FHWA-NJ-2012-004

Implementation of Maintenance Decision Support System in New Jersey

FINAL REPORT

July 27, 2012

Submitted by

Steven Chien, Ph.D.

Department of Civil and Environmental Engineering

New Jersey Institute of Technology

University Heights, Newark, NJ 07102-1982

NJDOT Research Project Manager

Alejandro Perez-Deleon

In cooperation with

New Jersey Department of Transportation
Bureau of Research
And
U.S. Department of Transportation
Federal Highway Administration

DISCLAIMER STATEMENT

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No. FHWA-NJ-2012-004	**2. Government Accession No.**	**3. Recipient's Catalog No.**
4. Title and Subtitle Implementation of Maintenance Decision Support System in New Jersey		**5. Report Date** July 17, 2012
		6. Performing Organization Code NJIT
7. Author(s) Steven Chien and Jay Meegoda		**8. Performing Organization Report No.** FHWA-NJ-2012-004
9. Performing Organization Name and Address Department of Civil and Environmental Engineering New Jersey Institute of Technology University Heights Newark, NJ 07102-1982		**10. Work Unit No.**
		11. Contract or Grant No. NJDOT 2007 – 09
12. Sponsoring Agency Name and Address N.J. Department of Transportation 1035 Parkway Avenue P.O. Box 600 Trenton, NJ 08625-0600	Federal Highway Administration U.S. Department of Transportation Washington, D.C.	**13. Type of Report and Period Covered** Final Report July 2007 – April 2012
		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

This study developed a maintenance decision support system for New Jersey, called NJ-MDSS. The state-of-the-art weather forecasting and data fusion techniques are merged with computerized winter road maintenance rules of practice, such as consolidated weather forecasting, specific current and future roadway and bridge deck condition information, and treatments and timeline applications in NJ-MDSS. Optimally, NJ-MDSS can assist NJDOT to make decisions based on information, mainly collected by Road Weather Information System (RWIS), including various surface transportation related sensors, via *Clarus*, which ultimately lead to a higher level of service, reduced weather-related congestion delay and accidents, reduced redundancy and environmental/ecological impacts, more efficient use of manpower, contractor services, fleet and asset management, and increased accountability, resulting in more prudent and efficient use of resources. The deliverables of NJ-MDSS in this pilot project include a set of guidelines for maintenance supervisors, and provide forecast of road surface conditions and treatment recommendations for selected roadways in Warren County, New Jersey. The developed NJ-MDSS can be quickly scaled up to regional or state wide winter road maintenance implementation and applications.

17. Key Words Winter Road Maintenance, Decision Support, Weather Forecast	**18. Distribution Statement** No Restrictions

19. Security Classification (of this report) Unclassified	**20. Security Classification (of this page)** Unclassified	**21. No of Pages** 75	**22. Price**

Form DOT F 1700.7 (8-69)

ACKNOWLEDGEMENTS

This study was completed with the help of the researchers at the New Jersey Institute of Technology (NJIT) and Rutgers the State University of New Jersey:

- Dr. David Robinson, Professor of Geography Department, Rutgers, The State University of New Jersey
- Dr. Kung-E Cheng, Research Scientist of Civil and Environmental Engineering, NJIT
- Haifeng Yu, Research Assistant, Interdisciplinary Program in Transportation, NJIT

This project could not have been accomplished without the assistance of numerous individuals. The authors would like to express gratitude to David Bowlby, Lisa Webber, Scott Sheldon, Nancy Ciarufolli, Robert Sasor, and Stefanie Potapa of the New Jersey Department of Transportation (NJDOT). The authors also would like acknowledge the support of Jim Cowie and Michael Chapman with the University Corporation for Atmospheric Research (UCAR), and Paul Pisano with the Federal Highway Administration.

TABLE OF CONTENTS

Page

EXECUTIVE SUMMARY **1**

INTRODUCTION **2**

OBJECTIVES **3**

LITERATURE REVIEW **3**

Maintenance Decision Support System (MDSS) 4

MDSS Benefits and Costs 5

SUMMARY OF THE WORK PERFORMED **7**

Study the FHWA’s MDSS Prototype 7

Road Condition and Treatment Module (RCTM) 8

Pavement Frost Product (Frost) 9

Alert Generator 9

Treatment Update Network Layer (TUNL) 9

Thematic Real-time Environmental Distributed Data Services (THREDDS) and Web Map Service (WMS) Server 9

Identify Pilot Study Region and Investigate Existing Data Sources 9

Weather Forecast Sites 10

Weather Observation Sites 10

MDSS Database Development 13

Collect Data 13

Development of NJ-MDSS 14

Implementation and Training 14

Web-based NJ-MDSS User Account Management 15

Training of NJDOT Personnel 16

CONCLUSIONS AND RECOMMENDATIONS 16

Summary of NJ-MDSS Benefits 16

Future Improvement of NJ-MDSS 16

REFERENCES 18

APPENDIX A –FWHA AND NJDOT SNOW MAINTENANCE GUIDELINES 23

APPENDIX B – NJ-MDSS FORMS FOR DATABASE DEVELOPMENT 26

APPENDIX C –PROGRAMS DEVELOPED FOR NJ-MDSS 29

Weather Forecast data From DTN 29

Weather Observations Data From DTN 30

Weather Observation Data From*Clarus* 32

Traffic Camera Images From NJ511 34

AVL Data From NJDOT 38

APPENDIX D - MDSS HARDWARE AND SOFTWARE REQUIREMENTS 41

APPENDIX E –USER'S MANUAL 44

Instruction of User Interface 44

Introduction of Treatment Inputs 53

Alerts in NJ-MDSS 56

APPENDIX F -WEATHER DATA 58

Weather Forecast Data 58

Observational History 65

LIST OF TABLES

Page
Table 1 Costs and benefits associated with MDSS 6
Table 2 Summary of cost and benefit analyses 7
Table 3 Weather forecast sites in NJ-MDSS 11
Table 4 NJDOT RWIS sites 12
Table 5 NJ-MDSS modules and forms 13
Table 6 NJ-MDSS data inputs 13
Table 7 NJ-MDSS user's privileges 15

LIST OF FIGURES

Page
Figure 1. Relationship between level of service and maintenance resources 6
Figure 2. Overview of MDSS 8
Figure 3. NJ-MDSS road sections 10
Figure 4. Web-based user account management: add user account 15
Figure 5. Web-based user account management: delete user account 15

EXECUTIVE SUMMARY

Managing winter maintenance activities is a fairly complex endeavor. Maintenance supervisors must have sufficient knowledge of regulations on chemical applications and their associated environmental impacts. In addition, maintenance supervisors are faced with tight budgets, which may complicate the process to perform and contract salt spreading and plowing services to keep roads safer while driving. Therefore, it is desirable that today's maintenance supervisors have the ability to efficiently handle multiple tasks, and process large volumes of information in adverse weather conditions.

This study developed a maintenance decision support system for New Jersey called NJ-MDSS. The state-of-the-art weather forecasting, and data fusion techniques, are merged with computerized winter road maintenance rules of practice, such as consolidated weather forecasting, specific current and future information on roadway and bridge deck condition, and treatments and timeline applications in NJ-MDSS. Optimally, NJ-MDSS can assist New Jersey Department of Transportation (NJDOT) to make decisions based on information collected by Road Weather Information System (RWIS) via *Clarus*. The implementation of NJ-MDSS is expected to lead to a higher level of service, reduced weather-related congestion delay and accidents, reduced redundancy and environmental/ecological impacts, efficient use of manpower and equipment, and increased accountability, resulting in more prudent and efficient spending.

The deliverables of NJ-MDSS include a set of guidelines for maintenance supervisors, which provide treatment recommendations for selected roadways in Warren County, New Jersey. With the developing weather conditions and the availability of chemicals and manpower/vehicles, NJ-MDSS provides real-time information (i.e., weather forecasts, road surface temperatures, and traffic conditions) for winter road treatment. The developed NJ-MDSS can be quickly scaled up for regional or statewide winter road maintenance implementation and applications.

INTRODUCTION

Winter road maintenance accounts for approximately 25 percent of NJDOT's maintenance budget. Each year, state and local agencies spend more than $2.5 billion on snow and ice control operations, and more than $5 billion to repair infrastructure damaged by snow and ice. Salt is found to be the most viable de-icing material which keeps roads, highways and pedestrian paths open during winter weather; its use accounted for $289.5 Million of operational costs in 1998 (USGS, 2002). Adverse weather conditions dramatically affect the nation's surface transportation system. Each year, 6,600 people die, 470,000 people are injured, and 544 million hours of time are lost on the nation's highways because of adverse weather conditions (FHWA, 1996).

Managing winter maintenance activities is a fairly complex endeavor. Maintenance supervisors must know the regulations of chemical applications and their environmental impacts, and be able to analyze and make sense of multiple, and often contradictory, weather forecasts. In addition, many maintenance supervisors are also faced with tight budgets. This is further complicated by the need to obtain salting and plowing services from outside contractors. All of these factors challenge public agencies to meet the traveling public's high expectation that roads be free of snow and ice. Therefore, it is desirable that today's maintenance supervisors have the ability to efficiently handle multiple tasks, and process high volumes of information in adverse winter weather conditions.

The Federal Highway Administration (FHWA), Office of Transportation Operations (HOTO), has long recognized the challenges faced by maintenance managers. With the creation of the Road Weather Management Program in the late 1990s, FHWA began to improve the type of information that was available for winter maintenance. The initial guidance for anti-icing operations were detailed in the FHWA "*Manual of Practice for an Effective Anti-icing program: A guide for Highway Maintenance Personnel*". In this FHWA document a series of tables were created to facilitate winter operations (see APPENDIX A). However, there was no link between the available weather information and the decisions made by maintenance supervisors about winter road treatments, such as when it is best to treat which roads, and whether salt spreading, plowing, or a combination of approaches is most effective. The recognition of this missing link led to the genesis of the winter Maintenance Decision Support System (MDSS). Later, FHWA's Road Weather Management Program, in conjunction with the Intelligent Transportation Systems (ITS) Joint Program Office, initiated a new program called "*Clarus*—the Nationwide Surface Transportation Weather Observing and Forecasting System". The objective of *Clarus* is to design, develop, and demonstrate advanced weather forecasting for the surface transportation community, based on data collected from state-funded surface transportation-related observations (atmospheric, road surface, and hydrologic).

OBJECTIVES

The primary goal of NJDOT is to provide a safe and dependable transportation infrastructure for the movement of people and goods throughout the state. Weather has become an increasingly challenging factor. In order to combat snow and ice conditions on state roads, it is urgent for NJDOT to have a winter roadway advisory system, such as NJ-MDSS based on NJDOT's rules of practices (See APPENDIX A), which can efficiently gather and disseminate information to help maintenance supervisors make timely and accurate decisions based on large amounts of information under a wide range of winter conditions. The decisions may include, but not be limited to, what treatments to apply, and when to apply them, based on up-to-the-minute current and forecasted pavement and weather conditions, as well as available maintenance techniques, resources, and limited staff and funding. Hence the objective of this study is to develop NJ-MDSS to:

- Bolster or improve the accuracy and speed of information gathering and dissemination;
- Combine data and create an open, integrated, and understandable presentation of current environmental, atmospheric and roadway conditions;
- Process and query data to generate diagnostic and prognostic GIS mapping of road conditions along study corridors;
- Predict future changes in roadway conditions to aid in resource management;
- Notify NJDOT of up-to-the-minute conditions and suggest optimal maintenance treatments for future changes in conditions; and
- Train NJDOT personnel for winter road maintenance operation with real-time weather and road surface conditions.

LITERATURE REVIEW

The development of a prototype Maintenance Decision Support System (MDSS) is part of the FHWA's effort to improve winter road maintenance. The effective use of deicing/anti-icing materials can minimize weather related accidents; however, storm run-off containing road salts used in deicing operations has become a source of contamination of surface and sub-surface water bodies and water ways, which provide water for human consumption and agriculture. Hence, an effective application of deicing/anti-icing materials is a balancing act, and can be best achieved by the deployment of a MDSS.

The optimal winter road treatment options depend largely on the timing of applications. This requires the extensive use of Road Weather Information Systems (RWIS), accurate weather predictions, timely and efficient mobilization of machinery and manpower, experience, decision making, and coordination at operations level. RWIS stations are equipped with various sensors to monitor road surface temperatures, and measure wind speed and direction, air temperatures and precipitation. The information gathered from these sensors, coupled with topography and road profile, can be used in a computer model that provides maintenance decision support to map road temperatures, and assist in identifying critical road sections and recommending corresponding treatment alternatives.

Maintenance Decision Support System (MDSS)

Controlling snow and ice buildup on roadways during winter weather events presents several challenges for winter maintenance personnel. In an effort to mitigate the challenges associated with winter maintenance decisions, the Federal Highway Administration (FHWA) Office of Transportation Operations (HOTO) initiated a program, in 2001, aimed at developing a winter road Maintenance Decision Support System (MDSS).

The following five national research centers participated in the development of the MDSS Functional Prototype:

- Army Cold Regions Research and Engineering Laboratory (CRREL)
- National Center for Atmospheric Research (NCAR)
- Massachusetts Institute of Technology - Lincoln Laboratory (MIT/LL)
- NOAA National Severe Storms Laboratory (NSSL)
- NOAA Forecast Systems Laboratory (FSL)

The goal of MDSS was to develop a prototype with the following capabilities (the Federal Highway Administration Office of Transportation Operations, 2001):

- Capitalizes on existing road and weather data sources;
- Augments data sources where they are weak or where improved accuracy could significantly improve the decision-making process;
- Fuses data to make an open, integrated and understandable presentation of current environmental and road conditions;
- Processes data to generate diagnostic and prognostic maps of road conditions along road corridors, with emphasis on the 1- to 48-hour forecast (historical information from the previous 48 hours will also be available);
- Provides a display of the atmosphere and roadway conditions
- Provides a decision support tool, which provides recommendations on road maintenance courses of action; and
- Provides all of the above on a single platform, with simple and intuitive operating requirements, and does so in a readily comprehensible display of results and recommended courses of action, together with anticipated consequences of action or inaction.

An MDSS field demonstration was conducted in 2003, which evaluated the operation system components in a real-time winter environment, including:

- Weather prediction component
- Treatment recommendations

- Impact of supplemental meso-scale models
- Potential benefit of an operational system
- Identification and evaluation of current system limitations

Based on field demonstrations, MDSS has shown promising results in assisting winter maintenance managers in fighting winter storms effects on roads (Kroeger and Burkheimer, 2003). Andrle et al. (2003) indicated that technologies are mature for private sectors to incorporate MDSS prototype capabilities into their products; however, the MDSS weather predictions need improvement.

Since the first field demonstration in 2003, the deployment of FHWA's MDSS prototype has been progressing along several fronts. The most extensive deployment is a pooled fund (PF) effort by eight states, including South Dakota, which is leading the effort, along with Colorado, Indiana, Iowa, Kansas, Minnesota, North Dakota, and Wyoming. The PF-MDSS is a partnership between several state DOTs and a commercial weather service provider (Meridian Environmental Technology, Inc.), whereby these DOTs contribute funds toward the development of a commercial MDSS. The PF-MDSS was developed based on user requirements, technical components, and lessons learned from the FHWA prototype MDSS project.

In 2007, twenty one state transportation agencies were using or developing MDSS tools. From 2002 to 2012, the prototype underwent six development cycles and three field demonstrations in Iowa and Colorado. The PF-MDSS led by the South Dakota Department of Transportation (SDDOT) was expanded to sixteen states in 2012. Version 6.1 of the MDSS software is now available from the NCAR's MDSS website. The FHWA offers free MDSS demonstrations for state agency managers and field personnel. The AASHTO Technology Implementation Group has selected the MDSS as a "Focus Technology". The Technology Implementation Group identifies and champions the deployment of "ready-to-use" technologies that are likely to yield significant benefits to users (FHWA Road Weather Management Program Projects and Activities, 2007).

MDSS Benefits and Costs

The FHWA has conducted cost and benefit (C&B) analyses of MDSS applications being used by the pooled fund states. The purpose of these analyses was to assess the benefits and the costs associated with the MDSS implementation, and to distill this information in a format that is accessible and actionable to transportation agency decision-makers and elected officials. In an MDSS case study in Maine, Cluett and Jenq (2007) indicated that costs and benefits vary depending upon local operation and snow event characteristics (i.e., the rate, timing, and duration of a snow fall).

Lead by SDDOT and other PF-MDSS partners in making decisions on future investments in MDSS, in 2006,Western Transportation Institute and Iteris, Inc. conducted a cost and benefit analysis, which provides transportation agencies a foundation to evaluate deployment requirements and potential benefits of MDSS; also methods for measuring improvements relevant to MDSS. The costs and benefits associated with MDSS are identified in Table 1.

Table 1 Costs and benefits associated with MDSS

	Agency	Motorist	Society
Benefits	Reduced materials costs; Reduced labor costs; Reduced equipment costs; Reduced fleet replacement costs; and Reduced infrastructure damage due to road salts.	Reduced motorist delays (through improved LOS); Improved safety (through improved LOS); Reduced response time; Reduced clearance time; and Reduced vehicular corrosion due to road salts.	Reduced environmental degradation
Costs	Software and support costs; Communications costs; In-vehicle computer hardware Investment; Training; Administrative costs; and Weather forecast provider costs.		

Some of the tangible benefits of implementing MDSS are reducing material usage (resources), improving mobility and safety, and decreasing travel time. The latter two items are realized through the improvement of pavement conditions or level of service. The relationship between level of service and winter maintenance resources is represented in Figure 1 (Western Transportation Institute and Iteris, Inc., 2006).

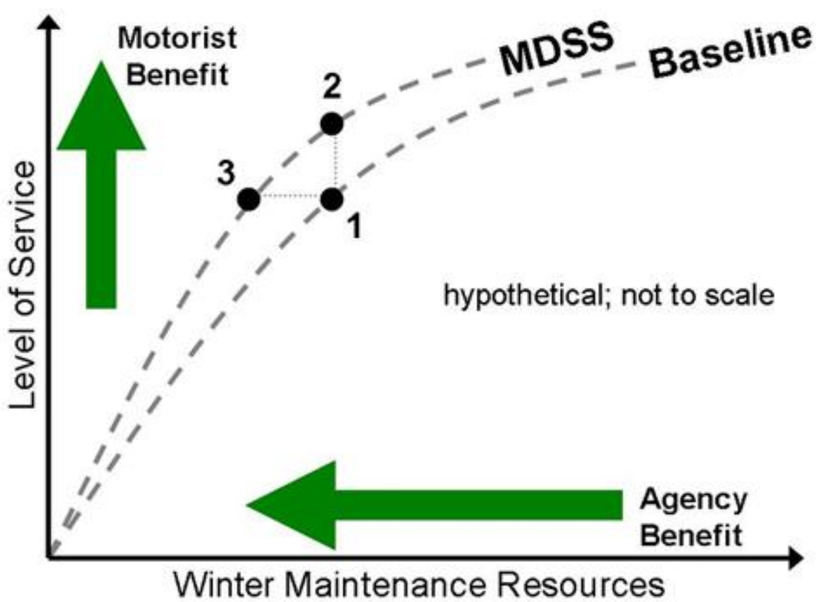


Figure 1.Relationship between level of service and maintenance resources

The methodology for benefit and cost analysis consists of two modules: a baseline module and a simulation module. In the baseline module, various data, including highway route information, winter maintenance resource usage, traffic volume, crash data, and weather information, are incorporated to establish detailed information for each route section. MDSS was applied in the simulation module to generate outputs considering each of the following three scenarios:

- Base Case: Assuming under a baseline condition by following rules of practice, how much would winter maintenance costs be?
- Same Resources: Assuming that winter maintenance costs are kept constant and MDSS treatment recommendations are followed, what would be the resulting level of service under MDSS use?
- Same Condition: Assuming that level of service is kept constant and MDSS treatment recommendations are followed, what would be the resulting winter maintenance costs when MDSS is used?

A summary of C & B analyses in New Hampshire, Minnesota, and Colorado are illustrated in Table 2.

