

**User's Guide for the
*North American Animal Disease Spread Model 3.0***

For NAADSM version 3.0.79

June 1, 2006

Ashley Hill and Aaron Reeves

Animal Population Health Institute
College of Veterinary Medicine & Biomedical Sciences
Colorado State University
Fort Collins, CO 80523
 [<Ashley.Hill@colostate.edu>](mailto:Ashley.Hill@colostate.edu)
 [<Aaron.Reeves@colostate.edu>](mailto:Aaron.Reeves@colostate.edu)
 [<http://www.cvmb.colostate.edu/aphi>](http://www.cvmb.colostate.edu/aphi)



User's Guide for the North American Animal Disease Spread Model 3.0

Copyright © 2005 - 2006 Animal Population Health Institute at Colorado State University.

Permission is granted to copy, distribute and/or modify this document under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike 2.5 License. A copy of this license is included in the appendix entitled **Creative Commons Public License**.

The NAADSM development team

Barbara A. Corso (*Primary contact, US*)
USDA-APHIS-VS-CEAH-CADIA

Caroline Dubé (*Primary contact, Canada*)
CFIA – Animal Health and Production Division

Neil Harvey
Department of Computing & Information Science, University of Guelph

Ashley E. Hill
Animal Population Health Institute, Colorado State University

W. Bruce McNab
Ontario Ministry of Agriculture, Food, and Rural Affairs

Aaron Reeves
Animal Population Health Institute, Colorado State University

M.D. Salman
Animal Population Health Institute, Colorado State University

Ann H. Seitzinger
USDA-APHIS-VS-CEAH-NCHAS-NAHMS

Deborah A. Stacey
Department of Computing & Information Science, University of Guelph

Francisco Zagmutt-Vergarra
Animal Population Health Institute, Colorado State University

and
Mark A. Schoenbaum
USDA-APHIS-VS-WRO

Complete contact information for members of the *NAADSM* development team may be found in Appendix G.

North American Animal Disease Spread Model 3.0

Copyright © 2004 – 2006 Colorado State University and University of Guelph

The *North American Animal Disease Spread Model (NAADSM)* is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 2 of the License, or (at your option) any later version. A copy of this license is included in the appendix entitled **GNU General Public License**.

Table of contents

The NAADSM development team.....	v
Table of contents	vii
List of figures.....	xv
List of tables.....	xix
Document conventions.....	xxi
Errata.....	xxiii
1. Introduction to NAADSM.....	1
1.1. A very brief history of NAADSM	1
1.2. Purposes of this guide	1
1.3. Purposes and uses of NAADSM	2
1.4. Capabilities of NAADSM	2
1.5. Limitations of NAADSM	2
1.6. Overview of simulation parameters	3
1.7. Overview of simulation output	4
2. System requirements	5
3. Basics of stochastic modeling	7
4. Installing and registering NAADSM.....	11
4.1. Installing NAADSM.....	11
4.2. Registering NAADSM	13
5. Running your first simulation	15
5.1. The starting population	15
5.2. The disease scenario	16
5.3. Launching the simulation.....	17
5.4. Viewing simulation output.....	18
6. Types of input parameters	21
6.1. Basic types	21
6.1.1. Yes/no values	21
6.1.2. Integer values	21
6.1.3. Floating point numbers	21
6.1.4. Probabilities	21
6.1.5. Probability density functions	22
6.1.5.1. Creating and editing <i>pdfs</i>	22
6.1.5.1.1. The <i>pdf</i> selection box	22
6.1.5.1.2. The <i>pdf</i> editor window	22
6.1.5.1.3. Creating and editing piecewise <i>pdfs</i>	25
Adding a new point.....	26
Moving an existing point	26

Changing the <i>x</i> axis	29
Deleting an existing point	29
Directly editing point values	29
6.1.5.1.4. Importing a piecewise <i>pdf</i>	29
6.1.5.2. Exporting a piecewise <i>pdf</i>	30
6.1.5.3. Saving, copying, or printing <i>pdf</i> images	30
6.1.6. Relational functions	31
6.1.6.1. The <i>rel</i> selection box	31
6.1.6.2. The <i>rel</i> editor window	31
6.1.6.3. Creating and editing <i>rel</i> functions	32
Adding a new point	32
Moving an existing point	32
Changing the axes	32
Deleting an existing point	33
Directly editing point values	33
6.1.6.4. Importing a <i>rel</i> function	33
6.1.6.5. Exporting a <i>rel</i> function	34
6.1.6.6. Saving, copying, or printing images of <i>rel</i> functions	34
6.2. Complex types	35
6.2.1. Herds or units	35
6.2.2. Production types	35
6.2.3. Production type combinations	36
7. Basic file operations: File menu commands	37
<i>NOTE: NAADSM scenario files and Microsoft Access</i>	37
7.1. New scenario file	37
7.2. Open scenario file	37
<i>NOTE: Compacting NAADSM scenario files</i>	38
7.3. Export scenario	38
7.4. Save scenario file	38
<i>NOTE: Using an NAADSM scenario file simultaneously in several applications</i>	38
7.5. Save As	38
7.6. Close scenario file	39
7.7. Exit	39
8. Setting up a scenario: The Scenario parameters menu	41
8.1. Using the wizard screens	41
8.2. The Start setup window	42
8.3. The Production type(s) window	44
<i>WARNING: Use care when removing production types</i>	44
<i>NOTE: Production type ID numbers</i>	45
8.4. Setting up a population of herds: The View/edit starting units window	45
<i>NOTE: Editing large herd data sets</i>	45
8.4.1. Editing the attributes of a herd	46
8.4.1.1. Changing basic attributes	46
<i>NOTE: Herd order and ID</i>	47
8.4.1.2. Changing the number of days left in status	47
8.4.2. Sorting and filtering the list of herds	48

8.4.2.1	Sorting herds	48
8.4.2.2.	Filtering herds by simple criteria	49
8.4.2.3.	Filtering herds by geographic range	50
8.4.3.	Importing and exporting herds.....	52
8.4.3.1.	File formats for import and export of herd data.....	52
	<i>NOTE: Production types in imported herd files.....</i>	53
8.4.3.2.	Import and replace existing herds	54
8.4.3.3.	Import and append to existing herd list.....	55
8.4.3.4.	Exporting herds.....	55
8.4.4.	Deleting herds	56
8.4.5.	Finalizing changes to starting units.....	56
8.5.	Modeling disease progression.....	57
8.5.1.	Concepts of disease progression in NAADSM	57
8.5.2.	Setting disease parameters.....	58
8.6.	Modeling disease spread	60
8.6.1.	The Spread options window	60
	<i>NOTE: No spread.....</i>	60
8.6.2.	The Production type combinations window.....	61
8.6.3.	The Contact spread window	62
8.6.3.1.	Direct versus indirect contact spread	63
8.6.3.2.	Contact rates.....	63
	<i>NOTE: Fixed initial contact rates.....</i>	64
8.6.3.3.	Setting contact spread parameters.....	64
8.6.3.3.1.	Direct contact spread parameters	65
	<i>WARNING: Shipping delay.....</i>	65
8.6.3.3.2.	Indirect contact spread parameters.....	66
8.6.4.	Airborne spread.....	66
8.6.4.1.	Linear versus exponential decline in probability of spread	66
8.6.4.2.	The Airborne spread window.....	67
8.7.	Modeling disease detection.....	69
8.7.1.	Description of disease detection	69
	<i>NOTE: Baseline probability of reporting disease</i>	70
8.7.2.	Setting disease detection parameters	70
8.8.	Modeling surveillance.....	72
8.8.1.	Description of surveillance	72
8.8.2.	Setting surveillance parameters	72
8.9.	Modeling destruction for disease control.....	74
8.9.1.	Global destruction options	74
8.9.2.	Destruction priorities	75
8.9.2.1.	Setting destruction priorities	76
8.9.2.2.	Examples of destruction priorities	77
8.9.3.	Production type settings for destruction	78
8.10.	Modeling vaccination.....	79
8.10.1.	Global vaccination options	79
8.10.2.	Vaccination priorities.....	81
8.10.2.1.	Setting vaccination priorities	81

