
<tr>
<th>TRT REFERENCE SYSTEM</th>
<th>VPI REFERENCE SYSTEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>Filename</td>
</tr>
<tr>
<td>Brown Sh</td>
<td>BrShRte</td>
</tr>
<tr>
<td>Brown Sh</td>
<td>BrShRte</td>
</tr>
<tr>
<td>Brown Sh</td>
<td>BrShRte</td>
</tr>
<tr>
<td>Brown Sh</td>
<td>BrShRte</td>
</tr>
<tr>
<td>Brown Sh</td>
<td>BrShRte</td>
</tr>
<tr>
<td>Brown Sh</td>
<td>BrShRte</td>
</tr>
<tr>
<td>Brown Sh</td>
<td>BrShRte</td>
</tr>
<tr>
<td>Headway</td>
<td>Headact</td>
</tr>
<tr>
<td>Headway</td>
<td>Headact</td>
</tr>
</tbody>
</table>
Appendix G. Figure showing the assignment of match values and the process of matching
The vita has been removed from the scanned document