Table 2 Summary of cost and benefit analyses

Case State	Scenario	Benefits	User Savings (%)	Agency Savings (%)	Costs	B-C Ratio
New Hampshire	Same Condition	$2,367,409	50	50	$332,879	7.11
	Same Resources	$2,884,904	99	1		8.67
Minnesota	Same Condition	$3,179,828	51	49	$496,952	6.40
	Same Resources	$1,369,035	187	-87		2.75
Colorado	Same Condition	$3,367,810	49	51	$1,497,985	2.25
	Same Resources	$1,985,069	90	10		1.33

The results show that the application of MDSS for winter road maintenance seems beneficial. However, the results also indicate that there is a trade-off between agency benefits and user benefits. The increased use of material seemed to increase agency spending, but reduced user cost.

SUMMARY OF THE WORK PERFORMED

To develop a New Jersey based Maintenance Decision Support System (NJ-MDSS), a list of tasks were conducted as discussed below.

Study the FHWA's MDSS Prototype

The proposed NJ-MDSS is customized based on the FHWA's MDSS prototype. The required data inputs for NJ-MDSS identified and collected in this project, include:

- General System Requirements;
- MDSS Coverage Area;
- Weather Forecast;
- Weather Observation;
- Road Condition Observation;
- Road Condition Prediction; and
- Forecast Confidence.

With the assistance of NJDOT, FHWA, and UCAR, the NJ-MDSS structure was developed as shown in Figure 2:

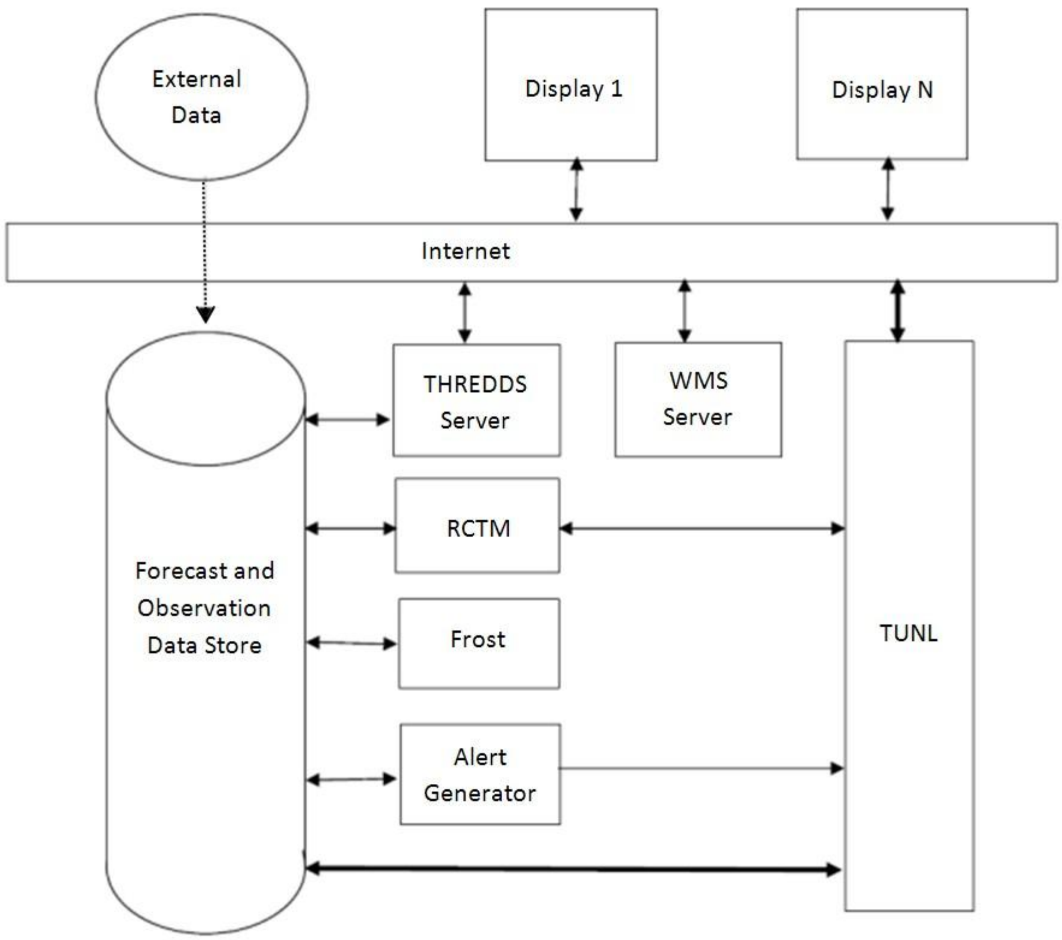


Figure 2. Overview of MDSS

(MDSS Prototype Relase-6 Technical Description)

Road Condition and Treatment Module (RCTM)

The RCTM processes environmental forecast data to predict the road surface temperature and snow depths at all forecast sites and lead times. Using the meteorological forecast data, the pavement temperature data, and the snow depth on the road, a predicted mobility index is calculated along with a treatment plan.

Pavement Frost Product (Frost)

The Frost Potential Algorithm was developed to forecast whether frost is likely to occur or not, based on weather and road condition forecasts. A frost deposition algorithm developed by IA-DOT (Takle and Greenfield, 2005) was used to calculate the deposition and evolution of frost on the road.

Alert Generator

The Alert Generator is a program which creates weather and road alerts based on observations and forecasts. Alerts are designed to be simple notifications of pre-determined hazardous weather or road conditions that are either occurring at the present time, or will be occurring in the next few hours. They are intended to prompt the attention of operators to examine weather, road, blowing and bridge frost conditions in more detail. Alerts are generated on a route-by-route basis, and are binary in nature; that is, for each route, an alert condition either exists or does not exist.

Treatment Update Network Layer (TUNL)

The Treatment Update Network Layer is a collection of Python CGI scripts which are run by a web server in response to a request from a display. The display may request the time of the most recent type of data, request the data itself, or may ask to run a what-if scenario, or select a treatment.

Thematic Real-time Environmental Distributed Data Services (THREDDS) and Web Map Service (WMS) Server

The THREDDS server provides gridded data sets to the display. In MDSS Release-6.1, these include radar and satellite images. And the WMS server provides dynamic base map images for use by the displays.

Identify Pilot Study Region and Investigate Existing Data Sources

The NJ-MDSS service region in this study is focusing on the road segments in Warren County, maintained by the crew from the Columbia Yard. Two major highways are covered: Interstate I-80 from mile post (MP) 0.5 to MP 19.9, and US 46 from MP 0 to MP 10. These road segments are divided into smaller sections for winter road maintenance. The sections are: MP 0.5 to 6, MP 6 to 12, and MP 12 to 19.9 on I-80, and MP 0 to 10 on US 46 (see Figure 3).

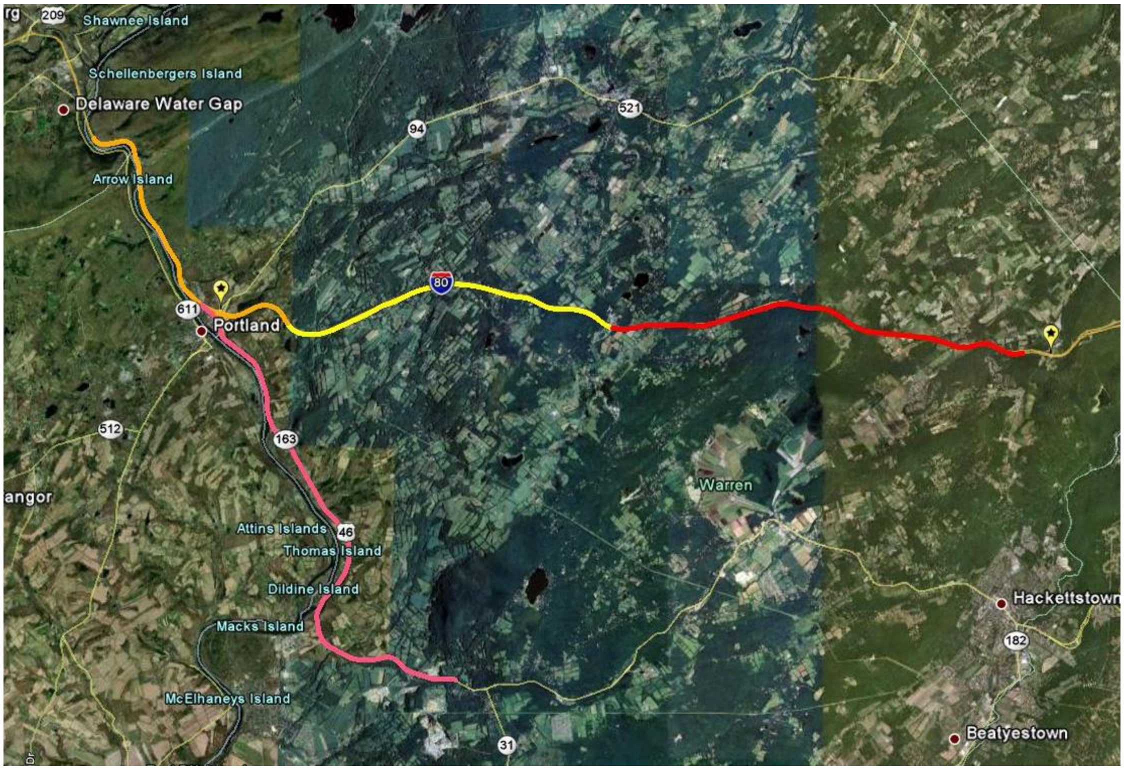


Figure 3. NJ-MDSS road sections

Weather Forecast Sites

NJ-MDSS retrieved weather forecast data from DTN/Telvent every hour. Table 3 lists the forecast sites in NJ-MDSS.

Weather Observation Sites

Road Weather Information System (RWIS) is the source for weather and road surface conditions. There are 36 NJDOT's RWIS sites listed in Table 4. The data collected by two RWIS sites near the routes within the studied region are used in computing road condition and recommending road treatment within the study service area: one is located at Allamuchy Township (I-80 MP 20, at CR 517 Interchange), and the other is located at Knowlton Township (I-80 MP 4.6, near NJ 94 overpass). All data collected by the RWIS sites are fed into NJ-MDSS for displaying weather and road surface observations.

Table 3 Weather forecast sites in NJ-MDSS

Name	**County**	**State**
Atlantic City International Airport	Atlantic	NJ
Teterboro Airport	Bergen	NJ
Mount Holly, South Jersey Regional Airport	Burlington	NJ
Mcguire AFB	Burlington	NJ
Monmouth Beach	Burlington	NJ
Wildwood	Cape May	NJ
Millville Municipal Airport	Cumberland	NJ
Newark International Airport	Essex	NJ
Caldwell/Fairfield	Essex	NJ
Trenton/Mercer Airport	Mercer	NJ
Belmar-Farmdale	Monmouth	NJ
Morristown Airport	Morris	NJ
Lakehurst NAS	Ocean	NJ
Toms River	Ocean	NJ
Somerville, Somerset Airport	Somerset	NJ
Aero/Andover Airport	Sussex	NJ
Sussex	Sussex	NJ
Allamuchy Township RWIS	Warren	NJ
Knowlton Township RWIS	Warren	NJ
Newburgh/Stewart International Airport	Orange	NY
Montgomery, Orange County	Orange	NY
Monticello	Sullivan	NY
Doylestown	Bucks	PA
Quakertown Airport	Bucks	PA
Allentown	Lehigh	PA
Wilkes-Barre/Scranton	Luzerne	PA
Mount Pocono Airport	Monroe	PA
Philadelphia International Airport	Philadelphia	PA
Northeast Philadelphia Airport	Philadelphia	PA

Table 4 NJDOT RWIS sites

MDSS ID	RWIS Site Name	County	Location
72407050	Lower Township	Cape May	NJ 47 at MP 1.0
72407051	Maurice Township 1	Cumberland	NJ 47 & 347
72407052	Maurice Township 2	Cumberland	NJ 55 & Schooner Landing Road
72407053	Atlantic City (NJDOT)	Atlantic	NJ 30 at Beach Thorofare
72407054	Hamilton Twp. (Atlantic)	Atlantic	NJ 322 & 50
72408180	Carneys Point	Salem	NJ 295 & 140
72517142	Greenwich Township	Gloucester	I-78 at MP 6.7
72408182	Logan Township	Gloucester	NJ 295 & Oldmans Creek
72408181	Monroe Township	Gloucester	NJ 322 & 42
72409072	Mansfield Township	Burlington	NJ 68 at MP 4.1
72409069	Pemberton Township	Burlington	NJ 70 at MP 26.4
72409071	Springfield Township	Burlington	I-295 at Springfield Rest Area
72408183	Cherry Hill Township	Camden	I-295 & 70
72409070	Dover Township	Ocean	NJ 37 & Bayview Ave
72409078	Eatontown Borough	Monmouth	NJ 18 & 36
72409076	Freehold Township	Monmouth	NJ 9 & 33
72409073	Howell Township	Monmouth	NJ 9 & I-195
72409074	Upper Freehold Twp.	Monmouth	I-195 at MP 10.2
72409077	Ewing Township	Mercer	NJ 29 & I-95
72409075	Hamilton Twp. (Mercer)	Mercer	I-295 &I-195
72409079	West Windsor Township	Mercer	NJ 1 & Alexander Road
72502192	New Brunswick (DOT)	Middlesex	NJ 1 at Raritan River
72502191	Old Bridge Township	Middlesex	NJ 9 & 18
72502193	Piscataway	Middlesex	NJ 18 & River Road
72502194	Woodbridge Township	Middlesex	NJ 9 & 440
72502195	Warren Township	Somerset	I-78 & CR 525
72502199	West Orange 1	Essex	I-280 at Pleasant Valley Road
72502200	West Orange 2	Essex	I-280 & Prospect Avenue
72502198	Chester Township	Morris	NJ 206 at Lushans Hill
72502201	Mount Olive Township	Morris	NJ 46 at Hackettstown Hill
72502202	Wanaque Borough	Passaic	I-287 at Wanaque River Bridge
72502203	Lafayette Township	Sussex	NJ 94 at Paulins Kill River
72502196	Berkeley Heights	Union	I-78 at Green Brook
72502197	Summit	Union	NJ 24 & River Road
72502184	Allamuchy Twp.	Warren	I-80 at CR 517
72517139	Knowlton Township	Warren	I-80 & SR 94

MDSS Database Development

To develop a suitable database schematic for NJ-MDSS, a comprehensive review of prototype MDSS input forms was conducted, illustrated in APPENDIX B, including:

- Route Characteristics Information
- Pavement Section Information
- Traffic Counts
- Treatment Information
- Route List in Studied Area

Table 5 NJ-MDSS modules and forms

Modules	**Types of Forms**
Road Condition and Treatment Module (RCTM)	Route List in Studied Area Route Characteristics Information Pavement Information Traffic Counts
Frost Potential Algorithm	Route List in Studied Area Pavement Information
Treatment Update Network Layer (TUNL)	Treatment Information

Collect Data

Several types of live data are required by NJ-MDSS. These data include weather forecasts, weather and road surface observations, and grid data sets such as radar and satellite images. Some of these data sets are made available to a display after simple reformatting, but many are used by the NJ-MDSS server processes in creating weather forecasts and alerts for the studied route sections. The programs/scripts for downloading, reformatting, or creating the raw data sets are illustrated in APPENDIX C.

The data entered into NJ-MDSS are obtained from a variety of sources (see table 6), including weather forecasts and observations, traffic camera images, and GPS-based AVL (Automatic Vehicle Location) data from DOT's maintenance trucks.

Table 6 NJ-MDSS data inputs

Data	**Source**
Weather forecast data	DTN/Telvent
Weather observation data	*Clarus*
Road condition observation data	DTN/Telvent
Traffic camera images	Vaisala, 511NJ.org
GPS-based AVL data	NJDOT
Route Characteristics data (e.g., pavement, traffic count)	NJDOT
GIS data	Web Map Server

All forecast and observation data are ultimately placed in a database which is accessible to internal NJ-MDSS processes and external display clients via TUNL or THREDDS. THREDDS provides grid data such as satellite and radar; TUNL is an interface which provides all other weather observation and forecast data.

The GIS data such as state and county maps, road, and topography are accessed dynamically at run-time from WMS (Web Map Server), which allows far more flexibility in relocating the system to a new geographic domain, as well as allowing the system to display data on different scales.

Development of NJ-MDSS

Extensive discussion about NJ-MDSS development took place between the Research Team and the RPSIP, yielding critical guidelines to develop an effective system, including:

- Incorporating NJDOT Maintenance's rules of practice for winter storm management;
- Providing real-time winter weather information from separate sources for both the atmosphere and road surface, making maximum use of this information;
- Combining data to create an understandable and effective display of current atmospheric, roadway, and environmental conditions, including GIS mapping for selected data queries;
- Providing winter maintenance treatment options with various application rates and times, and chemical types, and recommend a best treatment option;
- Displaying and evaluating the results (outcome) of multiple treatment options by including a "what if" scenario treatment selector to analyze how road conditions would change with different treatments for a given weather and road surface condition;
- Archiving data and displaying archived data and products;
- Applying historical data for demonstration, training, and analysis purposes; and
- Providing a large quantity of quality data in an efficient and effective manner.

APPENDIX D lists computer hardware and software requirements to operate and perform NJ-MDSS effectively.

Implementation and Training

Once NJ-MDSS was ready to be implemented in the study area, the associated tutorials and training were provided to NJDOT maintenance staff. The functions associated with NJ-MDSS are the following (discussed below):

- Web-based NJ-MDSS user account Management
- Training of NJDOT Personal

Web-based NJ-MDSS User Account Management

A web-based user account management system was developed to allow NJDOT personnel to access the NJ-MDSS system with different privileges based on user's level (See Table 7). Also adding/removing NJ-MDSS user account and changing users' password can be accomplished on this web-based management system.(See Figure 4 and Figure 5).

Table 7 NJ-MDSS user's privileges

		Change current treatment plans	
		Yes	No
View current treatment plans	Yes	Level A	Level B
	No	n/a	Level C

- Level A: able to view and change current treatment plans;
- Level B: able to view but not to change current treatment plans;
- Level C: unable to view and change treatment plans;

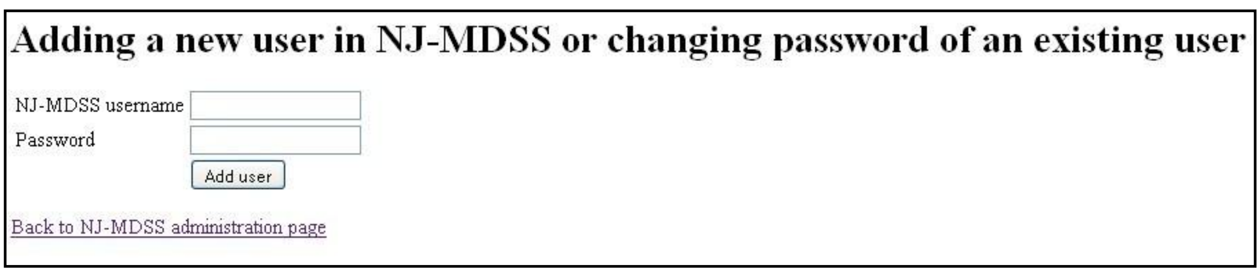


Figure 4. Web-based user account management: add user account

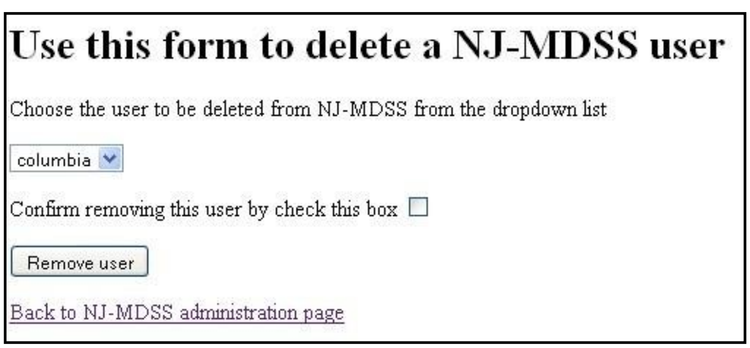


Figure 5.Web-based user account management: delete user account

Training of NJDOT Personnel

Training in the use of NJ-MDSS is provided to NJDOT Maintenance staff. NJDOT staff has used the NJ-MDSS for winter snow/ice events from 2009 to 2011.A User Tutorial of NJ-MDSS was developed by the research team. (See APPENDIX E)

CONCLUSIONS AND RECOMMENDATIONS

Summary of NJ-MDSS Benefits

The NJ-MDSS application is equipped with exceptional functionalities for winter storm road maintenance operation and planning. The benefits of implementing NJ-MDSS are summarized as follows:

- Provide real-time winter weather information for both atmospheric and road surface conditions, and present a large quantity of data in an efficient and effective manner
- Integrate a variety of critical weather data into a GIS-based platform, which can assist NJDOT to make timely winter road treatment decisions, including the timing, rate, and type of de-icing materials application
- A consolidated set of recommended treatment alternatives, which are easily accessible to the crews, to support winter road treatment decisions
- Allow NJDOT to customize road treatments to the particular needs and locations of maintenance yards, considering maintenance crews' observation of road surface temperature with trend forecasts
- Act as an important learning and training tool by evaluating "what if" scenarios in road conditions with different treatment applications

Implementation of NJ-MDSS shall be a useful tool for further studies related to Capital Improvements Planning, Transportation Planning, Traffic and Maintenance Operations, Traveler Information, and Incident Management (accidents and work zones).