<i>NOTE: Reason for vaccination</i>	82
8.10.3. Production type settings for vaccination.....	82
8.10.3.1. Minimum time between vaccinations	84
8.11. Cost accounting.....	84
8.11.1. Costs associated with destruction	86
8.11.2. Costs associated with vaccination.....	86
8.12. Specifying output options	87
<i>WARNING: Saving all daily output</i>	89
8.13. Using the Check scenario validity command	89
<i>NOTE: Attempting to run an invalid scenario</i>	90
8.14. Using the Clear scenario output command.....	90
<i>WARNING: Clearing scenario output</i>	90
9. Launching a simulation: The Run menu	91
10. Viewing model output: The Output windows menu	93
<i>NOTE: Obtaining raw data for simulation outputs</i>	93
10.1. “Actual” versus “apparent” events.....	94
10.2. Dynamic output windows for a single iteration.....	94
10.2.1. Map of units for 1 iteration	94
10.2.1.1. Map options	95
10.2.2. Daily unit status for 1 iteration	96
10.2.3. Summary of 1 iteration	97
10.2.3.1. Epidemiological outputs	98
10.2.3.1.1. Detection	99
10.2.3.1.2. Destruction.....	99
10.2.3.1.3. Vaccination	99
10.2.3.1.4. Reasons for infection	99
10.2.3.1.5. Actual and apparent epidemic curves	100
10.2.3.2. Cost accounting outputs.....	102
10.2.3.2.1. Table of daily costs	102
10.2.3.2.2. Graph of daily costs	103
10.3. Daily events and exposures.....	103
<i>NOTE: Enabling daily events and exposures</i>	103
10.3.1. The Events for 1 iteration window.....	104
<i>NOTE: Events on day “-1”</i>	106
10.3.1.1. Sorting events.....	106
10.3.1.2. Filtering events.....	108
10.3.2. The Exposures for 1 iteration window	109
<i>NOTE: Interpreting exposure success</i>	110
10.3.2.1. Sorting and filtering exposures	110
10.4. Summarizing output from multiple iterations.....	110
10.4.1. The Output statistics window.....	111
10.4.1.1. Epidemiological and cost accounting outputs.....	111
10.4.1.2. The output table	113
<i>NOTE: Variable definitions</i>	113
<i>NOTE: Obtaining the data used to generate summary statistics</i>	114
10.4.1.3. The histogram	114

10.4.1.4. The convergence plot.....	116
10.4.2. The Summary epidemic curves window	116
10.4.2.1. Summary epidemic curves by example	117
<i>NOTE: Interpreting summary epidemic curves</i>	119
10.4.2.2. Using the Summary epidemic curves graphical view	119
10.4.2.3. Using the Summary epidemic curves tabular view	120
10.5. Common features of the NAADSM output windows	121
10.5.1. Changing the production type	122
10.5.2. Saving, copying, or printing data.....	122
10.5.3. Saving, copying, or printing charts and graphs.....	123
10.6. Arranging the output windows.....	123
10.6.1. The Cascade command	123
10.6.2. The Arrange command.....	124
10.6.3. The Close all windows command	124
11. The Help menu	125
<i>NOTE: NAADSM version numbers</i>	126
<i>NOTE: Online help</i>	126
Appendix A. NAADSM 3.0 model description.....	127
Contents of appendix A	129
A1. Introduction.....	130
Keywords	130
A2. Basics	130
A3. Disease	131
Disease parameters.....	132
A4. Spread	132
A4.1. Direct contact spread.....	132
Parameters for direct contact spread	135
Examples of direct contact spread	136
Example 1	136
Example 2	136
Example 3	137
Example 4	137
A4.2. Indirect contact spread	137
Parameters for indirect contact spread	137
A4.3. Airborne spread.....	138
Parameters for airborne spread	139
A5. Detection	139
Detection parameters	140
A6. Surveillance.....	140
A6.1 Trace surveillance	140
Trace surveillance parameters.....	141
A7. Control measures	142

A7.2. Destruction	142
Destruction parameters	142
A7.2.1 Destruction capacity.....	143
A7.2.2 Destruction priorities.....	143
Examples of destruction priorities	144
A7.3. Vaccination	145
Vaccination program parameters	145
A7.3.1 Initiation of a vaccination program.....	146
A7.3.2 Vaccination capacity	146
A7.3.3 Vaccination priorities	146
A7.3.4 Minimum time between vaccinations	147
A8. Biological effects of vaccination	148
Parameters for effects of vaccination.....	148
A9. Priorities of action	148
A10. Costs	149
A10.1 Costs associated with destruction	149
Parameters for destruction costs	149
A10.2 Costs associated with vaccination.....	150
Parameters for vaccination costs.....	150
Appendix B. NAADSM outputs	151
B1. Summary outputs for each iteration	151
B2. Additional characteristics of each iteration.....	156
B3. Daily outputs	156
B4. Epidemic curves	163
Appendix C. Plain text comma-delimited (*.csv) files used in NAADSM	165
C1. Files containing population (herd or unit) data	165
C1.1. Importing population data	165
C1.2. Population files exported by NAADSM	166
C1.3. Sample population files.....	167
C2. Files containing piecewise probability density functions	168
C3. Files containing relational functions	169
Appendix D. Probability density functions supported by NAADSM	171
D1. The Beta distribution.....	171
D2. The BetaPERT distribution.....	172
D3. The Exponential distribution.....	173
D4. The Gamma distribution	174
D5. The Gaussian (Normal) distribution	174
D6. The Logistic distribution.....	175
D7. The Loglogistic distribution.....	175
D8. The Lognormal distribution	176
D9. The Pearson 5 distribution	176
D10. The Piecewise (General) distribution.....	177
D11. The Point “distribution”	177
D12. The Poisson distribution.....	177

D13. The Triangular distribution	178
D14. The Uniform distribution	179
D15. The Weibull distribution	180
D16. References	181
Appendix E. Scenario file database schema	183
E1. Table types in <i>NAADSM</i> scenario files	183
E2. The DBSchemaVersion database table	183
E3. Database tables for scenario input	185
E3.1. Database table inGeneral	185
E3.2. Database table inProductionType	186
E3.3. Database table inProductionTypePair	189
E3.4. Database table inDiseaseSpread	190
E3.5. Database table inControlsGlobal	192
E3.6. Database table inChart	193
E3.7. Database table inChartDetail	194
E4. The dynamic table dynHerd	194
E5. Database tables for scenario output	196
E5.1. Database table outGeneral	196
E5.2. Daily data tables	197
E5.2.1. Database table outDailyByProductionType	197
E5.2.2. Database table outDailyEvents	197
E5.2.2. Database table outDailyExposures	198
E5.3. Iteration data tables	199
E5.3.1. Database table outIteration	199
E5.3.2. Database table outIterationByProductionType	199
E5.3.3. Database table outIterationCosts	200
E5.3.4. Database table outEpidemicCurves	200
E6. Read-only tables	201
E6.1. Database table readStatusCodes	201
E6.2. Database table readApparentStateCodes	202
E6.3. Database table readEventCodes	202
E6.4. Database table readSpreadMethodCodes	203
Appendix F. NAADSM/PC and NAADSM/SC	205
F1. The two implementations of <i>NAADSM</i>	205
F2. Building <i>NAADSM/SC</i>	205
F3. Developing scenarios for <i>NAADSM/SC</i>	205
F4. Viewing output from <i>NAADSM/SC</i>	206
Appendix G. Development team contact information	207
Appendix H. The Creative Commons Public License	209
General terms of the Attribution-NonCommercial-ShareAlike 2.5 license	209
Creative Commons web site	209
Legal code of the Attribution-NonCommercial-ShareAlike 2.5 license	209
Appendix I. The GNU General Public License	215

List of figures

Figure 3-1. Distribution of height among males in the United States.....	7
Figure 3-2. Output of 1000 iterations of a stochastic model estimating the amount of fabric needed to make a pair of pants.....	8
Figure 3-3. Number of herds that can be vaccinated per day as a function of time since the start of an outbreak.....	9
Figure 4-1. The first installation message displayed by the <i>NAADSM</i> setup package.....	11
Figure 4-2. The final installation message displayed by the <i>NAADSM</i> setup package.....	11
Figure 4-3. Messages displayed by the <i>NAADSM</i> update package.....	12
Figure 4-4. The <i>NAADSM</i> icon created on the <i>Windows Start</i> menu.....	12
Figure 5-1. Initial map of herds in the US Midwest scenario.....	16
Figure 5-2. Menu items for viewing scenario parameters.....	17
Figure 5-3. Map showing herds and their disease states for an iteration in progress.....	18
Figure 5-4. Chart showing the number of herds in the various disease stages over the course of a single iteration.....	19
Figure 6-1. Selection box for probability density functions.....	22
Figure 6-2. The <i>pdf</i> editor window.....	23
Figure 6-3. The parameters of the triangular <i>pdf</i> have changed.....	24
Figure 6-4. There is an error in the specified parameters.....	24
Figure 6-5. A piecewise <i>pdf</i>	25
Figure 6-6. A new piecewise <i>pdf</i>	26
Figure 6-7. Adding a new point to a piecewise <i>pdf</i>	27
Figure 6-8. Moving an existing point of a piecewise <i>pdf</i>	28
Figure 6-9. A properly formatted *.csv file for importing a piecewise <i>pdf</i>	30
Figure 6-10. Selection box for relational functions.....	31
Figure 6-11. The <i>rel</i> editor window.....	32
Figure 6-12. A new relational function.....	33
Figure 6-13. A properly formatted *.csv file for importing a relational function.....	34
Figure 8-1. The “wizard” commands of the Scenario parameters menu.....	41
Figure 8-2. The “wizard” buttons.....	42
Figure 8-3. The Start setup wizard screen, activated by the Scenario parameters → Start setup command.....	43
Figure 8-4. The Production type(s) wizard screen, activated by the Scenario parameters → Production type(s) command.....	44
Figure 8-5. The View/edit starting units screen, activated by the Scenario parameters → Set up starting units command.....	46
Figure 8-6. Changing the production type and initial disease status of a herd.....	47
Figure 8-7. Sorting herds by production type.....	49
Figure 8-8. Filtering herds by herd size.....	50
Figure 8-9. Filtering herds by rectangular geographic range.....	51
Figure 8-10. Filtering herds by circular geographic range.....	52
Figure 8-11. Part of a *.csv file suitable for import into an <i>NAADSM</i> scenario.....	54
Figure 8-12. The View/edit starting units toolbar.....	54

Figure 8-13. Herd file export options.....	55
Figure 8-14. Selecting and deleting a herd.....	56
Figure 8-15. The disease cycle in <i>NAADSM</i>	58
Figure 8-16. The Disease window.....	59
Figure 8-17. Commands of the Scenario parameters → Disease spread menu.....	60
Figure 8-18. The Production type combinations window.....	61
Figure 8-19. The Contact spread window.....	63
Figure 8-20. Probability of disease spread by airborne transmission at various distances calculated by the linear decrease and the exponential decrease algorithm.....	67
Figure 8-21. The Airborne spread window.....	69
Figure 8-22. Calculating the overall chance of detection and reporting, based on the probability of observing clinical signs and the probability of reporting a detected unit.....	70
Figure 8-23. Commands of the Scenario parameters→Detection menu.....	71
Figure 8-24. The Detection window.....	71
Figure 8-25. Commands of the Scenario parameters→ Surveillance menu.....	72
Figure 8-26. The Surveillance window.....	73
Figure 8-27. Commands of the Scenario parameters→ Destruction menu.....	74
Figure 8-28. The Global destruction options window.....	75
Figure 8-29. The Destruction priorities window.....	76
Figure 8-30. The Destruction window.....	78
Figure 8-31. Commands of the Scenario parameters→ Vaccination menu.....	79
Figure 8-32. The Global vaccination options window.....	80
Figure 8-33. The Vaccination priorities window.....	81
Figure 8-34. The Vaccination window.....	83
Figure 8-35. Overlapping vaccination rings.....	84
Figure 8-36. Commands of the Scenario parameters → Cost accounting menu.....	85
Figure 8-37. The Cost accounting window.....	85
Figure 8-38. The Output options window.....	88
Figure 8-39. A sample plain text file showing daily herd (unit) states.....	89
Figure 9-1. Commands of the Run menu.....	91
Figure 10-1. Commands of the Output windows menu.....	93
Figure 10-2. A map of herds from a simulation in progress.....	95
Figure 10-3. A map of herds from a simulation in progress, showing apparent status.....	96
Figure 10-4. The Daily unit status for 1 iteration window.....	97
Figure 10-5. The Epidemiology tab of the Summary of 1 iteration window.....	98
Figure 10-6. Reasons for infection, as reported in the summary of 1 iteration.....	100
Figure 10-7. Apparent and actual epidemic curves.....	101
Figure 10-8. The Cost accounting tab of the Summary of 1 iteration window.....	102
Figure 10-9. The Events for 1 iteration window.....	105
Figure 10-10. Sorting events by event type.....	107
Figure 10-11. Filtering events by day.....	108
Figure 10-12. The Exposures for 1 iteration window.....	109
Figure 10-13. The Output statistics window.....	112
Figure 10-14. Displaying output definitions.....	113
Figure 10-15. Qualitative variables on the Epidemiology tab.....	114
Figure 10-17. Convergence plot generated for the mean of tscULat data from 10 iterations...	116

Figure 10-18. Epidemic curves (number of new cases by day) generated for three of the individual iterations shown in Table 10-1	117
Table 10-1. Sample data for the generation of summary epidemic curves.....	118
Figure 10-19. The Summary epidemic curves window, showing summary curves for the data from Table 10-1	119
Figure 10-20. The Tabular view tab of the Summary epidemic curves window.....	120
Figure 10-21. A typical output window toolbar.....	122
Figure 10-22. Schematic diagram showing output window arrangement after using Cascade or Arrange	124
Figure 11-1. The About NAADSM screen.....	125
Figure C-1. Default *.csv file format generated by <i>NAADSM</i> 3.0.63 (and higher).....	167
Figure C-2. A *.csv file generated by <i>NAADSM</i> 3.0.63 (and higher) using production type ID number	167
Figure C-3. A *.csv file generated by <i>NAADSM</i> 3.0.63 (and higher) using numeric transition state code.	167
Figure C-4. A *.csv file generated by <i>SpreadModel</i>	168
Figure C-5. A *.csv file for a piecewise pdf.	168
Figure C-6. A properly formatted *.csv file for importing a relational function.	169
Figure D-1. A sample Beta distribution.....	172
Figure D-2. Two sample BetaPERT distributions.	173
Figure D-3. A sample Exponential distribution.....	173
Figure D-4. A sample Gamma distribution.....	173
Figure D-5. A sample Exponential distribution.....	174
Figure D-6. A sample Logistic distribution.	174
Figure D-7. A sample Loglogistic distribution.	175
Figure D-8. A sample Lognormal distribution.....	175
Figure D-9. Two sample Pearson 5 distributions.....	176
Figure D-10. Two sample Piecewise distributions.	177
Figure D-11. Two sample Poisson distributions.	178
Figure D-12. Two sample Triangular distributions.	179
Figure D-13. A sample Uniform distribution.....	179
Figure D-14. Two sample Weibull distributions.	180
Figure E-1. Database schema diagram for <i>NAADSM</i> scenario files.....	184