Future Improvement of NJ-MDSS

The NJ-MDSS application can be further enhanced with information collected by the sensors located on the maintenance trucks, such as vehicle location, material type, spinner rate, air and road surface temperatures, etc.

Computer programs developed in this study can be enhanced to extract the data and display the real time information via NJ-MDSS as soon as a wireless data communication network is created.

In addition to displaying truck location, radar and satellite images will be a great asset to the maintenance managers in determining weather conditions and predicting storm tracks. A NJ-MDSS client can overlay radar and satellite images on the displayed map. The THREDDS (Thematic Real-time Environmental Distributed Data Services) module installed in NJ-MDSS R6 can process and display an area of observation, including radar and satellite images.

Overall, the implementation of NJ-MDSS shall be an important tool to NJDOT to support winter road maintenance operations.

REFERENCES

1. Andrle, S. J., Kroeger, D. A., and Sinhaa, R., (2003), “Deploying the Maintenance Decision Support System (MDSS) in Iowa,” Center for Transportation Research and Education, Iowa State University, Ames, IA.
2. ASOS (2007),“User Guide,” http://www.nws.noaa.gov/om/brochures/asosbook.htm, National Weather Service.
3. ASTM, (1994), Standard Specification for Highway Weigh-in-Motion (WIM) Systems with User Requirements and Test Methods, E 1318-94.
4. Automated Weather Observing System (ASOS), (2007), Federal Aviation Administration (FAA) http://www.faa.gov/asos/awosinfo.htm
5. Autoscope solo, (2003), Econolite control Products, Inc., www.econolite.com/Product/autoscope/solopro.htm.
6. Baldwin, M., R. Treadon, and S. Contorno, (1994), “Precipitation type prediction using a decision tree approach with NMCs mesoscale eta model,” 10th Conf. On Numerical Weather Prediction, Portland, OR, Amer. Meteor. Soc., 30-31.
7. Bonneson, J., and Abbas, M., (2002), “Video Detection For Intersection and Interchange Control,” Texas Transportation Institute, The Texas A&M University System, Report 4285-1, College Station, TX.
8. Boschun (2007), “Boschung is currently implementing the largest Maintenance Decision Support System (MDSS) project in the world,” http://www.boschungamerica.com/pages/mdss.php
9. Bourgoin, P., (2000), “A method to determine precipitation types,” Wea. Forecasting, 15, 583-592.
10. Cambridge Systematics, (2003),“Weather-Responsive Traffic Management: Concept of Operations,” draft report prepared for FHWA, U.S. Department of Transportation.
11. Chen, C., Kwon, J., Rice, J., Skabardonis, A., and Varaiya, P., (2003), “Detecting Errors and Imputing Missing Data for Single Loop Surveillance Systems,” 82nd Transportation Research Board Annual Meeting, Washington, DC.
12. Chin, S., et al., (2002), “Temporary Losses of Highway Capacity and Impacts on Performance,” ORNL/TM-2002/3, Oak Ridge National Laboratory.
13. Coifman, B., (1999), “Using Dual Loop Speed Traps to Identify Detector Errors,” Transportation Research Board Annual Meeting, Washington, DC.
14. Cortinas, J.V., Jr., and M. E. Baldwin, (1999), “A preliminary evaluation of six precipitationtype algorithms for use in operational forecasting,” 6th Workshop on Operational Meteorology, Halifax, Nova Scotia, Environment Canada, 207-211.
15. Cluett, C., and Jenq J., (2007), “A Case Study of MDSS in Maine,” FHWA-JPO-08-001
16. Ener, O., Jacobson, R., and Sowell, H. W., (2001), “Colorful Future, World Highways/ Routes,” Du Monde, Image Sensing Systems, Inc., www.imagesensing.com/technical_papers.htm

17. Federal Meteorological Handbook (FMH-1), (1995), “Surface Weather Observations and Reports,” Office of Federal Coordinator for Meteorology, U.S. Department of Commerce.

18. FHWA, (1996), “Maintenance Personnel,” FHWA-RD-95-202.

19. FHWA, (1998), “Test and Evaluation Project No. 28: Anti-icing Technology, Field Evaluation Report,” FHWA-RD-97-132.

20. FHWA, (2003), “Best Practices for Road Weather Management,” Version 2.0, prepared by Mitretek Systems, FHWAOP-03-081.

21. FHWA, (2007), FHWA Road Weather Management Program Projects and Activities, http://ops.fhwa.dot.gov/weather/mitigating_impacts/programs.htm#p3

22. Gonzalez, A., Papagiannakis, A. T., and O'Brien, E. J., (2003), “Evaluation of an Artificial Neural Network Technique Applied to Multiple-Sensor Weigh-In-Motion Systems,” Annual Meeting of Transportation Research Board, Washington, DC.

23. Goodwin, L., (2002), “Analysis of Weather-Related Crashes on U.S. Highways,”Mitretek Systems.

24. Gordon, R. L., (1996), “Traffic Control Systems Handbook,” Dunn Engineering Associates, Publication No. FHWA-SA-95-032, FHWA, U.S. DOT, p. 4-18.

25. Gordon, R.L., Reiss, R., A., Haenel, H., Case, E. R., French, R. L., Mohaddes, A., and Wolcott, R., (1996), “Traffic Control Systems Handbook,” FHWA-SA-95-032, Federal Highway Administration, U.S. Department of Transportation, Washington, DC.

26. Hughes, R., Huang, H., Zegeer, C., and Cynecki, M., (2001), “Evaluation of Automated Pedestrian Detection at Signalized Intersections,” FHWA-RD-00-097, Washington, DC.

27. Kell, J. H., Fullerton, I. J., and Mills, M. K, (1990), “Traffic Detector Handbook,” FHWA-IP-90-002, Federal Highway Administration, Washington, DC.

28. Ketcham, S., D. Minsk, R. Blackburn, and E. Fleege, (1996),“Manual of Practice for Effective Anti-Icing Program: A Guide for Highway Winter Maintenance Personnel.” US Army Cold Regions Research and Engineering Laboratory.

29. Kim, S. W., Kim, K., Lee, J. H., and Cho, D., (2001), “Application of Fuzzy Logic to Vehicle Classification Algorithm in Loop/Piezo-sensor Fusion Systems,” Asian Journal of Control, Vol. 3, No. 1, pp. 64-68.

30. Kistler Instruments Corporation, (2000), “Piezoelectric Force Transducers, Design & Use,” Amherst, NY.

31. Klein, L. A. (2001), “Sensor Technologies and Data Requirements for ITS,” Norwood, MA: Artech House, pp. 549.

32. Knapp, et al, (2000), “Winter Storm Event Volume Impact Analysis Using Multiple Source Archived Monitoring Data,” Transportation Research Board (TRB) Annual Meeting.

33. Kroeger, D. A. and Burkheimer, D., (2003), “Deploying the Winter Maintenance Support System (MDSS) in Iowa,” Proceedings of the 2003 Mid-Continent Transportation Research Symposium, Ames, IA.

34. Kwon, J., Varaiya, P., and Skabardonis, A., (2003), "Estimation of Truck Traffic Volume from Single Loop Detector Using Lane-to-lane Speed Correlation," 82nd Transportation Research Board Annual Meeting, Washington, DC.

35. MacCarley, C. A., Hockaday, S., Need, D., and Taff, S., (1992), "Evaluation of Video Image Processing Systems for Traffic Detection," Transportation Research Record No. 1360, National Research Council, Washington, DC.

36. Macro Corporation, (2004), "Arizona Phase II Final Report: Statewide Radio Interoperability Needs Assessment," Macro Corporation and The State of Arizona, pp. 165.

37. Mahoney, W.P, W. Myers, (2003), "The Winter Road Maintenance Decision Support System (MDSS) Project Update and Future Plans," 19th Conf. on Interactive Information and Processing Systems, Long Beach, CA, Amer. Meteor. Soc., 10.4, 2.

38. Mahoney III, W. P., et al., (2005) "FHWA's maintenance decision support system project: Results and recommendations," Transportation Research Record, No.1911, pp 133-142, 2005.

39. Mahoney, B., (2007), "Winter Road Maintenance Decision Support Winter Road Maintenance Decision Support System (MDSS) Prototype Development".

40. McCord, M. R., Yang, Y., Jiang, Z., Coifman, B., and Goel, P. K., (2003), "Estimating AADT from Satellite Imagery and Air Photos: Empirical Results," Annual Meeting of Transportation Research Board, Washington, DC.

41. McGinley, J., S. Albers, D. Birkenheuer, B. Shaw, P. Shultz, (2000), "The LAPS Water in All Phases Analysis: the Approach and Impacts on Numerical Prediction, 5th International Symposium on Tropospheric Profiling," Adelaide, Australia.

42. Mewes, J. J., Hart, R., Osborne, L., & Podoll, B., (2005),"The pooled fund study maintenance decision support system: A functional overview."

43. Motorola Communications and Electronics, (1979), "T1919A Metrocom II 150.8-174 MHz Vehicle Location Receiver, Schaumburg, IL.

44. National Center for Atmospheric Research (NCAR), (2002), "MDSS FUNCTIONAL PROTOTYPE OVERVIEW DESCRIPTION," Report to Federal Highway Administration (FHWA) on Maintenance Decision Support System (MDSS), Functional Prototype Development Project, Boulder, CO

45. National Center for Atmospheric Research (NCAR), (2003) "Maintenance Decision Support System (MDSS) Functional Prototype Development Project."

46. National Center for Atmospheric Research (NCAR), (2003), "The Maintenance Decision Support System (MDSS) Project Iowa Field Demonstration and Evaluation Plan," Version 0.1, 19, Boulder, CO., September 2003.

47. National Center for Atmospheric Research (NCAR), (2003), "The Maintenance Decision Support System (MDSS) Project, Technical Performance Report, Iowa Field Demonstration," Version 1.1, 12, Boulder, CO.

48. National Center for Atmospheric Research (NCAR), (2004), "The Maintenance Decision Support System (MDSS) Project, MDSS Prototype, Release-2, Technical Description, Version 1.0," Report to Federal Highway Administration, Office of Transportation Operations Road Weather Management Program, Boulder, CO.
49. National Center for Atmospheric Research (NCAR), (2007), "The Maintenance Decision Support System (MDSS) Project, MDSS Prototype, Release-5, Technical Description, Version 1.1," Report to Federal Highway Administration, Office of Transportation Operations Road Weather Management Program, Boulder, CO.
50. Nelson, G. G., (2001), "Maintenance Decision Support System (MDSS)," Second MDSS Prototype Review Hanover, NH June 21, 2001
51. Nooralahiyan A., Kirby, H., and McKeown, D., (1998), "Vehicle Classification by Acoustic Signature," Mathematical and Computer Modeling, vol. 27, no. 9-11, pp. 205-214.
52. Novax, (2007), "Lane King Detector," Novax Industries Corporation, www.novax.com.
53. Oh, S., Ritchie, S. G., and Oh, C., (2002), "Real Time Traffic Measurement from Single Loop Inductive Signatures, 81st Annual Meeting of the Transportation Research board, Washington, DC.
54. ORNL, (2003), "DP 121 Weigh-in-Motion Technology," Oak Ridge National Laboratory, www.ornl.gov/dp121.
55. Osborne, Jr., L. F., (2004), "Evolution of a Multi-State Maintenance Decision Support System," itsmn.org/ruralits2004/presentations/Osborne.pdf
56. Pack, M. L., Smith, B. L., and Scherer, W. T., (2003), "An Automated Camera Repositioning Technique for Integrating Video Image Vehicle Detection Systems with Freeway CCTV Systems," Annual Meeting of TRB, Washington, DC.
57. Panda, D. P., and Michalopoulos, P. G., (1998), Deployment of the Next Generation Machine Vision Technology and Integration with SCOOT in the City of Minneapolis," 5th ITS World Congress, Washington, DC.
58. Perchanok, M., Comfort, G., and Wahlgren, C., (2000), ''Decision Support System for Winter Maintenance: Feasibility Demonstration."
59. Perrin, et al., (2002), "Testing the Adverse Visibility Information System Evaluation (ADVISE) - Safer Driving in Fog," Transportation Research Board (TRB) Annual Meeting.
60. Pisano, P. A., Stern, A. D., & Mahoney III, W. P., (2005), "The U.S. federal highway administration winter road maintenance decision support system (MDSS) project: Overview and results."
61. Potter, T., (2003), "The Evolution of Inductive Loop Detector Technology," Reno A&E,www.renoae.com/Documentation/MISC/Advances%20in%20Loop%20Detector%20Technology.pdf.
62. Pushkar, A., Hall, F. L., and Acha-daza, J. A., (1994), "Estimation of Speeds from Single-Loop Freeway Flow and Occupancy Data Using Cusp Catastrophe Theory Model," Transportation Research Record 1457, p149-157, Washington, DC.

63. Ramer, J., (1993), “An empirical technique for diagnosing precipitation type from model output,” 5th International Conf. On Aviation Weather Systems, Vienna, VA, Amer. Meteor. Soc., 227-230.

64. Road Runner (2000), “SPVD-1 - Detector/Transmitter Quick Installation Guide,” the Road Runner System, Midian Electronics, Inc.,

65. RTMS, (2007), “Traffic Solutions,” Electronic Integrated Systems Inc., www.rtms-by-eis.com.

66. Safaai-Jazi, A., Ardekani, S. A., and Mehdikhani, M., (1990), “A Low-Cost Fiber Optic Weigh-in-Motion Sensor,” SHRP-ID/UFR-90-002, Strategic Highway Research Program, National Research Council, Washington, DC.

67. Takle, E., and Greenfield, T., (2005), “Test and Validation of a Model for Forecasting Frost on Bridges,” Iowa Department of Transportation

68. Transportation Research Board, Highway Capacity Manual, National Cooperative Highway Research Program (NCHRP) Report 209, U.S. DOT, 2000, Chapter 22.

69. Turner, S. M., Eisele, W. L., Benz, R. J., and Douglas J., (1998), “Travel Time Data Collection Handbook,” FHWA-PL-98-035, Texas Transportation Institute, College Station, TX.

70. U.S. Department of Commerce, (2007), http://www.ofcm.gov/fmh-1/fmh1.htm

71. U.S. DOT, (2002), “Motor Carrier Management Information System (MCMIS) Crash File,” http://ai.volpe.dot.gov/CrashProfile/datasource.asp.

72. USDOT, (2007), www.itsdocs.fhwa.dot.gov//JPODOCS/REPTS_TE/13589.html Assessment of the Denver Regional Transportation District's Automatic Vehicle Location System, US Department of Transportation.

73. U.S. Geological Survey (USGS). (2002). U.S. Geological Survey Minerals Yearbook, Center, Miss.

74. Western Transportation Institute, and Iteris Inc.,(2006) “ Analysis of Maintenance Decision Support System Benefits & Costs,” Study SD2006-10, Final Report

APPENDIX A –FWHA AND NJDOT SNOW MAINTENANCE GUIDELINES

Anti-icing treatment - FHWA guidelines

PAVEMENT	INITIAL OPERATION				SUBSEQUENT OPERATIONS					COMMENTS
TEMPERATURE RANGE, AND TREND	**pavement surface at time of**	**maintenance action**	**dry chemical spread rate, kg/lane-km (lb/lane-mi)**		**maintenance action**	**dry chemical spread rate, kg/lane-km (lb/lane-mi)**				
	initial operation		**liquid**	**solid or pre wetted**		**liquid**		**solid or pre wetted solid**		
				solid		**light snow**	**heavier snow**	**light snow**	**heavier snow**	
Above 0^{o}C (32^{o}F), steady or rising	Dry, wet, slush, or light snow cover	None, see comments			None, see comments					1) Monitor pavement temperature closely for drops toward 0^{o}C (32^{o}F) and below 2) Treat icy patches if needed with chemical at 28 kg/lane-km (100 lb/lane-mi); plow if needed
Above 0^{o}C (32^{o}F), 0^{o}C (32^{o}F) or below is imminent;	Dry	Apply liquid or pre wetted solid chemical	28 (100)	28 (100)	Plow as needed; reapply liquid or solid	28 (100)	55 (200)	28 (100)	55 (200)	1) Applications will need to be more frequent at lower temperatures and higher snowfall rates 2) Do not apply liquid chemical onto
ALSO **-4 to 0^{o}C** **(25 to 32^{o}F)**, remaining in range	Wet, slush, or light snow cover	Apply liquid or solid chemical	28 (100)	28 (100)	chemical when needed					heavy snow accumulation or packed snow 3) After heavier snow periods and during light snow fall, reduce chemical rate to 28 kg/lane-km (100 lb/lane-mi); continue to plow and apply chemicals as needed
-10 to -4^{o}C **(15 to 25^{o}F)**, remaining in range	Dry, wet, slush, or light snow cover	Apply pre wetted solid chemical		55 (200)	Plow as needed; reapply pre wetted solid chemical when needed			55 (200)	70 (250)	1) If sufficient moisture is present, solid chemical without pre wetting can be applied 2) Reduce chemical rate to 55 kg/lane-km (200 lb/lane-mi) after heavier snow periods and during light snow fall; continue to plow and apply chemicals as needed
Below -10^{o}C (15^{o}F), steady or falling	Dry or light snow cover	Plow as needed			Plow as needed					1) It is not recommended that chemicals be applied in this temperature range 2) Abrasives can be applied to enhance traction

Weather event: frost or black ice

PAVEMENT TEMPERATURE RANGE AND TREND, AND RELATION TO DEW POINT	INITIAL OPERATION			SUBSEQUENT OPERATIONS		COMMENTS
	Pavement surface at time of initial operation	**Maintenance action**	**Brine application rate in Gallons per Lane Mile**	**Maintenance action**	**Brine application rate in Gallons per Lane Mile**	
Above 32 degrees Fahrenheit. Steady or rising	Frost or black ice or forecast for frost or black ice.	See comments		See comments		1) Monitor pavement temperature closely. Begin treatment if temperatures to fall to 32 degrees Fahrenheit or and is at or below the dew point. Apply brine at 25 gallons per lane mile.
28 to 32 degrees Fahrenheit Remaining in that range or falling to 32 degrees Fahrenheit or below, and equal to or below the dew point.		Apply brine	25 gallons per lane mile (57.5 lb/lane mile) *25-65 lb/lane mile*	Reapply brine when needed.	40 gallons per lane mile (92 lb/lane mile) *40-115 lb/lane mile*	1) Monitor pavement closely. If pavement becomes wet or if thin ice forms, reapply brine at a higher application rate. 2) Do not apply brine on ice so thick that the pavement cannot be seen.
20 to 28 degrees Fahrenheit Remaining in that range and equal to or below the dew point. **15 to 20 degrees Fahrenheit** Steady or falling		Apply brine	50 gallons per lane mile (115 lb/lane mile) *65-130 lb/lane mile*	Apply brine	50 gallons per lane mile (115 lb/lane mile) *65-130 lb/lane mile*	1) Monitor pavement closely. If thin ice forms, reapply brine at higher application rates. 2) Applications will need to be more frequent at higher levels of condensation if traffic volumes are not enough to disperse condensation. 3) It is not advisable to apply brine at the indicated application rate when the pavement temperature drops below 20 degrees Fahrenheit.
		Do not apply liquids	Not applicable	**Do not apply liquids**	Not applicable	Apply only prewetted solid chemical (salt) in this temperature range.
Below 15 degrees Fahrenheit Steady or falling		Do not apply liquids	Not applicable	Do not apply liquids	Not applicable	It is not recommended that chemicals be applied in this temperature range.