List of tables

Table 8-1. Fields in <i>NAADSM *.csv</i> files.....	53
Table 8-2. Character codes used for disease transition states	53
Table 8-3. Disease and disease-like transition states used in <i>NAADSM</i>	57
Table 10-2. Events recorded by <i>NAADSM</i>	104
Table 10-3. Columns of the daily events window.....	105
Table 10-4. Columns of the daily exposures table.....	110
Table B-1. Summary epidemiological outputs for each iteration.....	151
Table B-2. Summary cost accounting outputs for each iteration.....	156
Table B-3. Daily epidemiological outputs.....	157
Table C-1. Required fields for <i>NAADSM *.csv</i> files.....	165
Table C-2. Codes used for disease transition states	166
Table E-1. Fields of database table DBSchemaVersion	183
Table E-2. Fields of the database table inGeneral	185
Table E-2. Fields of the database table inProductionType	186
Table E-3. Fields of the database table inProductionTypePair	189
Table E-4. Fields of the database table inDiseaseSpread	190
Table E-5. Fields of the database table inControlsGlobal	192
Table E-6. Fields of the database table inChart	193
Table E-7. Fields of the database table inChartDetail	194
Table E-8. Fields of the database table dynHerd	194
Table E-9. Fields of the database table outGeneral	196
Table E-10. Fields of the database table outDailyByProductionType	197
Table E-11. Fields of the database table outDailyEvents	197
Table E-12. Fields of the database table outDailyExposures	198
Table E-13. Fields of the database table outIteration	199
Table E-14. Fields of the database table outIterationByProductionType	199
Table E-15. Fields of the database table outIterationCosts	200
Table E-16. Fields of the database table outEpidemicCurves	201
Table E-17. Fields of the database table readStatusCodes	201
Table E-18. Values in the database table readStatusCodes	201
Table E-19. Fields of the database table readApparentStateCodes	202
Table E-20. Values in the database table readApparentStateCodes	202
Table E-21. Fields of the database table readEventCodes	202
Table E-22. Values in the database table readEventCodes	202
Table E-23. Fields of the database table readSpreadMethodCodes	203
Table E-24. Values in the database table readSpreadMethodCodes	203

Document conventions

The following font conventions are used throughout this guide:

Italic type

is used to denote the names of computer programs, such as *NAADSM* and *Microsoft Windows*.

Constant width bold type

is used to indicate a file or path name, such as **usMidwest.mdb**.

Sans serif bold type

indicates a menu command, the caption of a button, or some other text that will be displayed in a computer program. An underlined letter in this kind of label (e.g., the “F” in **File**) indicates the “hot key” associated with that menu command: rather than selecting the menu, you can use the ALT key in combination with the underlined letter to activate the command.

A menu and command may be shown with an arrow (→) separating them, as in **File** → **Open scenario file...**. This indicates that you should select the menu (**File**, in this case), and then the command that appears on the menu (such as **Open scenario file...**).

SMALL CAPS

indicate the name of a particular key to press, such as ALT or ENTER.

Boxes with sans serif type

contain useful notes and important warnings.

Fixed width type

shows the contents of a plain text file.

<Angled brackets>

are used to indicate email, website, and other Internet-related addresses.

Sans serif bold type

is also used, particularly in the appendices, to indicate table names in *NAADSM* scenario database files.

Errata

Updated versions of this section will be released as errors are found and corrected in the *NAADSM User's Guide*.

1. Introduction to NAADSM

1.1. A very brief history of NAADSM

The *North American Animal Disease Spread Model* has its origins in animal disease modeling efforts within the US Department of Agriculture's Centers for Epidemiology and Animal Health (CEAH), which date back nearly a decade. The animal disease modeling project at CEAH began with the evaluation of models developed elsewhere¹ and grew to encompass the creation of several versions of a new animal disease model for use in North America². Each of these new versions was developed with the primary objective of producing a comprehensive, user-friendly modeling application for the evaluation of control measures for foreign infectious animal diseases, preparedness planning, and training exercises.

Version 3.0 of the *North American Animal Disease Spread Model (NAADSM)* is latest in this line of animal disease models and is a much enhanced successor to earlier versions. This version has been developed through an international collaboration involving researchers from governmental and academic institutions in the United States and Canada, along with invaluable contributions from subject matter experts from all over the world.

NAADSM is still under active development: addition of new components and capabilities, as well as constant refinement of existing features, is ongoing. For more information about current *NAADSM* modeling activities, please contact one of the development team members (see Appendix G for contact information).

1.2. Purposes of this guide

This guide is designed to acquaint researchers, analysts, and decision makers in the fields of animal health economics and veterinary epidemiology with the principles and applications of stochastic modeling. The treatment of stochastic modeling provided here is clearly not definitive, but should be adequate to prepare readers to understand and evaluate the results of an *NAADSM* simulation.

More specifically, this manual is intended to provide instruction in the actual operation of *NAADSM*. After reading this guide, users should be able to run the program and perform all major tasks that it offers.

Finally, this guide is designed to provide a detailed reference. Each section should stand alone (more or less) for advanced users who need specific information about a particular topic

¹ Garner, M.G., and Lack, M.B. 1995. An evaluation of alternate control strategies for foot-and-mouth disease in Australia: a regional approach. *Prev. Vet. Med.* 23: 9-32.

² Schoenbaum, M.A., and Disney, W.T. 2003. Modeling alternative mitigation strategies for a hypothetical outbreak of foot-and-mouth disease in the United States. *Prev. Vet. Med.* 58: 25-52.

1. Introduction to *NAADSM*

rather than a beginning-to-end “how to” resource. The appendices provide a comprehensive list of the parameters required by *NAADSM*, a complete description of the conceptual model behind the application, and a thorough description of all model outputs available for analysis.

1.3. Purposes and uses of *NAADSM*

NAADSM simulates outbreaks of foot and mouth disease and other contagious animal diseases. As a training tool, *NAADSM* can be used to prepare emergency personnel and other first responders for disease outbreaks. *NAADSM* can also be used to demonstrate to policy makers the potential scope and impact of an outbreak in their region. As a research tool, *NAADSM* can be used to compare disease control strategies and estimate the resources needed in the event of an outbreak.

NAADSM is not recommended for use during an outbreak to predict direction and magnitude of disease spread, or during an outbreak as the basis for policy decisions. Information on disease incidence early in an outbreak is typically inaccurate. Simulated outbreaks based on inaccurate information will be useless, and may provide a false sense of security regarding the scope and control of the outbreak.

1.4. Capabilities of *NAADSM*

NAADSM is a flexible tool for simulating the temporal and spatial spread of foot and mouth disease and other contagious animal diseases at the herd level in a population of naïve animals. Users may import their own population data and adjust all parameters related to animal population, disease transmission, and disease control to closely mimic the situation in their region of interest. Users may also employ animal data and parameters provided with the program and which mimic the American Midwest.

1.5. Limitations of *NAADSM*

NAADSM is not capable of simulating chronic, endemic, fatal, vertically transmitted, sexually transmitted, or vector-borne diseases; diseases with environmental or biologically adapted reservoirs; or diseases that do not confer post-infection immunity.

As mentioned above, *NAADSM* is not intended to be used in the face of an outbreak to predict magnitude or direction of disease spread.

1.6. Overview of simulation parameters

Users should be familiar with the parameters described in the *NAADSM* model description (Appendix A). This appendix gives a complete, detailed list of all input parameters, and describes how each parameter is used in the course of a simulation. Input parameters apply to the following broad categories:

Animal populations. Herds may be classified based on user-defined production types. A production type is a collection of herds with similar disease transmission probabilities, disease manifestation, disease detection probabilities, and control strategies. The user defines the rates of contact within and between production types and the probability of airborne transmission to better mimic disease spread. Production types are typically defined by animal species and/or management practices applied to each herd. Production types may be broad (*e.g.*, cattle, swine, poultry) or narrow (*e.g.*, beef sale yard, beef cow-calf operation <50 head, beef cow-calf operation >50 head, beef feedlot).

Disease manifestation. There are five discrete disease states used in *NAADSM*. During a simulation, herds transition through these disease states according to user-defined parameters. These states are susceptible to disease, latently infected, infectious and subclinically ill, infectious and clinically ill, and immune. For each production type, the user defines the length in days of these periods using probability distributions. All animals within a herd are assumed to exist in the same disease state.

Disease transmission. *NAADSM* was designed to simulate foot and mouth disease, but can be used to simulate any contagious animal disease that is spread primarily through direct and indirect contact and/or aerosol transmission. User-defined parameters define the probabilities of transmission via each method among and within production types.

Disease detection and surveillance. *NAADSM* allows the user to simulate both passive and active disease surveillance. Passive surveillance is a function of the length of time since the outbreak began and the length of time that a given herd has been infected. Active surveillance targets contacts from infected herds.

Disease control. *NAADSM* simulates three methods of disease control: ring vaccination, movement restriction, and destruction. For each production type, the user defines the conditions under which these measures are adopted, the time to and length of vaccine-induced immunity, and the number of herds that can be vaccinated or destroyed per day. Vaccination is assumed to be 100% effective.

Direct costs. *NAADSM* can be used to estimate the direct costs associated with each simulated outbreak. User inputs include itemized costs associated with vaccination and destruction.

1. Introduction to *NAADSM*

1.7. Overview of simulation output

NAADSM provides results for single model iterations as well as summary results derived from multiple iterations. *NAADSM* generates a daily summary of each outbreak simulated, an epidemic curve, a breakdown of costs associated with the outbreak, and dozens of individual statistics (days to end of outbreak, number of herds and animals infected, number of herds and animals vaccinated, costs associated with vaccination, *etc.*). Model outputs are discussed in more detail in Section 10. A complete description of all available model outputs is provided in Appendix B.

NAADSM summarizes the results over all iterations using histograms, measures of central tendency and spread, and graphical depictions of convergence.

2. System requirements

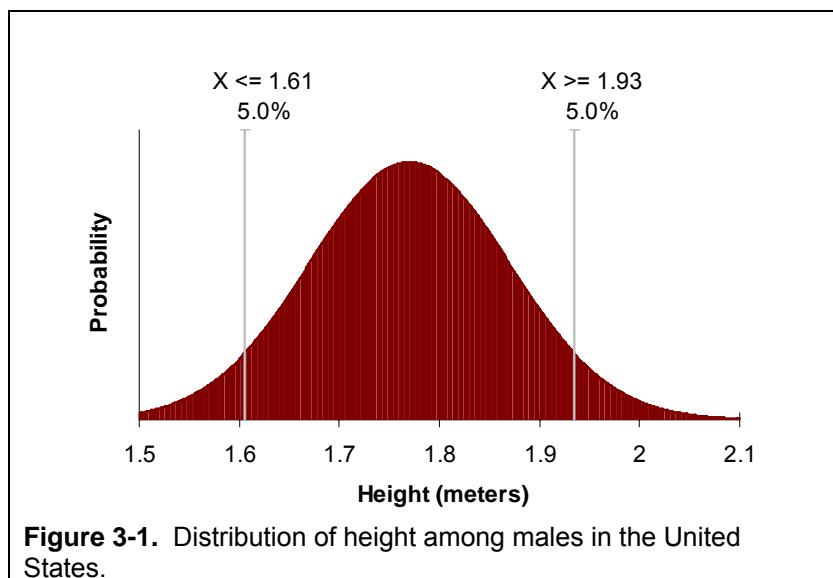
Two distinct but fully compatible and largely interchangeable implementations of *NAADSM* are available. This guide is intended for users of the personal computer version of *NAADSM* for *Microsoft Windows*. (This version is sometimes referred to as *NAADSM/PC*, where “PC” stands for “personal computer”.) Additional information for users of advanced computational systems and environments is available in Appendix F.

NAADSM/PC is supported on *Microsoft Windows ME*, *2000*, *XP*, and higher. At least 128 MB of RAM and 50 MB of available hard disk space are required. A faster computer with more memory will allow users to run larger and more sophisticated scenarios. The recommended system configuration for *NAADSM* is as follows: the equivalent of an Intel Pentium III or Pentium IV processor (or better), *Windows 2000* or *XP*, 512 MB or more of RAM, and 150 MB of available hard disk space.

3. Basics of stochastic modeling

Biological processes display inherent variability. Take height, for example: height for human males can range from 5 to 7 feet (1.5 to 2.1 meters), although the average is close to 5 feet, 9 inches (1.77 meters) in the United States. If you were to develop a computer model to predict the amount of cloth needed to make some number of pairs of pants, and you needed to include human male height as a variable, you could do so in two ways. The first method would incorporate height as a single value, *i.e.* 5 feet 9 inches, and assume that all men are that height. This is called *deterministic modeling*, because the value of the variable height has already been determined. If you wanted to estimate the amount of fabric needed to make 1000 pairs of pants with a deterministic model, you would run the model once to calculate the amount of fabric needed for one pair, then multiply the results by 1000. Running the model additional times would not provide any new information: the input variable is fixed, and the result will be the same each time.

A second approach would be to incorporate height as a distribution of values representative of the natural range (see Figure 3-1, below). Now, if the model is run several times, and if a different value is drawn at random from this distribution for each run, the results will be slightly different each time. This is a *stochastic model* (stochastic is defined as “being or having a random variable”) because height is no longer a fixed value, but is drawn randomly from a distribution. Values with higher probability of occurring (*i.e.* heights close to 5’9”) are more likely to be drawn than values at the extreme ends (“tails”) of the distribution. In Figure 3-1, we can see that values below 1.61 meters (5’3”) are drawn 5% of the time, and values above 1.93 meters (6’4”) are drawn 5% of the time. Thus, 90% of the time, the value of height will be between 5’3” and 6’4” (1.61 and 1.93 meters).



In a stochastic model, both the inputs and the outputs are distributions. Running the stochastic model from our example once is not particularly informative. However, by running

3. Basics of stochastic modeling

the model 1000 times (running 1000 “iterations”), we get 1000 different estimates of the fabric needed for one pair of pants. These estimates form a distribution (Figure 3-2). To estimate the fabric needed for 1000 pairs of pants, we can sum the 1000 estimates. Results of stochastic models incorporate variability, whereas results from deterministic models do not. We can improve our model by incorporating additional stochastic variables, such as variability in the amount of fabric used per inch of height (which is affected by waist size and pants style).