Note:
BRINE APPLICATIONS. (1) Conduct initial operation in advance of freezing. Apply brine up to 3 hours in advance. Use longer advance times in this range to effect drying when traffic volume is low. (2) In the absence of precipitation, brine at a rate of 60 gallons per lane miles has been successful in preventing bridge deck icing when placed up to 4 days before freezing on higher volume roads and 7 days before on lower volume roads. (3.) It is recommended that you **do not exceed 30 MPH when applying Salt Brine.**

Weather event: light snowstorm

PAVEMENT TEMPERATURE RANGE AND TREND	INITIAL OPERATION			SUBSEQUENT OPERATIONS		COMMENTS
	Pavement surface at time of initial operation	**Maintenance action**	**Brine application rate in Gallons per Lane Mile**	**Maintenance action**	**Brine application rate in Gallons per Lane Mile**	
Above 32 degrees Fahrenheit. Steady or rising	Dry, wet, slush, or light snow cover	None, see comments		None, see comments		1) Monitor pavement temperature closely for drops toward 32 degrees Fahrenheit and below. 2) Treat icy patches if needed with brine at 40 gallons per lane mile. Plow if needed.
Above 32 degrees Fahrenheit 32 degrees Fahrenheit or below is imminent;	Dry	Apply brine	40 gallons per lane mile (92 lb/lane mile) *100 lb/lane mile*	Plow as needed. Reapply brine when needed.	40 gallons per lane mile (92 lb/lane mile) *100 lb/lane mile*	1) Application will need to be more frequent at lower temperatures and higher snowfall rates. 2) It is not advisable to apply brine at the indicated spread rate when the pavement temperature drops below 20 degrees Fahrenheit. 3) Do not apply brine onto heavy snow accumulation or packed snow.
ALSO **20 to 32 degrees Fahrenheit** Remaining in that range	Wet, slush, or light snow cover	Apply brine	40 gallons per lane mile (92 lb/lane mile) *100 lb/lane mile*			
15 to 20 degrees Fahrenheit Remaining in that range	Dry, wet, slush or light snow cover	**Do not apply liquids**	Not applicable	**Do not apply liquids**	Not applicable	Apply only prewetted solid chemicals (salt) in this temperature range
Below 15 degrees Fahrenheit Steady or falling	Dry or light snow cover	Plow as needed	Not applicable	Plow as needed	Not applicable	It is not recommended that brine be applied in this temperature range.

Note:
BRINE APPLICATIONS. (1) Time initial and subsequent brine applications to prevent deteriorating conditions or development of packed and bonded snow. (2) Apply brine ahead of traffic rush periods occurring during storm. (3.) It is recommended that you **do not exceed 30 MPH when applying Salt Brine.**

PLOWING. If needed, plow before brine applications so that excess snow, slush, or ice is removed and pavement is wet, slushy, or lightly snow covered when treated.

Weather event: light snowstorm with periods of moderate or heavy snow

PAVEMENT TEMPERATURE RANGE AND TREND	INITIAL OPERATION			SUBSEQUENT OPERATIONS			COMMENTS
	Pavement surface at time of initial operation	**Maintenance action**	**Brine application rate in Gallons per Lane Mile**	**Maintenance action**	**Brine application rate in Gallons per Lane Mile**		
					Light snow	**Heavier snow**	
Above 32 degrees Fahrenheit. Steady or rising	Dry, wet, slush, or light snow cover	None, see comments		None, see comments			1) Monitor pavement temperature closely for drops toward 32 degrees Fahrenheit and below. 2) Treat icy patches if needed with brine at 40 gallons per lane mile. Plow if needed.
Above 32 degrees Fahrenheit 32 degrees Fahrenheit or below is imminent;	Dry	Apply brine	40 gallons per lane mile (92 lb/lane mile) *100 lb/lane mile*	Plow as needed. Reapply brine when needed.	40 gals/lm (92 lb/lane mile) *100 lb/lane mile*	80 gals/lm (184 lb/lane mile) *200 lb/lane mile*	1) Application will need to be more frequent at lower temperatures and higher snowfall rates. 2) Do not apply brine onto heavy snow accumulation or packed snow. 3) After heavier snow periods and during light snowfall, reduce brine rate to 40 gallons per lane mile. Continue to plow and apply brine as needed.
ALSO **20 to 32 degrees Fahrenheit** Remaining in that range	Wet, slush, or light snow cover	Apply brine	40 gallons per lane mile (92 lb/lane mile) *100 lb/lane mile*				
15 to 20 degrees Fahrenheit Remaining in that range	Dry, wet, slush or light snow cover	**Do not apply liquids**	Not applicable	**Do not apply liquids**			Apply only prewetted solid chemical (salt) in this temperature range
Below 15 degrees Fahrenheit Steady or falling	Dry or light snow cover	Plow as needed	Not applicable	Plow as needed			It is not recommended that brine be applied in this temperature range.

Note:
BRINE APPLICATIONS. (1) Time initial and subsequent brine applications to prevent deteriorating conditions or development of packed and bonded snow. (2) Anticipate increases in snowfall intensity and apply higher application rates prior to or at the beginning of heavier snowfall periods to prevent development of packed and bonded snow. (3) Apply brine ahead of traffic rush periods occurring during storm. (4.) It is recommended that you **do not exceed 30 MPH when applying Salt Brine.**

PLOWING. If needed, plow before brine applications so that excess snow, slush, or ice is removed and pavement is wet, slushy, or lightly snow covered when treated.

Weather event: moderate or heavy snow

PAVEMENT TEMPERATURE RANGE AND TREND	INITIAL OPERATION			SUBSEQUENT OPERATIONS		COMMENTS
	Pavement surface at time of initial operation	**Maintenance action**	**Brine application rate in Gallons per Lane Mile**	**Maintenance action**	**Brine application rate in Gallons per Lane Mile**	
Above 32 degrees Fahrenheit. Steady or rising	Dry, wet, slush, or light snow cover	None, see comments		None, see comments		1) Monitor pavement temperature closely for drops toward 32 degrees Fahrenheit and below. 2) Treat icy patches if needed with brine at 40 gallons per lane mile. Plow if needed.
Above 32 degrees Fahrenheit 32 degrees Fahrenheit or below is imminent;	Dry	Apply brine	40 gallons per lane mile (92 lb/lane mile) *100 lb/lane mile*	Plow as needed. Reapply brine when needed.	40 gallons per lane mile (92 lb/lane mile) *100 lb/lane mile*	1) If the desired plowing/treatment frequency cannot be maintained, the application rate can be increased to 80 gallons per lane mile to accommodate longer operational cycles. 2) Do not apply brine onto heavy snow accumulation or packed snow.
ALSO **20 to 32 degrees Fahrenheit** Remaining in that range	Wet, slush, or light snow cover	Apply brine	40 gallons per lane mile (92 lb/lane mile) *100 lb/lane mile*			
20 to 30 degrees Fahrenheit Remaining in that range	Dry	Apply brine	80 gallons per lane mile (184 lb/lane mile) *200 lb/lane mile*	Plow as needed. Reapply brine when needed	80 gallons per lane mile (184 lb/lane mile) *200 lb/lane mile*	1) If the desired plowing/treatment frequency cannot be maintained, the application rate can be increased to accommodate longer operational cycles. 2) Do not apply brine onto heavy snow accumulation or packed snow.
	Wet, slush, or light snow cover	Apply brine	80 gallons per lane mile (184 lb/lane mile) *200 lb/lane mile*			
15 to 20 degrees Fahrenheit Remaining in that range	Dry, wet, slush or light snow cover	**Do not apply liquids**	Not applicable	**Do not apply liquids**	Not applicable	Apply only prewetted solid chemical (salt) in this temperature range.
Below 15 degrees Fahrenheit Steady or falling	Dry or light snow cover	Plow as needed	Not applicable	Plow as needed	Not applicable	It is not recommended that brine be applied in this temperature range.

Note:
BRINE APPLICATIONS. (1) Time initial and subsequent brine applications to prevent deteriorating conditions or development of packed and bonded snow. (2) Apply brine ahead of traffic rush periods occurring during storm. (3.) It is recommended that you **do not exceed 30 MPH when applying Salt Brine.**

PLOWING. If needed, plow before brine applications so that excess snow, slush, or ice is removed and pavement is wet, slushy, or lightly snow covered when treated.

APPENDIX B – NJ-MDSS FORMS FOR DATABASE DEVELOPMENT

Form 1. Route Information

Milepost		Longitude	Latitude	Elevation	Number of Lanes	Route Length	Time Req to Treat (minutes)
From	To						

Form 2. Pavement information

Pavement		
Layer#	Material	Thickness (m)
1		
2		
3		
4		

Material types: Clay, Sand, Loam, Concrete (Cement), Aggregate base (Crushed rock), and Asphalt

Form 3. Traffic counts

Traffic Level											
12 midnight	1:00 AM	2:00 AM	3:00 AM	4:00 AM	5:00 AM	6:00 AM	7:00 AM	8:00 AM	9:00 AM	10:00 AM	11:00 AM
12 noon	1:00 PM	2:00 PM	3:00 PM	4:00 PM	5:00 PM	6:00 PM	7:00 PM	8:00 PM	9:00 PM	10:00 PM	11:00 PM

Traffic Level											
12 midnight	1:00 AM	2:00 AM	3:00 AM	4:00 AM	5:00 AM	6:00 AM	7:00 AM	8:00 AM	9:00 AM	10:00 AM	11:00 AM
12 noon	1:00 PM	2:00 PM	3:00 PM	4:00 PM	5:00 PM	6:00 PM	7:00 PM	8:00 PM	9:00 PM	10:00 PM	11:00 PM

Traffic levels for
Traffic levels for each hour:
Low (less than 250 vehicles per hour per lane)
Medium (between 250 and 2000 vehicles per hour per lane)
High (more than 2000 vehicles per hour per lane)

Form 4. Treatment information

Treatment						
Strategy	Min snow to trigger treatment (mm)	Recommended chemical	Recommended chemical form	Lowest recommended application rate	Highest recommended application rate	Treatment rate increment

Pretreatment				
Recommended Pretreatment chemical	Recommended pretreatment chemical form	Lowest recommended pretreatment rate	Highest recommended pretreatment rate	Pretreatment rate increment

Treatment Strategies:
Trigger on Hazardous Events
Trigger at every Route Time while storm is ongoing
Chemical: Plow only, NaCl, CaCl2, MgCl2, CMA, KAC, Caliber, Ice Slicer, and Ice Ban
Chemical form: Dry, Pre wet, and Liquid
Application rate: lbs/lane-mile for dry or pre wet, gal/lane-mile for liquid

Form 5. Routes in studied area

Route	Milepost	Name	Type	Longitude	Latitude	Elevation
Rt 46	0.21	Rt 94 at Rt 46 West	Overpass			
Rt 46	0.32	Rt 80 at Rt 46	Overpass			
Rt 46	0.74	Paulins Kill	Bridge			
Rt 46	3.74	Stream	Bridge			
Rt 46	4.28	Creek	Bridge			
Rt 46	7.29	Beaver Brook	Bridge			
Rt 46	9.09	Creek	Bridge			
Rt 80	0.74	Dunnfield Creek	Bridge			
Rt 80	0.87	Rt 80 Exit 2 Parking Lot	Overpass			
Rt 80	1.45	Stream	Bridge			
Rt 80	1.67	Stream	Bridge			
Rt 80	2.05	Rt 80 Exit 3 Rest Area	Overpass			
Rt 80	2.45	Trib. of Delaware River	Bridge			
Rt 80	2.92	Stream	Bridge			
Rt 80	4.58	Rt 94 At Rt 80	Overpass			
Rt 80	4.94	Paulins Kill	Bridge			
Rt 80	6.01	Rt 605 Delaware Rd	Overpass			
Rt 80	6.4	Delawana Creek	Bridge			
Rt 80	11.95	Beaver Brook	Bridge			
Rt 80	12.03	Hope-Blairstown Rd	Overpass			
Rt 80	12.97	Hope-Johnsonburg Rd	Overpass			
Rt 80	13.08	Trout Brook	Bridge			
Rt 80	15.87	Shades of Death Rd	Overpass			
Rt 80	16.13	Bear Creek	Bridge			
Rt 80	17.99	Pequest River	Bridge			
Rt 80	18.26	Abandoned RR	Overpass			

APPENDIX C –PROGRAMS DEVELOPED FOR NJ-MDSS

Weather Forecast data From DTN

```
import os, time
import logging.handlers
#from ftplib import FTP
import ftplib

def handleDownload(block):
file.write(block)

hostname    = 'wxftp.dtn.com'
username = 'njit'
password = 'kcG4j=6WIgX9FA'
base_dir = '/home/mdss/data/'
mesh_fcst_dir = 'mesh_fcst/'

# get current time
ts = 3600*int((time.time()/3600))
t = time.gmtime(ts)

l_name = time.strftime('%Y%m%d.%H00',t)
mesh_fcst_name = 'rdwx_fcst.' + l_name  + '.nc'
full_mesh_fcst_dir = base_dir + mesh_fcst_dir + time.strftime('%Y%m%d',t) +'/'
full_mesh_fcst_name = full_mesh_fcst_dir + mesh_fcst_name

# setup logging file
log = logging.getLogger("MDSS_FTP")
handler = logging.handlers.RotatingFileHandler(base_dir+'log/ftp_transfer.log', 'a',
1048576, 2)
formatter = logging.Formatter("%(asctime)s - %(levelname)8s : %(message)s")
handler.setFormatter(formatter)
log.addHandler(handler)
log.setLevel(logging.INFO)

if not os.path.isdir(full_mesh_fcst_dir):
os.mkdir(full_mesh_fcst_dir)

file = open(full_mesh_fcst_name, 'wb')

# start logging
log.info('FTP file transfer session started')

ftp = ftplib.FTP(hostname)   # connect to host, default port
```

```
ftp.login(username, password)

transfer_error = False
try:
ftp.retrbinary('RETR '+ mesh_fcst_name, handleDownload)
except ftplib.all_errors:
log.error('error retrieving forecast file: '+ mesh_fcst_name)
    transfer_error = True

ftp.quit()

if not transfer_error:
log.info('Forecast file ' + mesh_fcst_name + ' retrieved')
    f_ind = open(base_dir + mesh_fcst_dir + 'index.' + time.strftime('%Y%m%d',t), 'a')
    f_ind.write(mesh_fcst_name+ ' ' + str(int(time.time())) + '\n')
    f_ind.close()

log.info('FTP file transfer session ended')
```

Weather Observations Data From DTN

```
import os, time
import logging.handlers
import ftplib

def handleDownload(block):
file.write(block)

hostname    = 'wxftp.dtn.com'
username = 'njit'
password = 'kcG4j=6WIgX9FA'
base_dir = '/home/mdss/data/'
obs_hist_dir = 'metro_input/metro_input_obs_hist/'

# get current time
ts = 3600*int((time.time()/3600))
t = time.gmtime(ts)

l_name = time.strftime('%Y%m%d.%H15',t)
obs_hist_name = 'obs_hist.' + l_name  + '.nc'
full_obs_hist_dir = base_dir + obs_hist_dir + time.strftime('%Y%m%d',t) +'/'
full_obs_hist_name = full_obs_hist_dir + obs_hist_name

# setup logging file
log = logging.getLogger("MDSS_FTP")
```

```
handler = logging.handlers.RotatingFileHandler(base_dir+'log/ftp_transfer.log', 'a',
1048576, 2)
formatter = logging.Formatter("%(asctime)s - %(levelname)8s : %(message)s")
handler.setFormatter(formatter)
log.addHandler(handler)
log.setLevel(logging.INFO)

if not os.path.isdir(full_obs_hist_dir):
os.mkdir(full_obs_hist_dir)

file = open(full_obs_hist_name, 'wb')

# start logging
log.info('FTP file transfer session started')

ftp = ftplib.FTP(hostname)   # connect to host, default port
ftp.login(username, password)

transfer_error = False
try:
ftp.retrbinary('RETR '+ obs_hist_name, handleDownload)
except ftplib.all_errors:
log.error('error retrieving observation file: '+ obs_hist_name)
   transfer_error = True

if transfer_error:
   transfer_error = False
   l_name = time.strftime('%Y%m%d.%H14',t)
   obs_hist_name = 'obs_hist.' + l_name  + '.nc'
   full_obs_hist_dir = base_dir + obs_hist_dir + time.strftime('%Y%m%d',t) +'/'
   full_obs_hist_name = full_obs_hist_dir + obs_hist_name

try:
ftp.retrbinary('RETR '+ obs_hist_name, handleDownload)
except ftplib.all_errors:
log.error('error retrieving observation file: '+ obs_hist_name)
      transfer_error = True

ftp.quit()

if not transfer_error:
log.info('Observation file ' + obs_hist_name + ' retrieved')
   f_ind = open(base_dir + obs_hist_dir + 'index.' + time.strftime('%Y%m%d',t), 'a')
   f_ind.write(obs_hist_name+ ' ' + str(int(time.time())) + '\n')
   f_ind.close()
```

```
log.info('FTP file transfer session ended')
```

Weather Observation Data From*Clarus*

```
# Retrieve Clarus observation
#
# Written by Kung-E Cheng
#
# 2009/10/19

import sys, os, time
import logging.handlers

# setup local file structure
base_dir        = '/home/mdss/data/'
clarus_basedir  = 'clarus/'
bin_dir = 'conv_meso/'

# setup logging file
log = logging.getLogger("MDSS_HTTP")
handler = logging.handlers.RotatingFileHandler(base_dir+'log/http_transfer.log', 'a',
1048576, 2)
formatter = logging.Formatter("%(asctime)s - %(levelname)8s : %(message)s")
handler.setFormatter(formatter)
log.addHandler(handler)
log.setLevel(logging.INFO)

# start logging
log.info('Clarus file transfer session started')

# get current time
ts = 1200*int((time.time()/1200))
t = time.gmtime(ts)

# set local file structure
clarus_name = time.strftime('%Y%m%d_%H%M',t)
conv_meso_time = time.strftime('%Y%m%d.%H%M',t)
dir_time = time.strftime('%Y%m%d',t)
local_dir = base_dir + clarus_basedir + dir_time + '/'
local_filename = local_dir + clarus_name + '.csv'
nc_name = base_dir + clarus_basedir + clarus_name + '.nc'
meso_bin_dir = base_dir + bin_dir + dir_time
meso_bin_name = meso_bin_dir + '/conv_meso.' + conv_meso_time + '.bin'

# abort if target file already existed
if os.path.exists(meso_bin_name):
```

```
log.info('Bin files ' + meso_bin_name + ' already exists')

else:
# create data directory if necessary
if not os.path.exists(local_dir):
os.mkdir(local_dir)

# get Clarus file
cmd = '~/scripts/perl/get_clarus.pl ' + time.strftime('%Y%m%d %H%M ',t) +
local_filename
   rtn_code = os.system(cmd)

#  convert Clarus data from CSV format to MADIS NetCDF format
if (rtn_code == 0):
log.info('Clarus files ' + local_filename + ' retrieved')
cmd = '~/bin/csv2madis ~/data/static_data/site_list/clarus_sites.asc
~/data/static_data/config/clarus_template.asc ' +  local_filename + '
~/data/static_data/cdl/clarus.cdl ' + nc_name
      rtn_code = os.system(cmd)

if (rtn_code == 0):
log.info('Madis files ' + nc_name + ' converted')

# create target directory if necessary and convert NetCDF to bin format
if not os.path.isdir(meso_bin_dir):
os.mkdir(meso_bin_dir)
cmd = '~/bin/meso2bin ~/data/static_data/site_list/rwis_sites.asc
~/data/static_data/site_list/site_var_file.asc 1 ' +  nc_name + ' ' + meso_bin_name
         rtn_code = os.system(cmd)

if (rtn_code == 0):
log.info('Binary files ' + meso_bin_name + ' converted')
            f_ind = open(base_dir + bin_dir + '/index.' + time.strftime('%Y%m%d',t), 'a')
            f_ind.write('conv_meso.' + conv_meso_time + '.bin' + ' ' + str(int(time.time())) +
'\n')
            f_ind.close()
else:
log.error('Binary files ' + meso_bin_name + ' convertsion failed')

os.remove(nc_name)

else:
log.error('Madis files ' + nc_name + ' conversion failed')

#       os.remove(local_filename)
```

```
else:
log.error('Clarus files ' + local_filename + ' retrievial failed')

log.info('Clarus file transfer session ended')
```