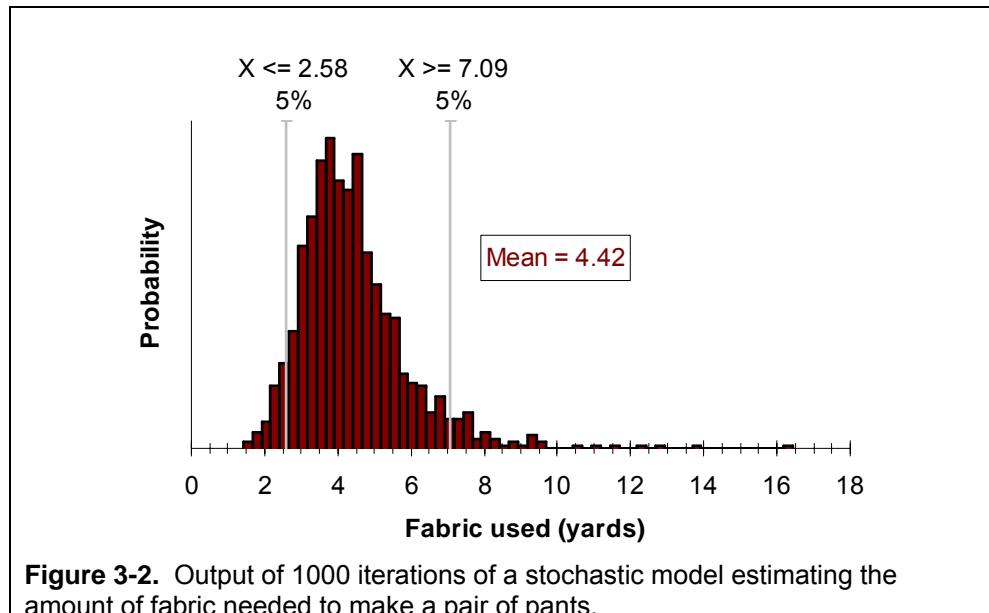
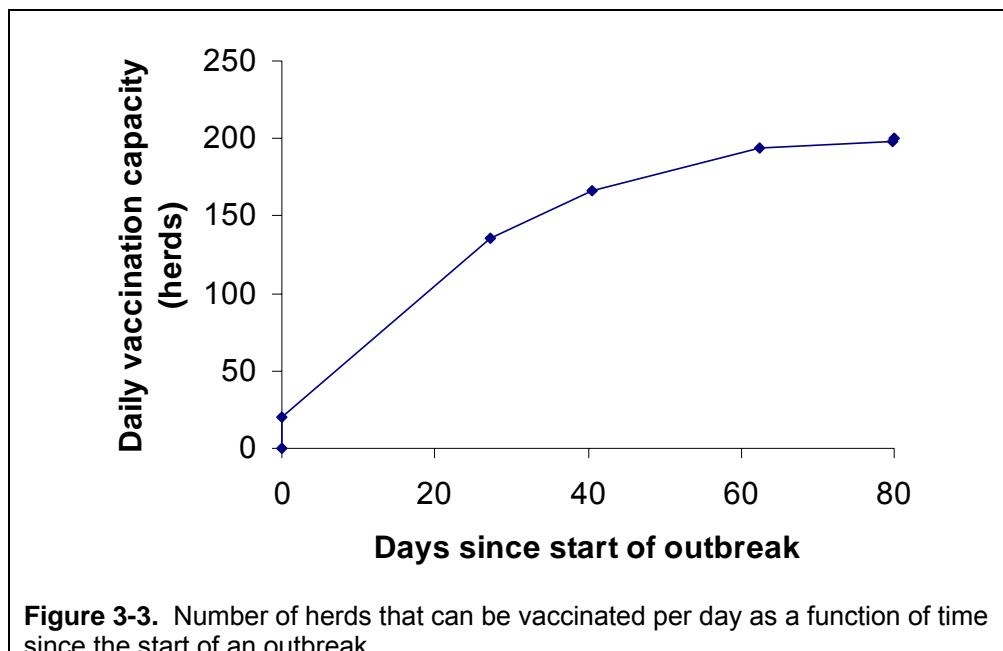


Figure 3-2. Output of 1000 iterations of a stochastic model estimating the amount of fabric needed to make a pair of pants.

NAADSM is a stochastic model. Variability is incorporated in the model using *distributions* and *relationships*. A *distribution* is used when the variability in a parameter is assumed to be random. The distribution describes the range of values a variable can take, and how likely those values are to be selected. Model parameters described as distributions include the length of infectious period and the distance that animals are likely to be transported. A *relationship* is used when the variability in one parameter is due to another factor. A relationship describes one variable as a function of another (Figure 3-3). Model parameters described as relationships include the probability of detecting an infectious herd (which is a function of time since the herd was infected), and number of herds that can be depopulated per day (a function of time since the outbreak was detected).



4. Installing and registering *NAADSM*

4.1. Installing *NAADSM*

Installation of *NAADSM* should be straight-forward, and takes only one or two steps. First, obtain the most recent full installation package for *NAADSM*. (The *NAADSM* website, <<http://www.naadsm.org>>, is probably the most convenient source for this installation package, but it is also available directly from members of the *NAADSM* development team: please see Appendix G for contact information.) The full installation package will have a file name similar to **NAADSMsetup-3_0_x.exe**, where the more recent full installation version number is given in place of “x”. Run this setup package, *e.g.*, by double-clicking on its icon. A screen like the one shown in Figure 4-1 should be displayed.

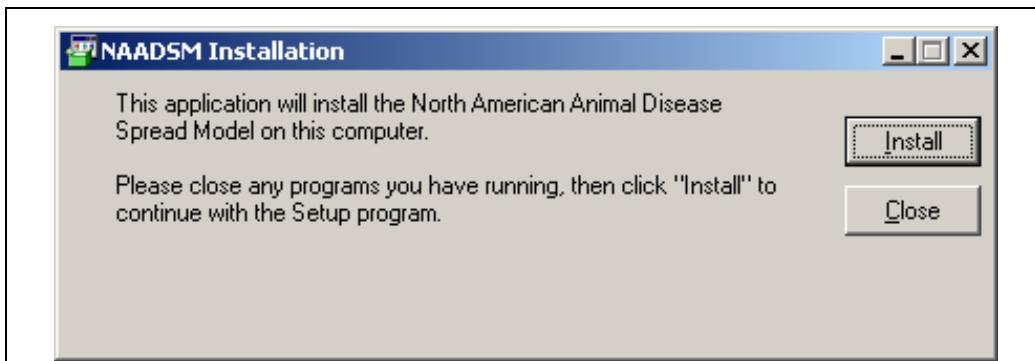


Figure 4-1. The first installation message displayed by the *NAADSM* setup package.

Click **Install** to begin the installation process. On the following screen, click **I agree** to accept the license agreement. The *NAADSM* application and related library files will be installed to the folder **C:\Program Files\NAADSM** on your computer. If installation proceeds properly, the message shown in Figure 4-2 will be shown. (If an installation error occurs, please consult with one of the *NAADSM* developers, listed in Appendix G).

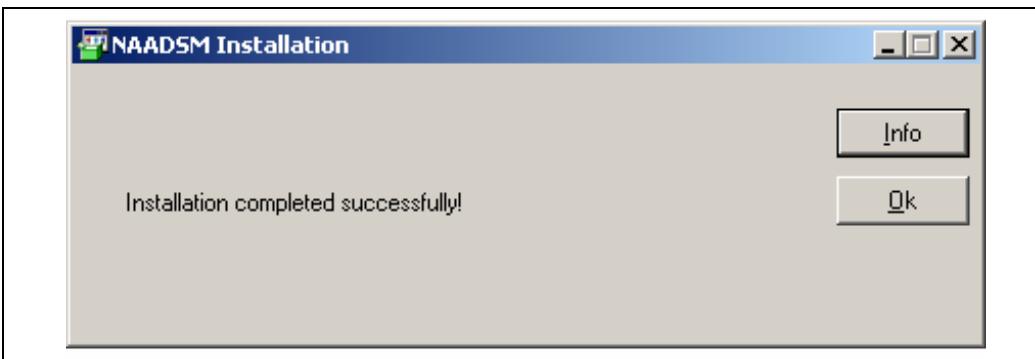


Figure 4-2. The final installation message displayed by the *NAADSM* setup package.

4. Installing and registering NAADSM

Once a full installation is complete, check for the availability of an *NAADSM* update package. Update packages will have file names similar to **NAADSMUpdator-3_0_x.exe**, and are used to update the one or two files that have changed since the last full installation version. If there are several update packages newer than your full installation, it is only necessary to obtain and run the newest update package. If successfully run, the update package will display messages as shown in Figure 4-3. If an update error occurs, please consult with one of the *NAADSM* developers.