Traffic Camera Images From NJ511

```
import sys, os, time, urllib, ftplib
import logging.handlers
import xml.dom.minidom

# setup local file structure
base_dir        = '/home/mdss/data/'
camera_basedir  = 'webcam/'
camera_config_file = base_dir + 'static_data/site_list/fixed_cameras.asc'
#ftp
def handleDownload(block):
    file_ftp.write(block)
hostname='dsftp.vaisala.com'
username='njmdss'
password='s7ecEGen'
camera_config_ftpfile=base_dir +'static_data/site_list/fixed_cameras_old.asc'
#ftp
# get current time
# get camera images every 10 minutes
ts = 600*int((time.time()/600))
t = time.gmtime(ts)
dir_time = time.strftime('%Y%m%d/',t)
file_time = time.strftime('%Y%m%d_%H%M',t)

#update site list

site_list=[]
cam_file=open(camera_config_file)
for cam_list in cam_file:
    site_list.append((cam_list.strip().split(";")))
cam_file.close()
cam_file=open(camera_config_file,"w")
for camera in site_list:
    camera_link=camera[4]

#get image new pubtoken

page=urllib.urlopen("http://embed.trafficland.com/nj511/trafficland_singlecam.php?webi
d="+camera[5]+"&key=2d0e656c67dde5ffc06286a3953ff934")
text=page.read().decode("utf8")
```

```
where=text.find('&pubtoken=')
    where1=text.find('&r="+Math.random()')
    start_end=where1
    start_read=where+10

#replace pubtoken

    key_index=camera_link.find('&pubtoken=')+10
    camera_link[:key_index]
new=text[start_read:start_end]

cam_file.write(camera[0]+';'+camera[1]+';'+camera[2]+';'+camera[3]+';'+camera_link.repl
ace(camera_link[key_index:],new)+';'+camera[5]+'\n')
cam_file.close()

# get site list
#site_list = []
#cam_f = open (camera_config_file,"r")
#for cam_site in cam_f:
   # if (cam_site.startswith('#') or (len(cam_site.strip()) == 0)):
       # continue
   # site_list.append((cam_site.strip().split(";")))
#cam_f.close()

# setup remote file structure

# setup logging file
log = logging.getLogger("Traffic_camera")
handler = logging.handlers.RotatingFileHandler(base_dir+'log/camera_transfer.log', 'a',
1048576, 2)
formatter = logging.Formatter("%(asctime)s - %(levelname)8s : %(message)s")
handler.setFormatter(formatter)
log.addHandler(handler)
log.setLevel(logging.INFO)

# start logging
log.info('Camera images file transfer session started')

# loop through camera list
for camera in site_list:

# get camera images
    remote_filename = camera[4]
    camera_dir = base_dir + camera_basedir + camera[0] + '/'
    img_name = camera[0] + '.' + file_time + '.jpg'
    local_filename = camera_dir + dir_time + img_name
```

```
if (os.path.exists(local_filename)):
         log.info('Camera image file ' + local_filename + ' already exists')

else:
if (not os.path.isdir(camera_dir)):
os.mkdir(camera_dir)

if (not os.path.isdir(camera_dir + dir_time)):
os.mkdir(camera_dir + dir_time)

       transfer_error = False
camera[4] = False

try:
urllib.urlretrieve(remote_filename, local_filename)
except urllib.IOError:
log.error('error retrieving camera image file: '+ local_filename + '.')
          transfer_error  = True
except urllib.ContentTooShortError:
log.error('error retrieving camera image file: '+ local_filename+ 'File truncated.')
          transfer_error  = True

if not transfer_error:
camera[4] = True
log.info('Camera files ' + local_filename + ' retrieved')

       f_ind = open(camera_dir + 'index.' + time.strftime('%Y%m%d',t), 'a')
       f_ind.write(img_name+ ' ' + str(int(time.time())) + '\n')
       f_ind.close()
#ftp site list
site_ftp_list=[]
cam_ftp_f=open(camera_config_ftpfile,"r")
for  cam_ftp_site in cam_ftp_f:
    site_ftp_list.append((cam_ftp_site.strip().split(";")))
cam_ftp_f.close()
  #ftp logging
ftp=ftplib.FTP(hostname)
ftp.login(username,password)
remote_img_dir='/fromvaisala'
ftp.cwd(remote_img_dir)
 #ftp camera images
for camera_ftp in site_ftp_list:
   camera_ftp_dir=base_dir+camera_basedir+camera_ftp[0]+'/'
   img_ftp_name=camera_ftp[0]+'.jpg'
   img_local_name=camera_ftp[0]+'.'+file_time+'.jpg'
```

```
    local_ftp_filename=camera_ftp_dir+dir_time+img_local_name
if (os.path.exists(local_ftp_filename)):
log.info('Camera image file '+local_ftp_filename+' already exists')
else:
if (not os.path.isdir(camera_ftp_dir)):
os.mkdir(camera_ftp_dir)
if (not os.path.isdir(camera_ftp_dir+dir_time)):
os.mkdir(camera_ftp_dir+dir_time)
        file_ftp=open(local_ftp_filename, 'wb')
        transfer_error=False
        camera_ftp[4]=False
try:
ftp.retrbinary('RETR ' + img_ftp_name, handleDownload)
except ftplib.all_errors:
log.error('error retrieving camera image file: '+img_ftp_name+'.')
            transfer_error=True
print("error+++++++++++")
if not transfer_error:
            camera_ftp[4]=True
log.info('Camera files '+img_ftp_name+'retrieved')
        f_ftp_ind=open(camera_ftp_dir+'index.'+time.strftime('%Y%m%d',t),'a')
        f_ftp_ind.write(img_local_name+' '+str(int(time.time()))+'\n')
        f_ftp_ind.close()
ftp.quit()
#ftp finished

camera_meta_dir = base_dir + camera_basedir + 'meta/' + dir_time
meta_filename =  'meta.' + file_time + '.xml'
camera_meta_file = camera_meta_dir + meta_filename

doc = xml.dom.minidom.Document()
traffic_camera = doc.createElement("CAMERA_METADATA")
doc.appendChild(traffic_camera)
camera_type = doc.createElement("GPSData")
traffic_camera.appendChild(camera_type)
camera_type = doc.createElement("FixedLocationCameras")
traffic_camera.appendChild(camera_type)
#ftp update
new_list=site_list+site_ftp_list
print(new_list)
#for camera in site_list:
for camera1 in new_list:
#ftp update finished
   # print(camera1)
if camera1[4]:
        camera_site = doc.createElement("NJDOTCam_last_image")
```

```
	camera_site.setAttribute("serial_number", camera1[0])
	camera_site.setAttribute("entry_UTC_time_stamp", time.strftime('%Y-%m-
%dT%H:%M:%S.000',t))
	camera_loc = doc.createElement("NJDOTCam_last_location")
	camera_loc.setAttribute("latitude", camera1[2])
	camera_loc.setAttribute("longitude", camera1[3])
	camera_site.appendChild(camera_loc)
	camera_type.appendChild(camera_site)

if not os.path.isdir(camera_meta_dir):
os.mkdir(camera_meta_dir)

f = open(camera_meta_file, 'w')
f.write(doc.toprettyxml(indent="  "))
f.close()

doc.unlink()

f_ind = open(base_dir + camera_basedir + 'meta/' + 'index.' +
time.strftime('%Y%m%d',t), 'a')
f_ind.write(meta_filename + ' ' + str(int(time.time())) + '\n')
f_ind.close()
#print(f)
print("finish")
log.info('Camera images transfer session ended')
```

AVL Data From NJDOT

```
import sys, os, time
import xml.dom.minidom
import MySQLdb

db_user = 'trucklog'
db_name = 'trucklog'
db_passwd = 'w3TctTVRdqKDdm6v'

db = MySQLdb.connect(host="localhost", user=db_user, passwd=db_passwd,
db=db_name)
cursor = db.cursor()

# setup local file structure
base_dir	= '/home/mdss/data/'
avl_basedir  = 'avl/'

# get time
ts = int(time.time() / 60) * 60
```

```
t = time.gmtime(ts)
ot = time.gmtime(ts - 432000)

avl_dir = base_dir + avl_basedir + time.strftime('%Y%m%d',t) + '/'
avl_filename = 'avl.' + time.strftime('%Y%m%d.%H%M%S',t) + '.xml'
avl_file = avl_dir + avl_filename

cursor.execute("""SELECT t1.* FROM logs AS t1 INNER JOIN ( SELECT truck_id,
MAX( obs_time ) AS MaxDateTime FROM logs WHERE obs_time < '%s' AND obs_time
> '%s' GROUP BY truck_id ) AS t2 ON t1.truck_id = t2.truck_id AND t1.obs_time =
t2.MaxDateTime""" % (time.strftime('%Y-%m-%d %H:%M:00', t), (time.strftime('%Y-%m-
%d %H:%M:00', ot))))

if (cursor.rowcount > 0):
res = cursor.fetchall()
doc = xml.dom.minidom.Document()
avl = doc.createElement("GPSdata")
doc.appendChild(avl)

for row in res:
lat = str(row[2])
lon = str(row[3])
	truck_id = row[1]
	obs_time = row[0]
dir = str(row[4])
speed = str(row[5])

	last_loc = doc.createElement("NJDOT.dbo.gps_last_location")
avl.appendChild(last_loc)
	last_loc.setAttribute("UTC_date_time_of_fix", obs_time.strftime('%Y-%m-
%dT%H:%M:%S'))
	last_loc.setAttribute("latitude", lat)
	last_loc.setAttribute("lat_hemisphere", 'N')
	last_loc.setAttribute("longitude", lon)
	last_loc.setAttribute("long_hemisphere", 'W')
	last_loc.setAttribute("speed_over_ground", speed)
	last_loc.setAttribute("course_over_ground", dir)
	last_loc.setAttribute("GPS_unit_status", 'A')
	last_loc.setAttribute("serial_number", truck_id)
	last_report = doc.createElement("NJDOT.dbo.mdss_last_reported")
	last_loc.appendChild(last_report)
	last_report.setAttribute("serial_number", truck_id)
	last_report.setAttribute("status", 'Truck started')
	last_report.setAttribute("status_time_stamp", obs_time.strftime('%Y-%m-
%dT%H:%M:%S'))
```

```
if not os.path.isdir(avl_dir):
os.mkdir(avl_dir)

    f = open(avl_file, 'w')
f.write(doc.toxml())
#    f.write(doc.toxml(encoding="UTF-8"))
f.close()

doc.unlink()

    f_ind = open(base_dir + avl_basedir + 'index.' + time.strftime('%Y%m%d',t), 'a')
    f_ind.write(avl_filename + ' ' + str(int(ts)) + '\n')
    f_ind.close()
```

APPENDIX D - MDSS HARDWARE AND SOFTWARE REQUIREMENTS

MDSS consists of input processes, algorithm processes, and a display process. This section discusses the hardware and software architecture and the communications and network interfaces of the prototype MDSS.

Computer Hardware: The computational requirements of the MDSS are linearly related to the number of forecast sites. For all MDSS site list configurations seen thus far (~500 forecast sites), one machine is sufficient to run the entire data input and algorithms. The MDSS is scalable and if the number of forecast sites is sufficiently large, either more machines with the same configuration will be required, or a single machine with faster processors and more memory will be required. The display can also be run on this same single platform, but it is forecasted that users at remote sites would most likely be running a display locally. The system described here assumes that the data input and algorithms run on a single machine and the output is accessed by the end-users via the Internet. Since processes may be user-request driven, the system load will be slightly more unscheduled and unpredictable than described above. These user requests will generally be rare and require a relatively small amount of processing. However, if the user base is large enough and many requests are generated simultaneously, the computational resources of the system may be maxed. In this case, it may be necessary to add further computational resources. The MDSS hardware requirements for FY 2003 are displayed below. Requirements for running only the display PC component of the system are less than what is required for running the RWFS and RCTM components. Minimal PC requirements are shown. Additional memory and a faster processor should provide better display performance.

RCTM components

- Dual 3.6 GHz processors
- 300 GB disk space
- 2.0 GB memory

Display PC

- 1 GHz processor
- 500 MB disk space
- 512 MB memory

Communications and Network Interfaces: The MDSS uses standard TCP/IP protocols. This is the underlying protocol standard for most internet communication. An Internet connection with at least T1 data rates is required. This allows timely download of large numerical weather prediction data sets, but it is not required for the display component of the system. If communications redundancy is desired (See Redundancy Section of this document), a NOAA Port satellite downlink system may be purchased. In this case, the dish vendor should be consulted for interface compatibility. If more than one machine is required to run the MDSS system, the database should be cross-mounted

(NFS) on a Local Area Network (LAN). In this way, all data will appear to be local on any machine. I/O is generally not a bottleneck for the internal processing of the data.

Software Architecture: The MDSS data input and algorithms have been designed to run on common UNIX workstations. Currently these subsystems have only been compiled and tested thoroughly under the Intel-based GNU/Debian Linux operating system. The display has been developed as a Java application. As such, the binary-generated Java byte code is machine independent and should run on any properly configured hardware system. The MDSS is a distributed system. It consists of a number of independent processes each performing a straightforward, well-defined function. The individual modules of the MDSS have been designed to be relatively simple. Each process, while perhaps doing sophisticated processing, has been designed to know very little about the outside world. For example, the processes know nothing about the file system. Nor do they consult the system clock to find out the current time. Instead, all information required for processing, such as file names and relevant time parameters, is passed to the process as command line arguments. The rationale behind this design is to create a system that is both reliable and easy to maintain. One major advantage of this methodology is that every instance of every process is completely repeatable. Log files, described later, keep track of the command line executed, as well as the status of each process run. Bugs can easily be traced by repeatedly running the code with the same command line.

Computer Languages: Each component of the system is coded in one of the following languages: C/C++; Java; FORTRAN; Perl and Python. Most of the MDSS algorithm code is written in C++. The exceptions for compiled code are the road temperature and snow depth model, (METRo - Model of the Environment and Temperature of Roads), which are written in FORTRAN. The scripting languages, Perl and Python, are used for data reformatting, server-side request processing (CGI handling of display requests), and process invocation scripts. To the greatest extent possible, C and C++ code is POSIX compliant.

External Software Requirements: A number of third-party software packages are required to run the entire MDSS system. These packages are freely available and must be installed before installing the MDSS system. The table below lists these packages, the minimum version required, and where they can be obtained:

- libxml2 2.6.x ftp://xmlsoft.org/libxml2/
- libxslt 1.1.x ftp://xmlsoft.org/libxml2/
- Python 2.4 http://www.python.org/download/
- Numarray 1.5.x http://downloads.sourceforge.net/numpy/
- libxml2 binding 2.6+ ftp://xmlsoft.org/libxml2/python/
- SWIG 1.3.x http://downloads.sourceforge.net/swig
- Perl 5.8+ http://www.perl.org/get.html
- Java JDK 1.6.+ http://java.sun.com/javase/downloads/
- Apache Webserver 2.x http://httdp.apache.org/
- suExec http://httpd.apache.org/docs/2.0/suexec.html

▪ Apache Ant	1.7+		http://ant.apache.org/
▪ PHP	4.x		http://www.php.net/
▪ php4-common			
▪ libzzip-0-12			
▪ apache2-mpm-prefork			
▪ GNU gcc/g++/	g77	3.3+	http://www.gnu.org/software/
▪ make/wget			
▪ netCDF	3.6+		http://www.unidata.ucar.edu/software/netcdf
▪ Unidata LDM	5.1.4+		http://www.unidata.ucar.edu/software/ldm/

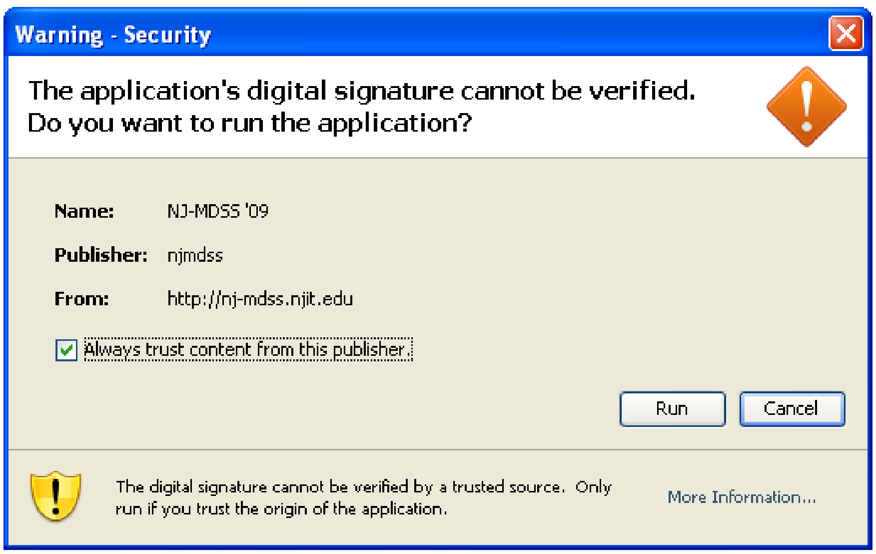
Inter-Process Communication: MDSS inter-process communication is done through the file system. Each process obtains its inputs solely through filenames and UNIX times specified on its command line. Many of these files specified are data files generated by upstream processes; many are static binary data files. The remainder is configuration files that contain the list of sites and forecast variables to be processed. Each process reads its input data from the files on its command line, processes it, and writes its output to a file. The output file name is specified on the command line. The output file name's format is specified in a file named on the command line. The interface between the display and upstream processes is a client-server relationship. The display is the client. Its requests are handled by a web server and processed using CGI scripts. The CGI scripts parse the requests and obtain the requested data. These data are then returned to the display.

APPENDIX E –USER'S MANUAL

Instruction of User Interface

1. NJ-MDSS Activation Page

To operate the MDSS software, Java Runtime Environment (JRE) Version 1.6 or later must be installed in the computer. NJ-MDSS can be launched with a Web browser from the NJ-MDSS starting page: http://nj-mdss.njit.edu/rdwx_mdss/xdss/xdss.jnlp.NJ-MDSS display client is a signed Java application. User needs to accept the security certificate to run NJ-MDSS.

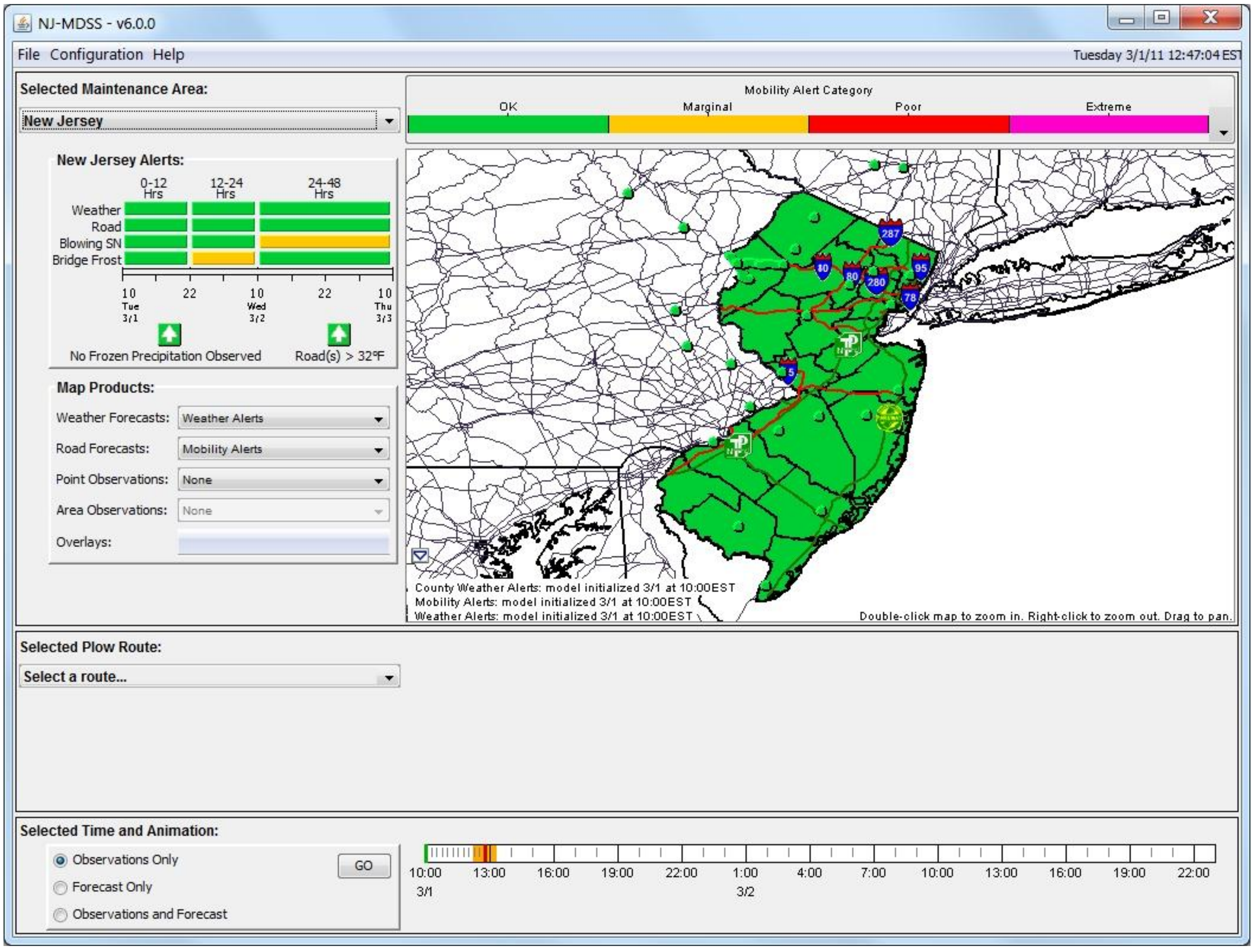


2. NJ-MDSS User Login Page

To log in NJ-MDSS system, user name and password are required.

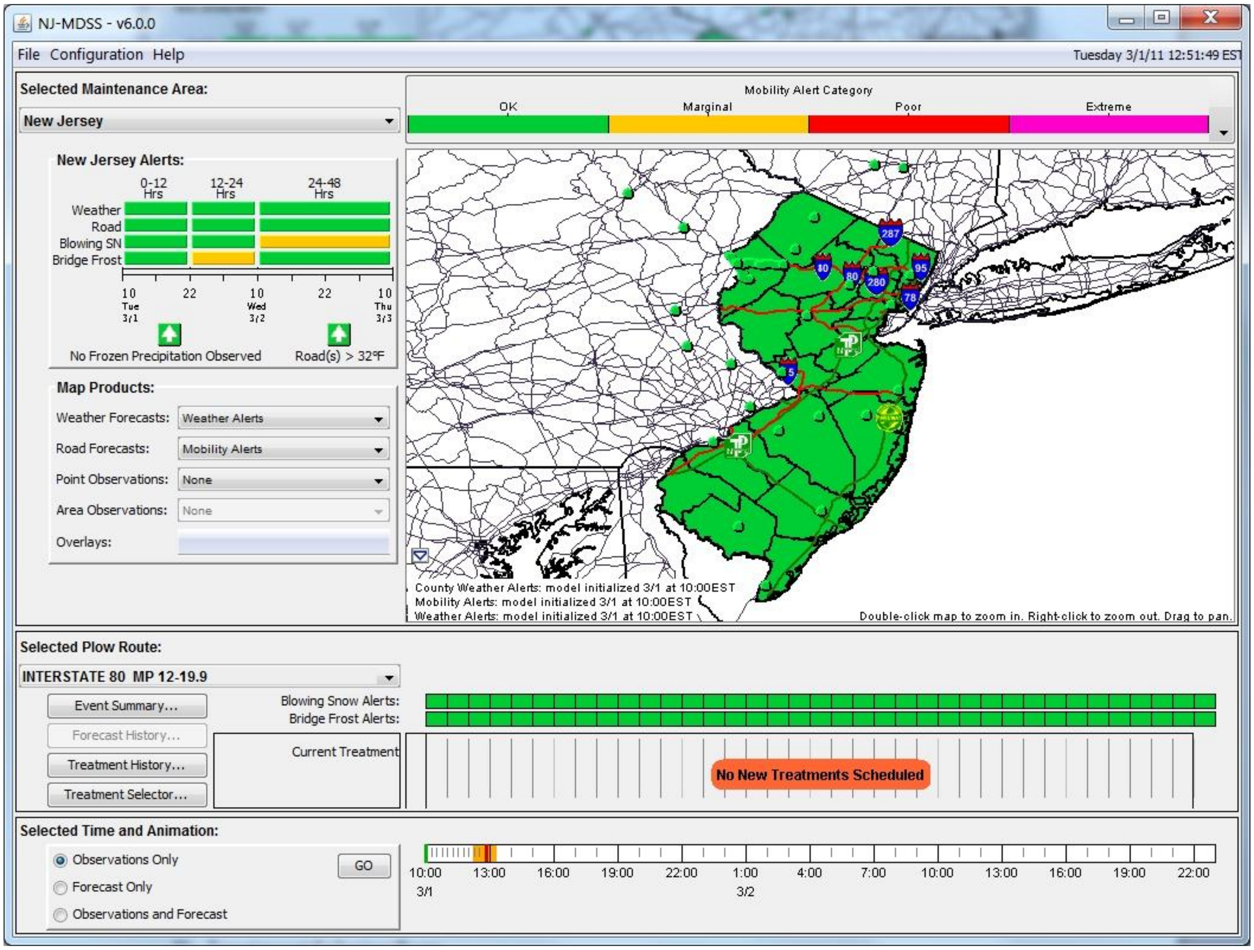
3. NJ-MDSS State View Page

It provides a summary of weather and road condition information across a broad region. Users can animate through the forecast period to assess weather and road condition threats. Weather and road condition alerts are also provided when weather and road conditions threaten the region within the next 48 hrs.



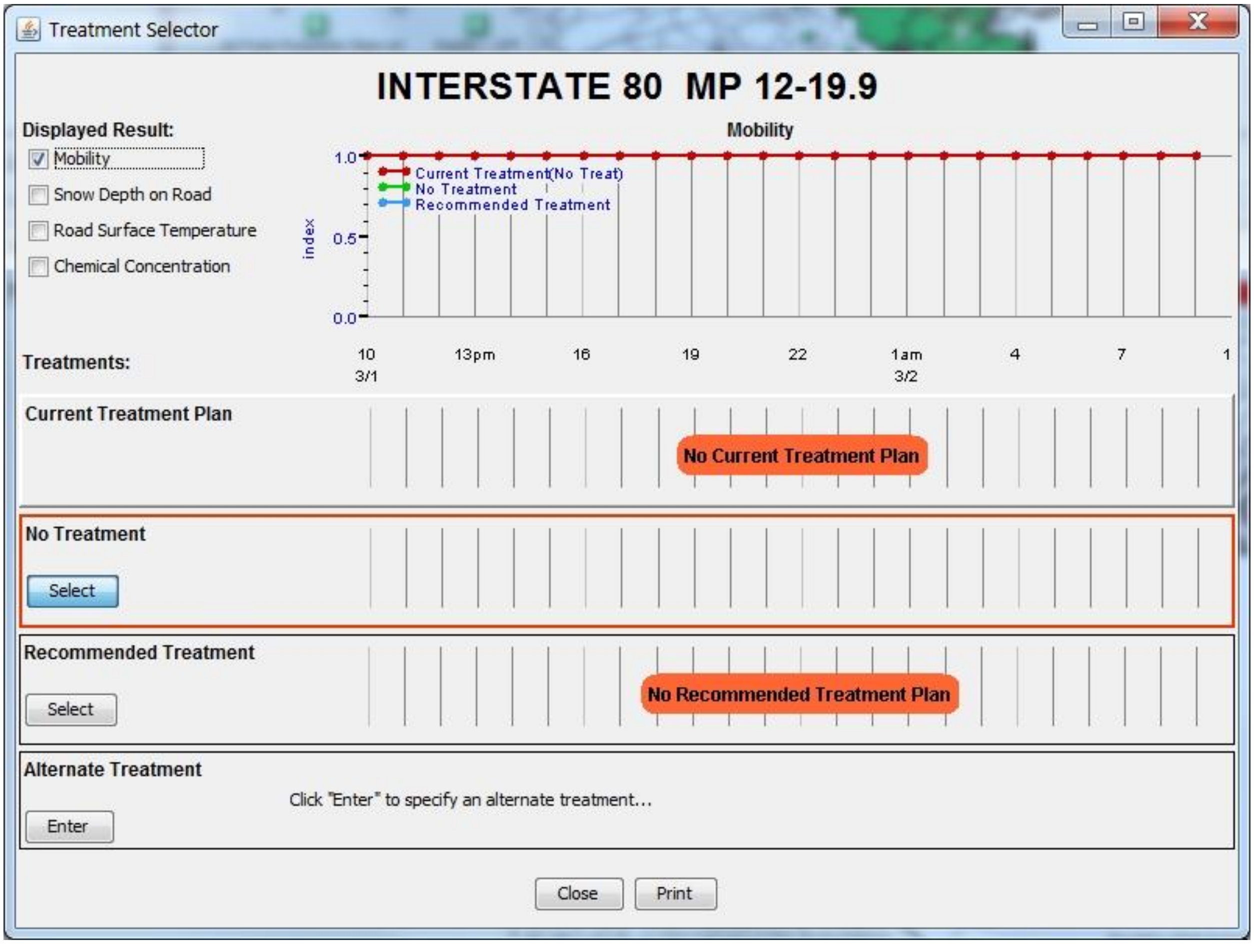
4. NJ-MDSS Route View Page

It provides a summary of weather and road condition information across a local region. Users can animate through the forecast period to assess weather and road condition threats. Weather and road condition alerts specific to the region are also provided when weather and road conditions threaten the region within the next 48 hrs. Users can also view treatment recommendations from this page and perform "what if" scenarios.



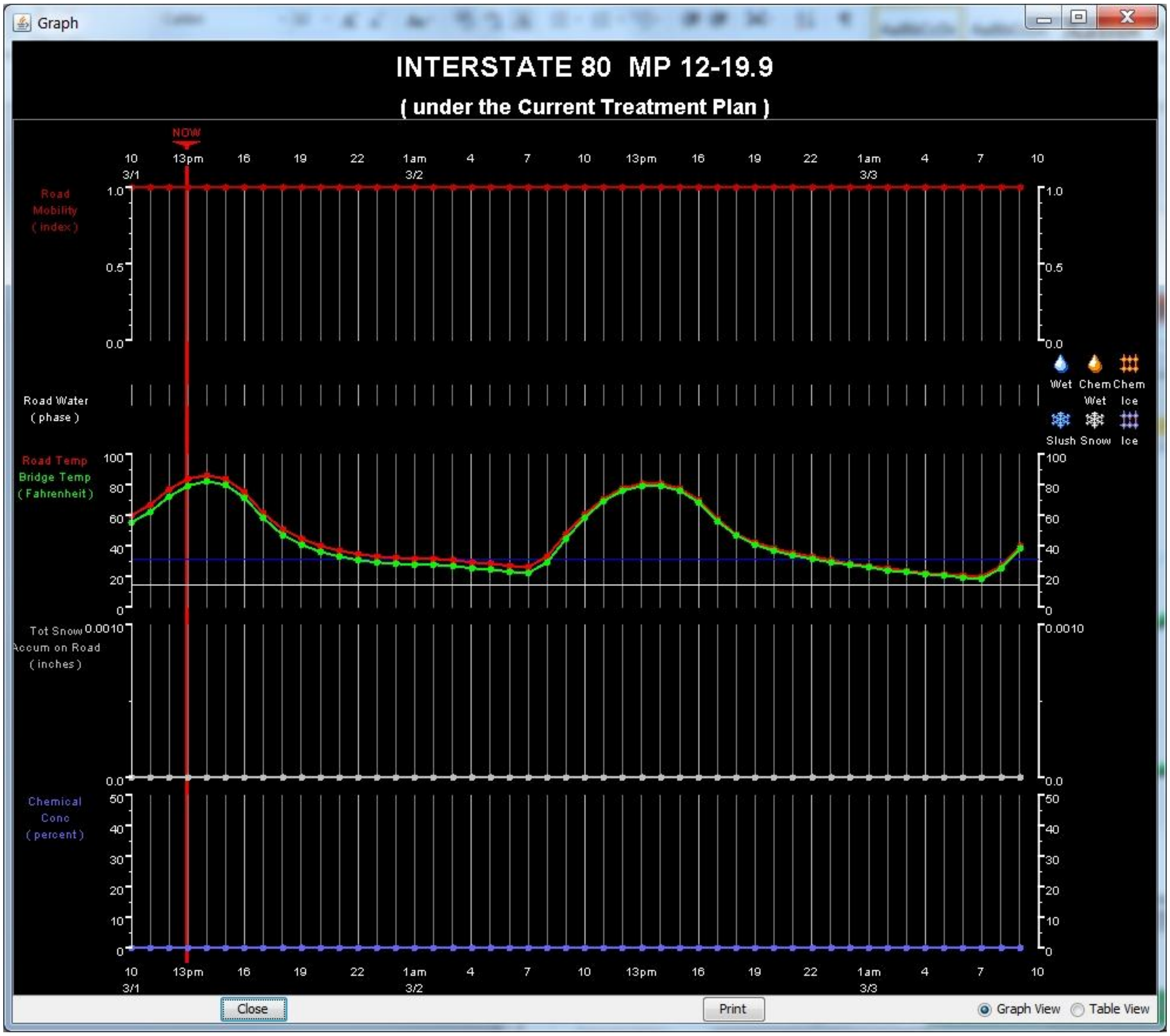
5. Treatment Selector Page

It allows users to view specific road condition prediction information for selected routes. Users can view recommended treatment plans, perform “what if” scenarios, and view alternate treatment plans.



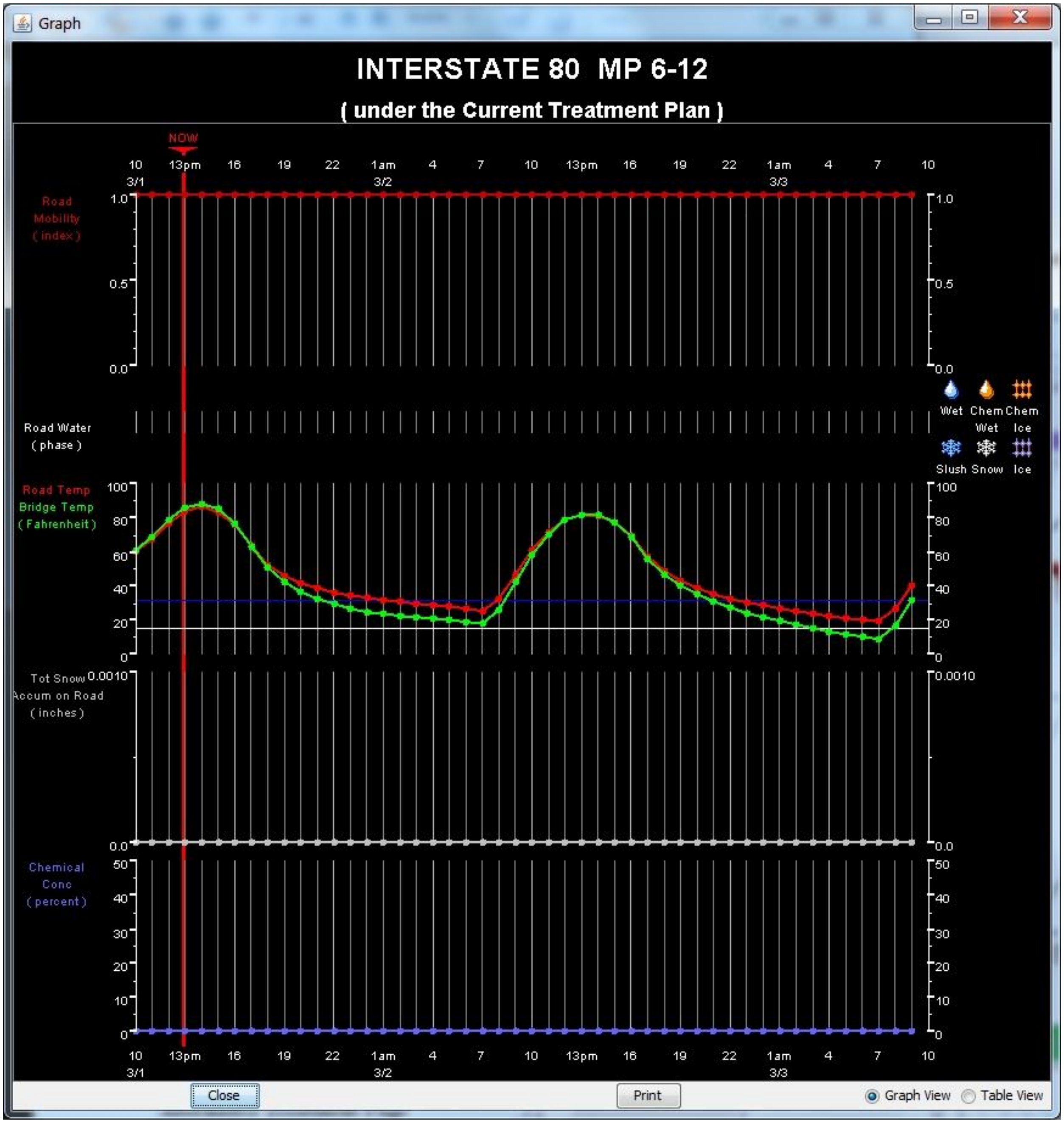
6. Weather Time Series

Users can view time series data of critical weather parameters over the 48-hr period for each route. Parameters include air temperature, wind speed and direction, snowfall rate, relative humidity, precipitation type, total snow accumulation, and probability of precipitation.



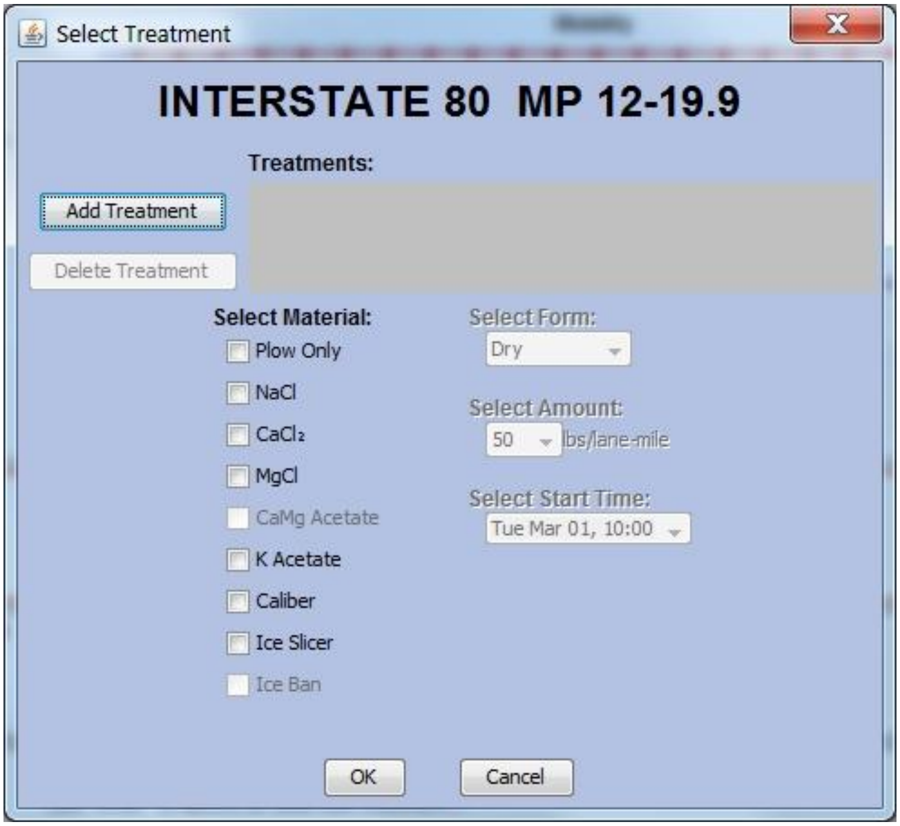
7. Road Condition Time Series

Users can view time series data of critical road condition parameters over a48-hr period. Parameters include road temperature, mobility, total snow accumulation on the road, and estimated chemical concentration based on selected treatment plans.