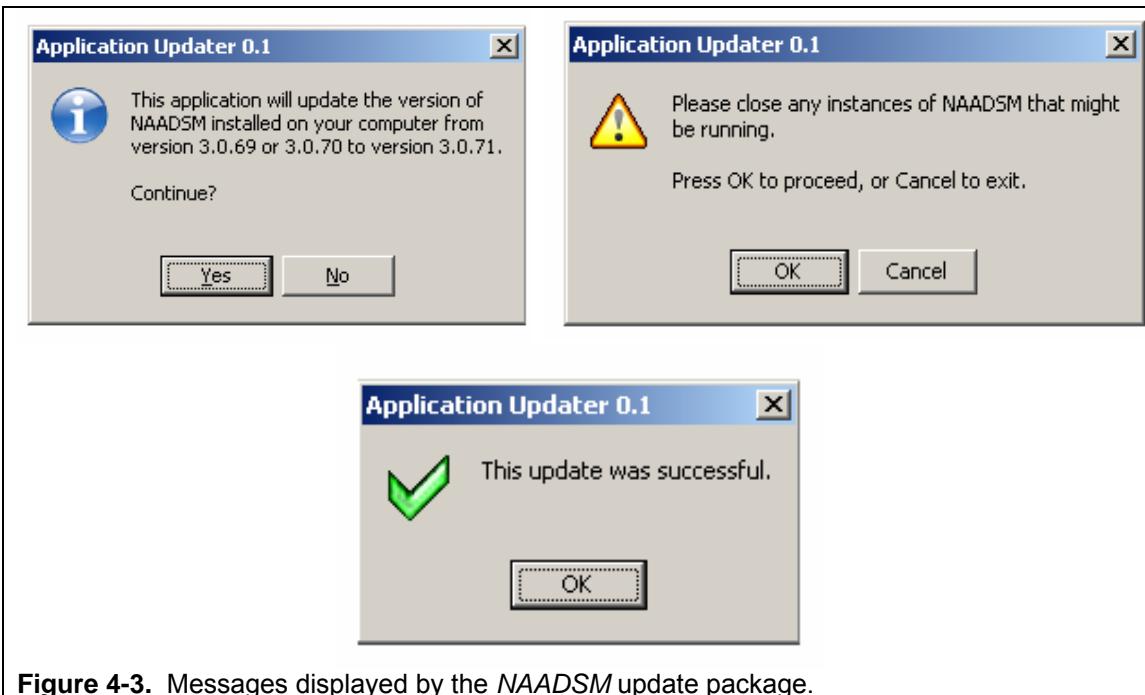


Figure 4-3. Messages displayed by the *NAADSM* update package.

When installation is complete, an *NAADSM* menu item will be added to your *Windows Start* menu, as shown in Figure 4-4. Select this item to launch *NAADSM*.



Figure 4-4. The *NAADSM* icon created on the *Windows Start* menu (the arrangement of your menu will vary).

4.2. Registering *NAADSM*

While registration is not required, users are encouraged to register their copies of *NAADSM* through the *NAADSM* website at <<http://www.naasdm.org>>. Registration is quick and simple, and provides the development team with the means to keep users up to date regarding bug fixes, the introduction of new features and releases.

In addition to the registration form, the website provides installation packages, source code, user documentation (including the latest version of this manual and any errata), developer's notes, and other resources for users of the *North American Animal Disease Spread Model*.

5. Running your first simulation

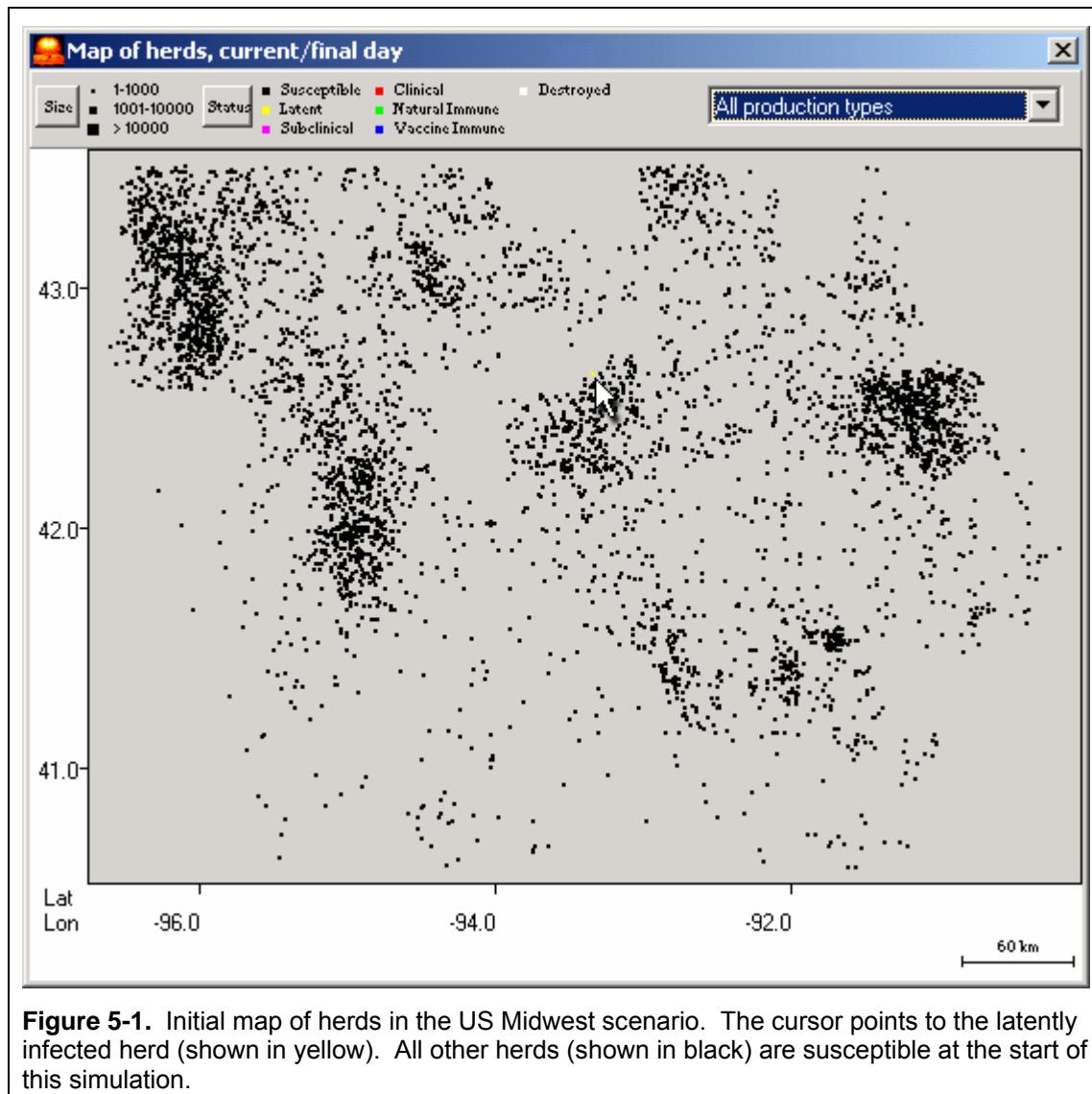
NAADSM is supplied with a sample scenario, which includes herd population data and preset parameters representative of the American Midwest. To view and run this sample scenario, start the *NAADSM* application, go to the **File** menu, choose **New scenario...**, select **US Midwest scenario**, and select a folder in which to save the new scenario when prompted.

5.1. The starting population

Once you have created and saved the scenario, you will see a map of herds, with latitude and longitude coordinates, on your screen (Figure 5-1). Note that nearly all dots on the map are black: these indicate herds that are susceptible to disease. Looking carefully, you will see a single yellow dot on the map. This represents a latently infected herd, which will be the source of infection for the simulated outbreak.

The options for setting up and viewing an initial population are discussed in Section 8.4.

5. Running your first simulation



5.2. The disease scenario

To review the parameters for this scenario, go to the **Scenario parameters** menu and choose **Start setup**. NAADSM parameters are specified with a series of “wizard” screens. By clicking on the **Next >** button on any of these screens, you may review these parameters in sequence. Alternatively, you may select one of the other items from uppermost section of the **Scenario parameters** menu to skip directly to the parameters of interest (Figure 5-2). Click the **Finish** button on any wizard screen when you are done reviewing the model parameters.