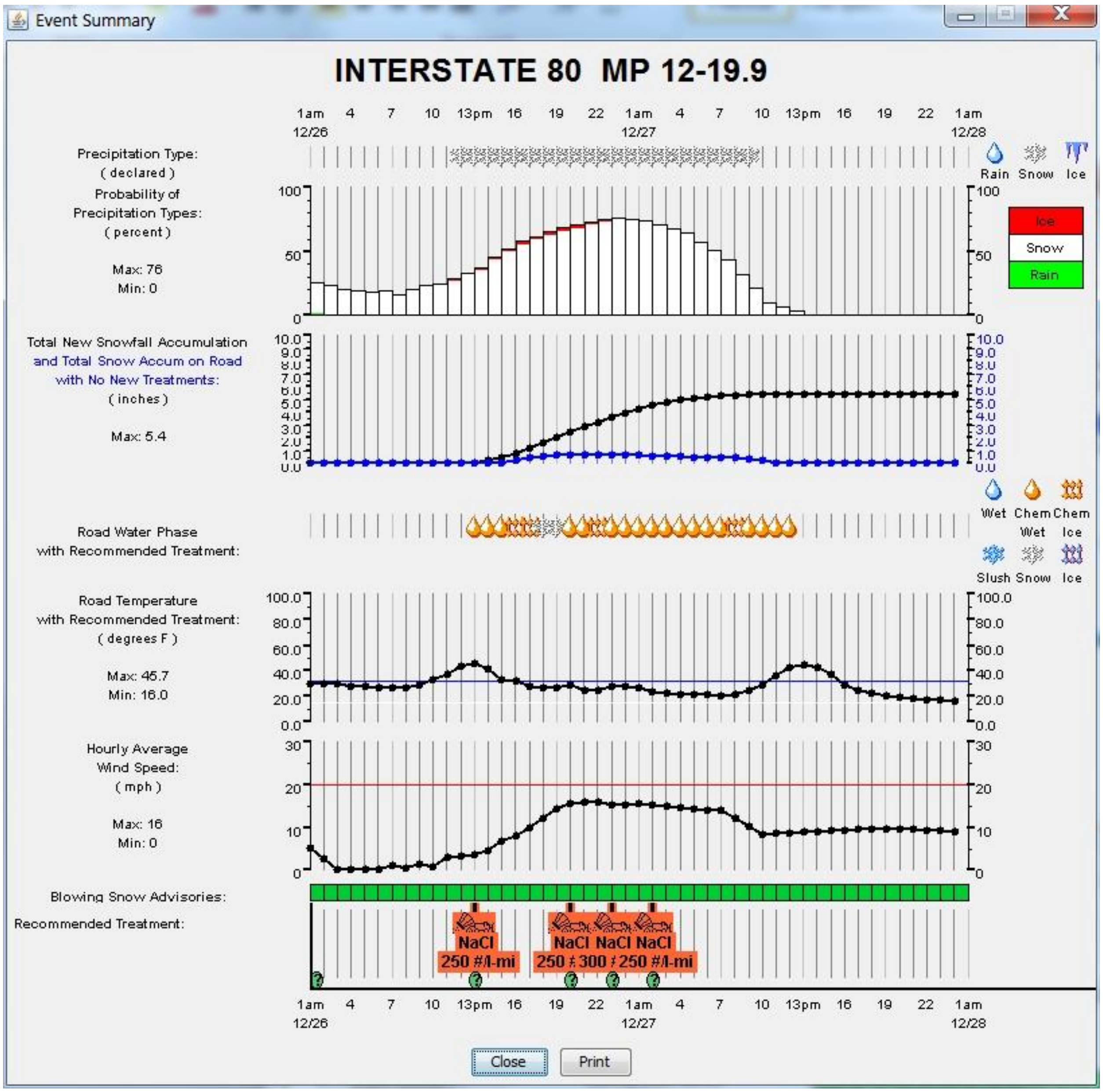
8. Alternative Treatment Page

This page allows users to input alternative treatments from the recommended treatments and view resulting road temperatures, snow accumulation, and chemical concentration. Users can select chemical types, application times, and amounts.



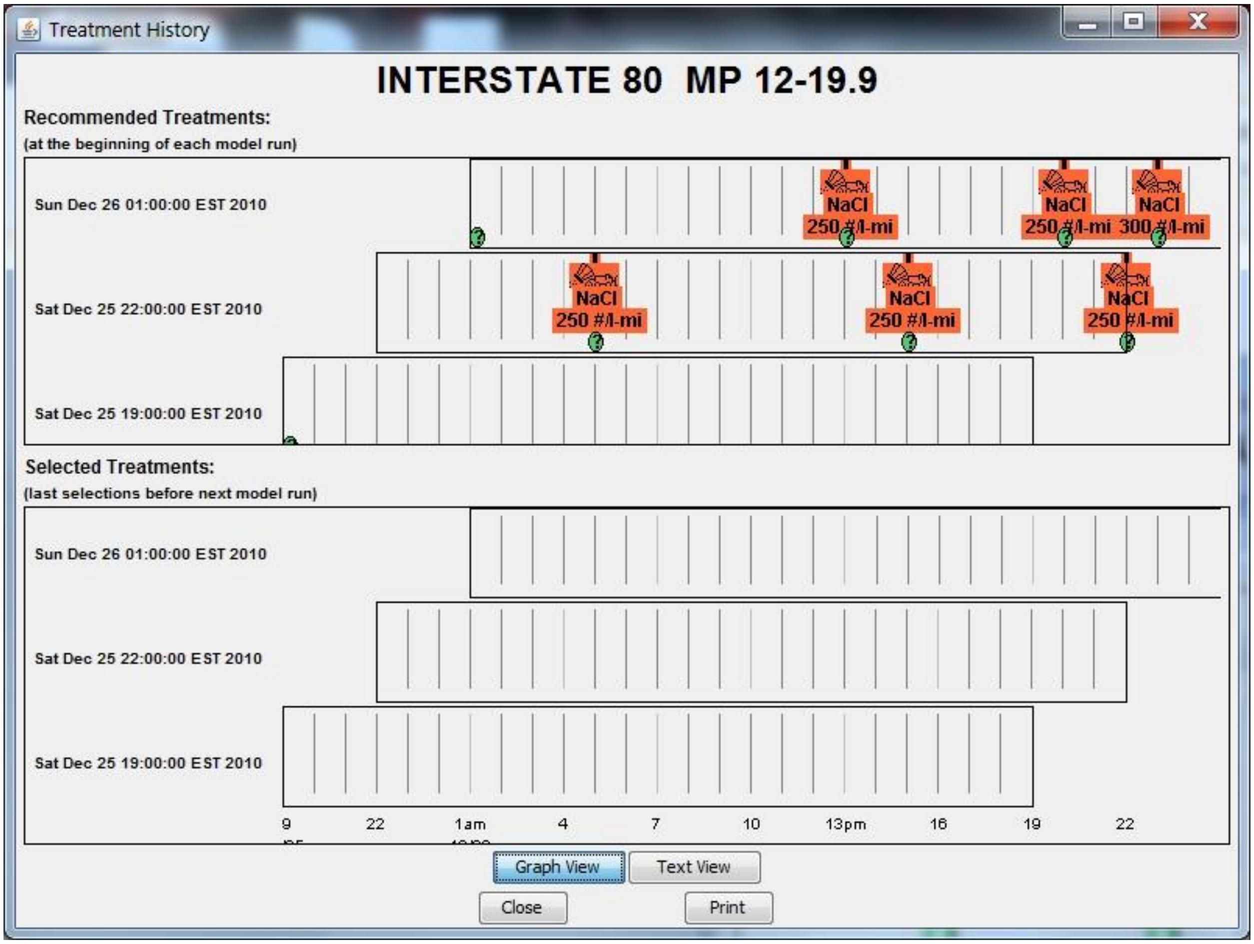
9. Storm Summary Page

This page shows storm event summary data for each plow route in graphical form. Products include maximum and minimum predicted road temperatures, wind speed, and snow accumulation on the road (with no treatment). The probability of precipitation is also shown. It is divided into precipitation type categories to give the user an indication of how the precipitation will change over the course of the storm event. Periods when blowing snow are likely, are also provided.



10. Forecast History Page

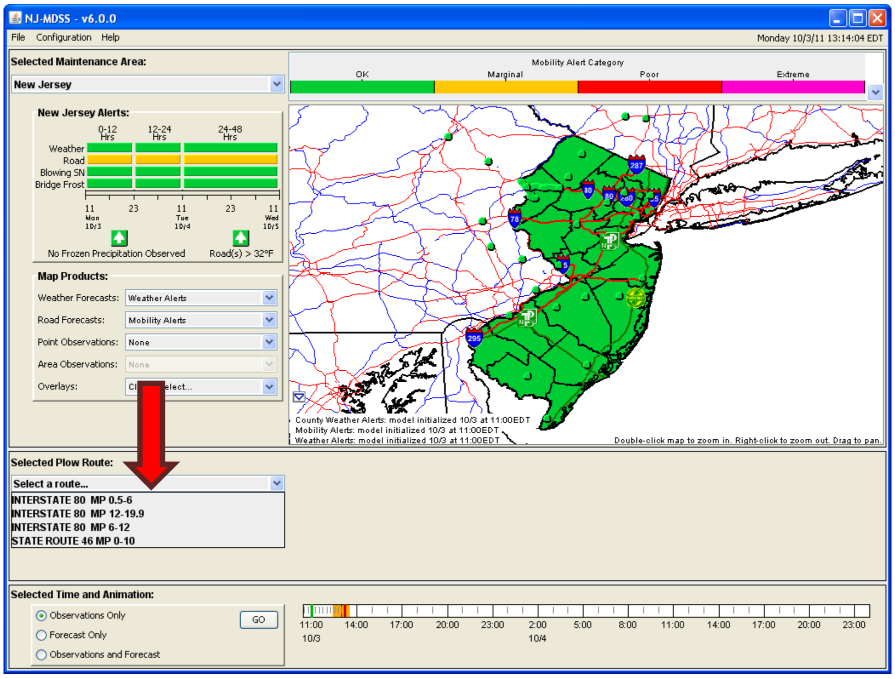
This page allows users to view previous weather and road condition predictions for selected forecast periods, and compare them to actual observations. The prototype is configured to allow users to compare air temperature, road temperature, relative humidity, and wind speed for user selected maintenance routes.



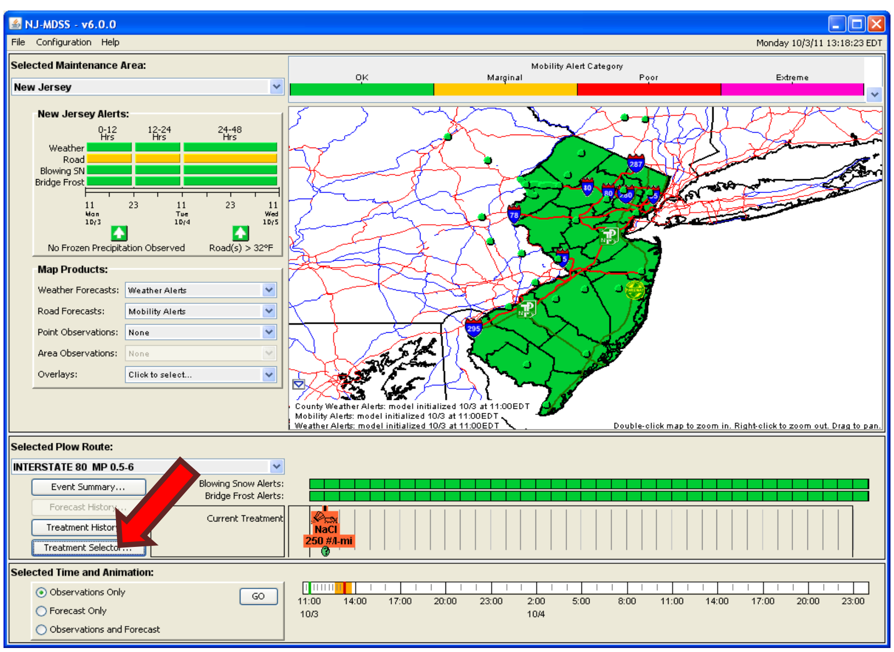
Introduction of Treatment Inputs

NJ-MDSS needs to input the road treatment information for predicting the surface condition. A step procedure for inputting a treatment is given below:

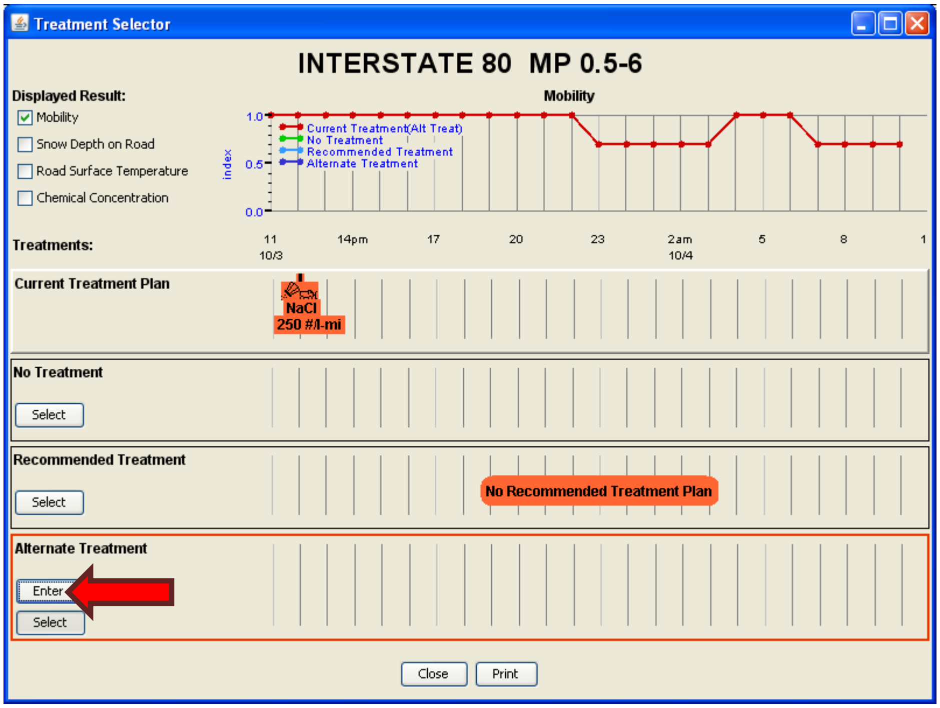
1. Select the route to be treated



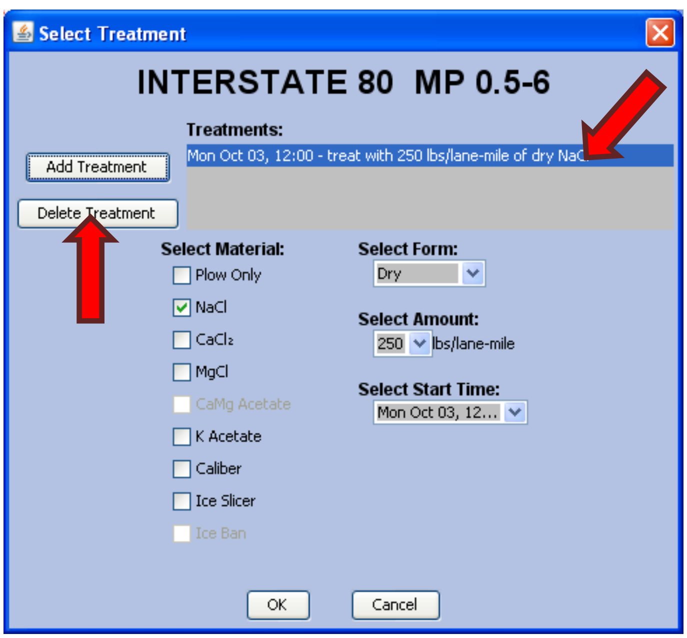
2. Check current treatment, if needed, click “Treatment Selector” to enter an alternative treatment



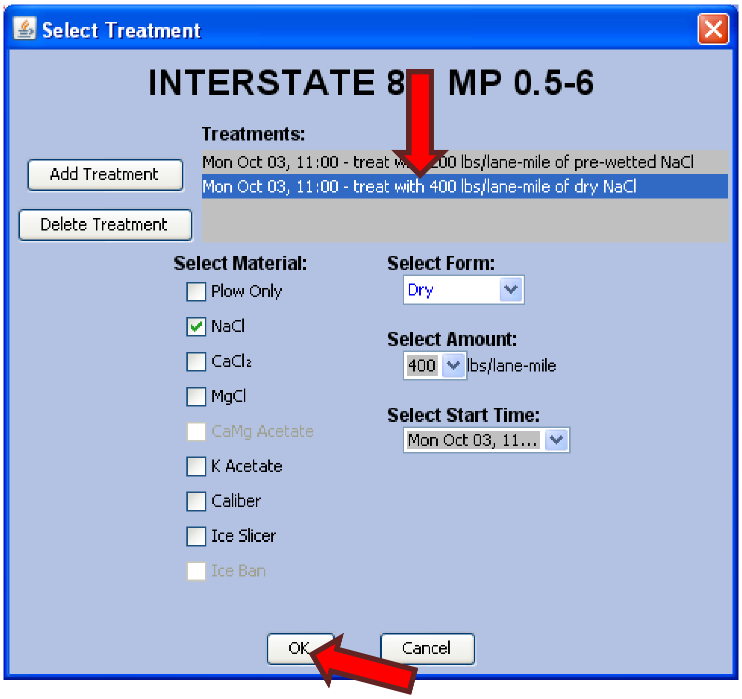
3. Click on the "Enter" under "Alternative Treatment" to input treatment



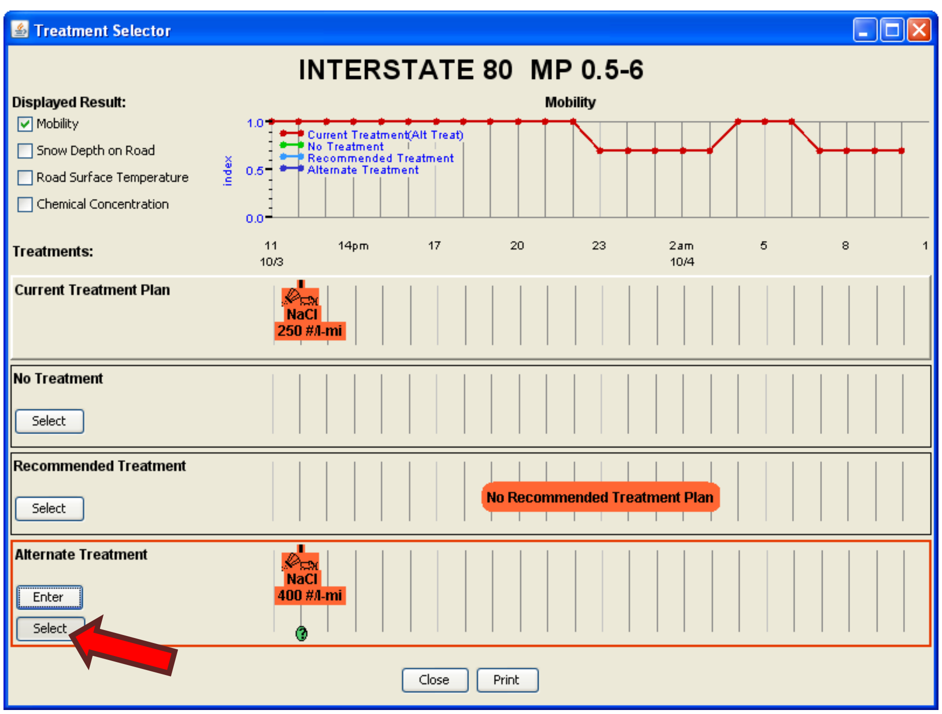
4. Click "Delete Treatment" to remove current treatment.



5. Click “Add Treatment”, then (1) Select Material; (2) Select Form; (3) Select Amount; (4) Select Start Time. If there is more than one treatment for a different time period, click “Add Treatment” again, then repeat selection from (1) to (4). Click “OK” to close “select treatment” window.

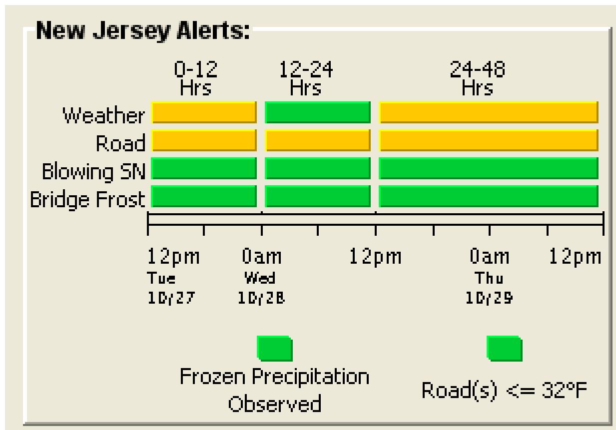


6. Click the “Select” in “Alternative Treatment” to apply treatment plan



Alerts in NJ-MDSS

The alert panel gives a summary of weather and road conditions for the next 48 hours for area showing in the map. The alerts are divided into short (0-12 hour), mid (12-24 hour), and long (24-48 hour) periods for maintenance managers making treatment planning. The color of the alert reflects the worst condition during the time period in the area. The alert is color coded with green showing normal condition, yellow marginal condition, red poor condition, and purple extreme condition. Tables A to C describe the alert categories for each alert type.



NJ-MDSS alerts panel

Table A Weather alert category

Weather Alert Category	Reason	Precip Type	Precip Rate	Temperature
Extreme	Ice or Freezing Rain	Ice		
Extreme	Heavy Snow	Snow	>= 1.5"/hr	
Poor	Moderate Snow	Snow	>= 0.5"/hr	
Poor	Cold Rain	Rain	> 0.25"/hr	< 35° F
Marginal	Light Snow	Snow		
Marginal	Light Rain	Rain	>= 0.1"/hr	
OK	No Other Conditions Met			

Table B Road alerts category

Road Alert Category	Mobility Index	Pavement Condition
OK	1.0	Dry
Marginal	0.7	Wet
Marginal	0.6	Snow < 4"
Poor	0.4	Snow 4"-6"
Poor	0.3	Snow > 6"
Extreme	0.2	Ice

Table C Blowing snow alerts and bridge frost alerts category

Alert Category	Likelihood
OK	None
Marginal	Low
Poor	Medium
Extreme	High

Color-coded weather alert for county-based zones can be displayed on the map. For a county without MOS weather stations, the weather forecast from a nearby weather station is applied. The weather stations applied to each alert zone are summarized in Table D. This alert reflects the worst case forecast of the weather station(s) in the next 48 hours.

Table D Alert zones

Alert Zone	Weather Station	Location
Bergen	Teterboro Airport	Bergen
Essex	Newark International Airport	Essex
	Caldwell/Fairfield	Essex
Hudson*	Teterboro Airport	Bergen
Morris	Morristown Airport	Morris
Passaic*	Caldwell/Fairfld Airport	Essex
Sussex	Aero/Andover Airport	Sussex
	Sussex Airport	Sussex
Union*	Newark International Airport	Essex
Warren	Allamuchy Township RWIS	Warren
	Knowlton Township RWIS	Warren
Hunterdon*	Somerville, Somerset Airport	Somerset
Mercer	Trenton/Mercer County Airport	Mercer
Middlesex*	Belmar-Farmdale	Monmouth
Monmouth	Belmar-Farmdale	Monmouth
Ocean	Lakehurst NAS	Ocean
Somerset	Somerville, Somerset Airport	Somerset
Atlantic	Atlantic City International Airport	Atlantic
Burlington	Mount Holly, South Jersey Regional Airport	Burlington
	McGuire AFB	Burlington
Camden*	Mount Holly, South Jersey Regional Airport	Burlington
Cape May	Wildwood	Cape May
Cumberland	Millville Municipal Airport	Cumberland
Gloucester*	Philadelphia International Airport	Philadelphia, PA
Salem*	Millville Municipal Airport	Cumberland

* Use a forecast station outside of county boundary

APPENDIX F -WEATHER DATA

The operation of MDSS depends on high quality weather forecasts and timely observations. Each region might have its unique weather data requirement, thus FHWA MDSS prototype was designed without a default weather data source. NJ-MDSS uses weather data provided by DTN/Telvent. The weather data provided by DTN/Telvent is in MDSS compliant format. They are retrieved by NJIT and placed into MDSS database.

Weather Forecast Data

Data format in the weather forecast data file:

Name	Length	Description
max_site_num		number of locations
days	4	number of days
fc_times_per_day	24	forecast times per day

Data fields in the weather forecast data file:

Name	Description	Unit/Comment
creation_time	time at which forecast file was created	seconds (since 1970-1-1 00:00:00)
forc_time	time of earliest forecast	seconds (since 1970-1-1 00:00:00)
num_sites	number of actual sites	
site_list	forecast site id numbers	see Appendix D for assigned ID number and location for each forecast site
T	Temperature	degrees Celsius
dewpt	Dewpoint	degrees Celsius
	cloud cover	percent
	conditional probability of rain	percent
	conditional probability of snow	percent
	conditional probability of ice	percent
	Windspeed	meters per second
	windspeed in mph	miles per hour
	wind direction clockwise from north	degrees north
	relative humidity	decimal
	percent relative humidity	percent
	precipitation (SWE) rate	mm/hr
	precipitation (SWE) rate	in/hr
	precipitation type	0=NONE, 1=RAIN, 2=SNOW, 5=ICE
	3 hr precipitation	inches

	accumulation	
	Temperature	degrees Fahrenheit
	Dewpoint	degrees Fahrenheit
	snowfall rate	in/hr
	3 hr snowfall accumulation	inches
	snowfall accumulation since start of forecast	mm
	snowfall accumulation since start of forecast	inches
	48hr snowfall accumulation since start of forecast	inches
	probability of precipitation, 3 hr	percent (0-100)
	blowing snow potential	index (0-3) (low-high)
	blowing snow potential values	floating point (0-1) (low-high)
	Pressure at 2m above surface	millibars
	downward long wave radiation flux at surface	W/m2
	downward short wave radiation flux at surface	W/m2

MDSS requires hourly weather forecast for the next 48 hours in netCDF format, while the specification of the forecast file calls for 96 hours (4 days) of forecast data. That is to ensure continuous system operation for another 48 hours in case there is an interruption in the weather forecast data feed. The weather forecast netCDF definition is depicted below:

```
netcdf mesh_derive {

dimensions:
        max_site_num = UNLIMITED;     // number of locations
        days = 4;                     // number of days
        fc_times_per_day = 24;        // number of forecasts per day
        daily_time = 1;               // same, for once-a-day variables

variables:
        double          creation_time;
                creation_time:long_name = "time at which forecast file was created";
                creation_time:units = "seconds since 1970-1-1 00:00:00" ;

        double          forc_time;
                forc_time:long_name = "time of earliest forecast";
                forc_time:units = "seconds since 1970-1-1 00:00:00" ;
```

```
int     num_sites;
        num_sites:long_name = "number of actual_sites";

int     site_list(max_site_num);
        site_list:long_name = "forecast site list";
        site_list:_FillValue = -99999;

float   T(max_site_num, days, fc_times_per_day);
        T:long_name = "temperature";
        T:units = "degrees Celsius";

float   max_T(max_site_num, days, daily_time);
        max_T:long_name = "maximum temperature";
        max_T:units = "degrees Celsius";

float   min_T(max_site_num, days, daily_time);
        min_T:long_name = "minimum temperature";
        min_T:units = "degrees Celsius";

float   dewpt(max_site_num, days, fc_times_per_day);
        dewpt:long_name = "dewpoint";
        dewpt:units = "degrees Celsius";

float   cloud_cov(max_site_num, days, fc_times_per_day);
        cloud_cov:long_name = "cloud cover";
        cloud_cov:units = "percent";

float   prob_fog(max_site_num, days, fc_times_per_day);
        prob_fog:long_name = "probability of fog";
        prob_fog:units = "percent";

float   prob_thunder(max_site_num, days, fc_times_per_day);
        prob_thunder:long_name = "probability of thunder";
        prob_thunder:units = "percent";

float   cprob_rain(max_site_num, days, fc_times_per_day);
        cprob_rain:long_name = "conditional probability of rain";
        cprob_rain:units = "percent";

float   cprob_snow(max_site_num, days, fc_times_per_day);
        cprob_snow:long_name = "conditional probability of snow";
        cprob_snow:units = "percent";

float   cprob_ice(max_site_num, days, fc_times_per_day);
        cprob_ice:long_name = "conditional probability of ice";
```

```
        cprob_ice:units = "percent";

float   prob_precip01(max_site_num, days, fc_times_per_day);
        prob_precip01:long_name = "probability of precipitation, 1 hr";
        prob_precip01:units = "percent";

float   prob_precip03(max_site_num, days, fc_times_per_day);
        prob_precip03:long_name = "probability of precipitation, 3 hr";
        prob_precip03:units = "percent";

float   prob_precip06(max_site_num, days, fc_times_per_day);
        prob_precip06:long_name = "probability of precipitation, 6 hr";
        prob_precip06:units = "percent";

float   prob_precip24(max_site_num, days, daily_time);
        prob_precip24:long_name = "probability of precipitation, 24 hr";
        prob_precip24:units = "percent";

float   qpf01(max_site_num, days, fc_times_per_day);
        qpf01:long_name = "amount of precipitation";
        qpf01:units = "meters";

float   qpf03(max_site_num, days, fc_times_per_day);
        qpf03:long_name = "amount of precipitation";
        qpf03:units = "meters";

float   qpf06(max_site_num, days, fc_times_per_day);
        qpf06:long_name = "amount of precipitation";
        qpf06:units = "meters";

float   wind_u(max_site_num, days, fc_times_per_day);
        wind_u:long_name = "eastward-component of wind";
        wind_u:units = "meters per second";

float   wind_v(max_site_num, days, fc_times_per_day);
        wind_v:long_name = "northward-component of wind";
        wind_v:units = "meters per second";

float   visibility(max_site_num, days, fc_times_per_day);
        visibility:long_name = "visibility";
        visibility:units = "km";

float   wind_speed(max_site_num, days, fc_times_per_day);
        wind_speed:long_name = "windspeed";
        wind_speed:units = "meters per second";
```

```
float   wind_speed_mph(max_site_num, days, fc_times_per_day);
        wind_speed_mph:long_name = "windspeed in mph";
        wind_speed_mph:units = "miles per hour";

float   wind_dir(max_site_num, days, fc_times_per_day);
        wind_dir:long_name = "wind direction clockwise from north";
        wind_dir:units = "degrees north";

float   rh(max_site_num, days, fc_times_per_day);
        rh:long_name = "relative humidity";
        rh:units = "decimal";

float   rh_pct(max_site_num, days, fc_times_per_day);
        rh_pct:long_name = "percent relative humidity";
        rh_pct:units = "percent";

float   precip_rate(max_site_num, days, fc_times_per_day);
        precip_rate:long_name = "precip (SWE) rate";
        precip_rate:units = "mm/hr";

float   precip_rate_inches(max_site_num, days, fc_times_per_day);
        precip_rate_inches:long_name = "precip (SWE) rate";
        precip_rate_inches:units = "in/hr";

float   precip_type(max_site_num, days, fc_times_per_day);
        precip_type:long_name = "precipitation type";
        precip_type:units = "0=NONE, 1=RAIN, 2=SNOW, 5=ICE";

float   precip_accum(max_site_num, days, fc_times_per_day);
        precip_accum:long_name = "3 hr precip accumulation";
        precip_accum:units = "mm";

float   precip_accum_inches(max_site_num, days, fc_times_per_day);
        precip_accum_inches:long_name = "3 hr precip accumulation";
        precip_accum_inches:units = "inches";

float   TempF(max_site_num, days, fc_times_per_day);
        TempF:long_name = "temperature";
        TempF:units = "degrees Fahrenheit";

float   dewptF(max_site_num, days, fc_times_per_day);
        dewptF:long_name = "dewpoint";
        dewptF:units = "degrees Fahrenheit";

float   snow_rate(max_site_num, days, fc_times_per_day);
        snow_rate:long_name = "snowfall rate";
```

```
                snow_rate:units = "mm/hr";

        float   snow_rate_inches(max_site_num, days, fc_times_per_day);
                snow_rate_inches:long_name = "snowfall rate";
                snow_rate_inches:units = "in/hr";

        float   snow_accum(max_site_num, days, fc_times_per_day);
                snow_accum:long_name = "3 hr snowfall accumulation";
                snow_accum:units = "mm";

        float   snow_accum_inches(max_site_num, days, fc_times_per_day);
                snow_accum_inches:long_name = "3 hr snowfall accumulation";
                snow_accum_inches:units = "inches";

        float   snow_accum_total(max_site_num, days, fc_times_per_day);
                snow_accum_total:long_name = "snowfall accumulation since start of
forecast";
                snow_accum_total:units = "mm";

        float   snow_accum_total_inches(max_site_num, days, fc_times_per_day);
                snow_accum_total_inches:long_name = "snowfall accumulation since
start of forecast";
                snow_accum_total_inches:units = "inches";

        float   snow_accum_48hr_total_inches(max_site_num, days, fc_times_per_day);
                snow_accum_48hr_total_inches:long_name = "48hr snowfall
accumulation since start of forecast";
                snow_accum_48hr_total_inches:units = "inches";

        float   prob_precip03_pct(max_site_num, days, fc_times_per_day);
                prob_precip03_pct:long_name = "probability of precipitation, 3 hr";
                prob_precip03_pct:units = "percent (0-100)";

        float   blowing_snow_potential(max_site_num, days, fc_times_per_day);
                blowing_snow_potential:long_name = "blowing snow potential";
                blowing_snow_potential:units = "index (0-3) (low-high)";

        float   blowing_snow_pot_vals(max_site_num, days, fc_times_per_day);
                blowing_snow_pot_vals:long_name = "blowing snow potential values";
                blowing_snow_pot_vals:units = "floating point (0-1) (low-high)";

        float   P_sfc(max_site_num, days, fc_times_per_day) ;
                P_sfc:long_name = "Pressure at 2m above sfc";
                P_sfc:units = "millibars" ;

        float   T_cb(max_site_num, days, fc_times_per_day) ;
```

```
        T_cb:long_name = "cloud base temp" ;
        T_cb:units = "degrees Celsius" ;

float   T_bls(max_site_num, days, fc_times_per_day);
        T_bls:long_name = "sub-sfc temperature";
        T_bls:units = "degrees Celsius";

float   T_lbls0(max_site_num, days, fc_times_per_day);
        T_lbls0:long_name = "0-10 cm layer sub-sfc temperature";
        T_lbls0:units = "degrees Celsius";

float   T_lbls1(max_site_num, days, fc_times_per_day);
        T_lbls1:long_name = "10-40 cm layer sub-sfc temperature";
        T_lbls1:units = "degrees Celsius";

float   T_lbls2(max_site_num, days, fc_times_per_day);
        T_lbls2:long_name = "40-100 cm layer sub-sfc temperature";
        T_lbls2:units = "degrees Celsius";

float   T_lbls3(max_site_num, days, fc_times_per_day);
        T_lbls3:long_name = "100-200 cm layer sub-sfc temperature";
        T_lbls3:units = "degrees Celsius";

float   snow_depth(max_site_num, days, fc_times_per_day);
        snow_depth:long_name = "water equiv of accum snow depth";
        snow_depth:units = "kg/m2";

float   cloud_low(max_site_num, days, fc_times_per_day);
        cloud_low:long_name = "low cloud layer amt";
        cloud_low:units = "decimal";

float   cloud_middle(max_site_num, days, fc_times_per_day);
        cloud_middle:long_name = "middle cloud layer amt";
        cloud_middle:units = "decimal";

float   cloud_high(max_site_num, days, fc_times_per_day) ;
        cloud_high:long_name = "high cloud layer amt" ;
        cloud_high:units = "decimal" ;

float   dlwrf_sfc(max_site_num, days, fc_times_per_day) ;
        dlwrf_sfc:long_name = "downward long wave radiation flux at surface" ;
        dlwrf_sfc:units = "W/m2" ;

float   dswrf_sfc(max_site_num, days, fc_times_per_day) ;
        dswrf_sfc:long_name = "downward short wave radiation flux at surface" ;
        dswrf_sfc:units = "W/m2" ;
```

```
        float   albedo_sfc(max_site_num, days, fc_times_per_day) ;
                albedo_sfc:long_name = "albedo at surface" ;
                albedo_sfc:units = "percent" ;
}
```

Observational History

Dimensions in the observational history data file:

Name	Length	Description
max_site_num		number of locations
num_times	48	

Data fields in the observational history data file:

Description	Unit
time at which observations file was created	seconds (since 1970-1-1 00:00:00)
time of earliest observations	seconds (since 1970-1-1 00:00:00)
number of actual sites	
forecast site id numbers	see Appendix E for assigned ID number and location for each observation site
cdl file type	
2m air temperature	degrees Celsius
dew point temperature	degrees Celsius
windspeed	meters per second
Presence of precipitation	0=None, 1=Precip
road surface temperature	degrees Celsius
bridge surface temperature	degrees Celsius
road condition	33=dry, 34=wet
subsurface temperature	degrees Celsius

NJ-MDSS requires an observational history at each forecast site for the road temperature model (METRo) to derive an initial estimate of the current road subsurface state. At least three hours of observations history are required, and twelve hours of history are optimal, but at a minimum there must be complete observations (no fields missing) for the current and preceding two hours before the forecast start time. The observational history file is in net CDF format and contains actual observations from the previous 24 hours, and estimated observations of the next 24 hours. The estimated observations is to keep MDSS functioning in case there is a disruption in the data feed.

```
netcdf obs_history {

dimensions:
	max_site_num = UNLIMITED ;	// number of locations
	num_times = 48 ;		// number of times

\variables:
	double creation_time ;
		creation_time:long_name = "time at which forecast file was created" ;
		creation_time:units = "seconds since 1970-1-1 00:00:00" ;

	double obs_time ;
		obs_time:long_name = "time of earliest obs" ;
		obs_time:units = "seconds since 1970-1-1 00:00:00" ;

	int num_sites ;
		num_sites:long_name = "number of actual_sites" ;

	int site_list(max_site_num) ;
		site_list:long_name = "forecast site id numbers" ;

	int type ;
		type:long_name = "cdl file type" ;

	float T(max_site_num, num_times) ;
		T:long_name = "2m air temperature" ;
		T:units = "degrees Celsius" ;

	float dewpt(max_site_num, num_times) ;
		dewpt:long_name = "dew point temperature" ;
		dewpt:units = "degrees Celsius" ;

	float wind_speed(max_site_num, num_times) ;
		wind_speed:long_name = "windspeed" ;
		wind_speed:units = "meters per second" ;

	int Precip(max_site_num, num_times) ;
		Precip:long_name = "Presence of Precip" ;
		Precip:units = "0=None, 1=Precip" ;

	float road_T(max_site_num, num_times) ;
		road_T:long_name = "road surface temperature" ;
		road_T:units = "degrees Celsius" ;

	float bridge_T(max_site_num, num_times) ;
		bridge_T:long_name = "bridge surface temperature" ;
```

```
        bridge_T:units = "degrees Celsius" ;

    int road_condition(max_site_num, num_times) ;
        road_condition:long_name = "road condition" ;
        road_condition:units = "33=dry, 34=wet" ;

    float subsurface_T(max_site_num, num_times) ;
        subsurface_T:long_name = "subsurface temperature" ;
        subsurface_T:units = "degrees Celsius" ;

data:
    type = 2;
}
```