These wizard screens are discussed in greater detail in Section 8.

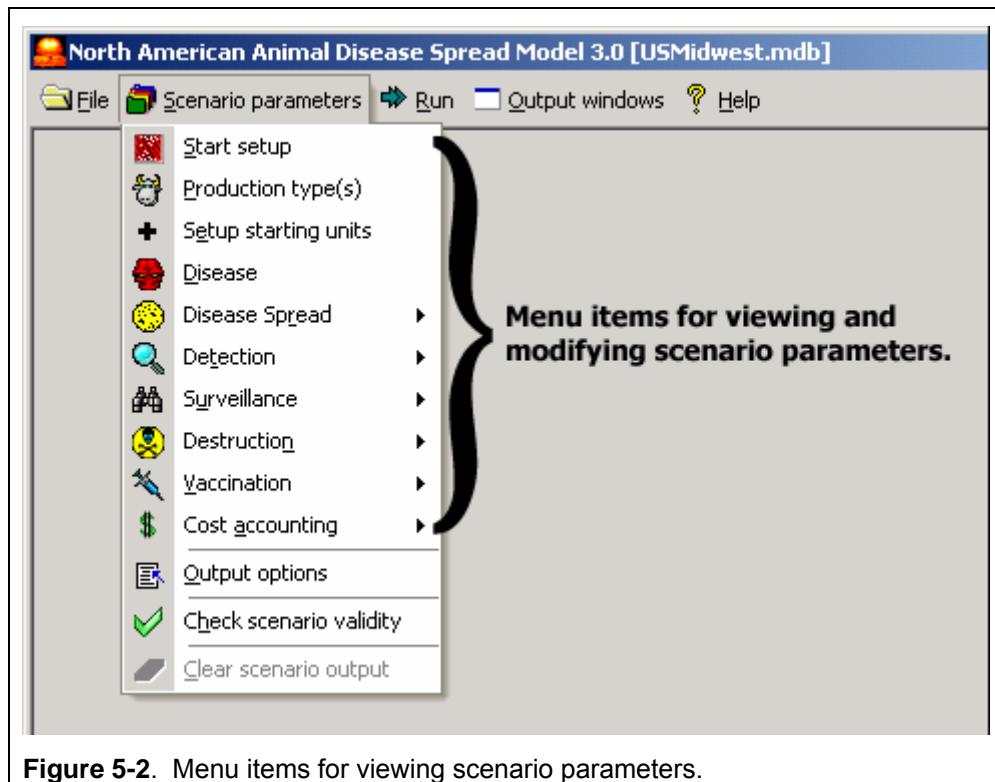


Figure 5-2. Menu items for viewing scenario parameters.

After selecting **Start setup**, the **Start setup** screen will be displayed. In the box labeled **Number of Iterations**, type “1”. In the box labeled **Random number generator seed**, make sure that the option **Generate seed automatically** is selected. Click **Finish** to save your changes and exit the input wizard screen.

5.3. Launching the simulation

To start the simulation, go to the **Run** menu, and select **Start and run until end of outbreak(s)**. *NAADSM* will now run the simulation. On the map, the dots representing herds will change colors to indicate their disease states (Figure 5-3). Clicking on the **Status** button on the map will change the legend to show disease status from an emergency personnel standpoint. At the bottom of your screen, a counter tracks the iteration number, and the day of the outbreak. A single run (iteration) of this scenario can take some time, depending on population size and computer speed, to reach completion. If you want to stop the iteration prematurely, click on the **Stop** button in the lower right corner of your screen. *NAADSM* will finish the current simulation “day” and then stop.

The options available for launching a simulation and specifying when each iteration should end are discussed in Section 9.

5. Running your first simulation

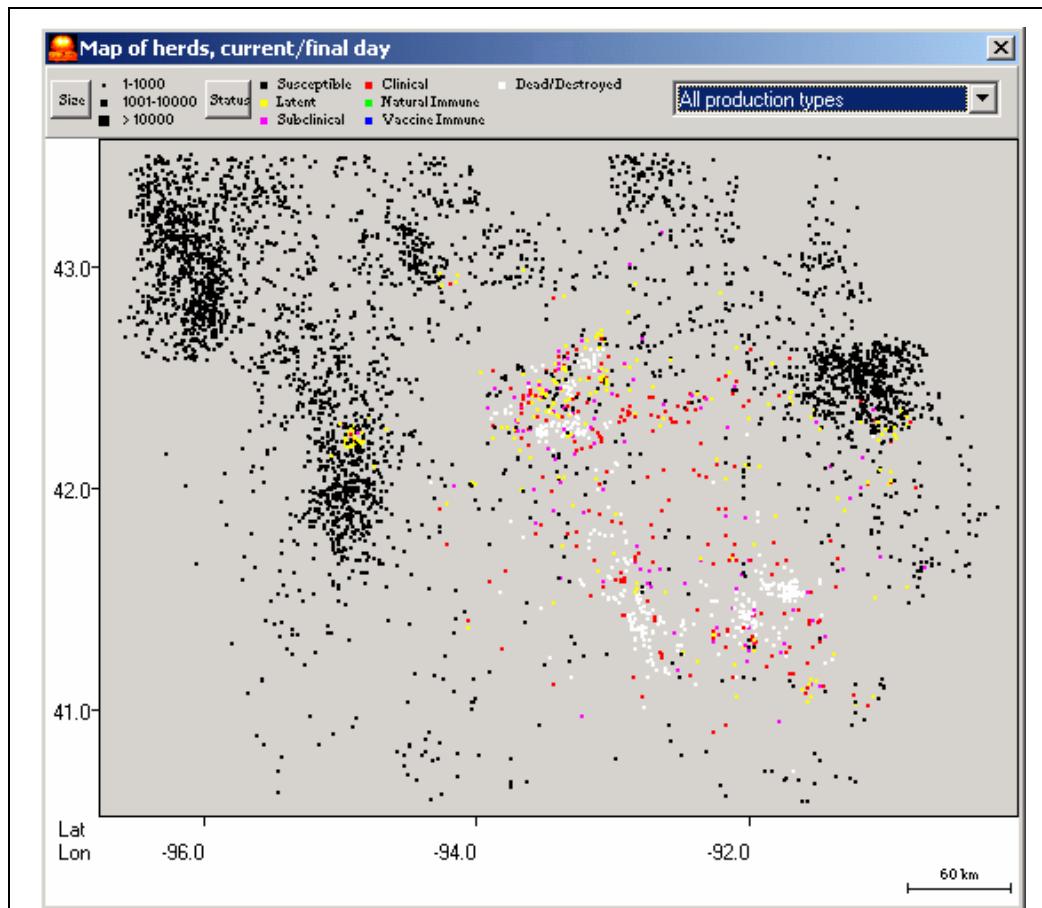


Figure 5-3. Map showing herds and their disease states for an iteration in progress. Points on the map will change color as herds are infected and transition through the defined disease stages. *Color codes:* black: susceptible herds; yellow: latently infected herds; pink: subclinically infected herds; red: infected herds showing clinical signs; white: herds that have been destroyed as a control measure.

5.4. Viewing simulation output

After the simulation stops (either on its own or as a result of using the **Stop** button), go to the **Output windows** menu and choose **Summary of 1 iteration**. This window shows the number of diseased herds and animals that were detected, and the numbers herds and animals destroyed or vaccinated. Additionally, you can see the epidemic curve apparent to emergency management personnel, as well as the “true” actual epidemic curve, which includes undetected diseased herds.

To see how the numbers of herds in the various disease stages changed over the course of the iteration, select **Output windows → Daily unit status for 1 iteration**. A graph similar to the one shown in Figure 5-4 will be displayed. Note in Figure 5-4 how the number of susceptible herds

5. Running your first simulation

declines as the numbers of herds in the various disease stages and the number of destroyed herds increase over time.

NAADSM produces output for single iterations, as well as summary output generated from multiple iterations.

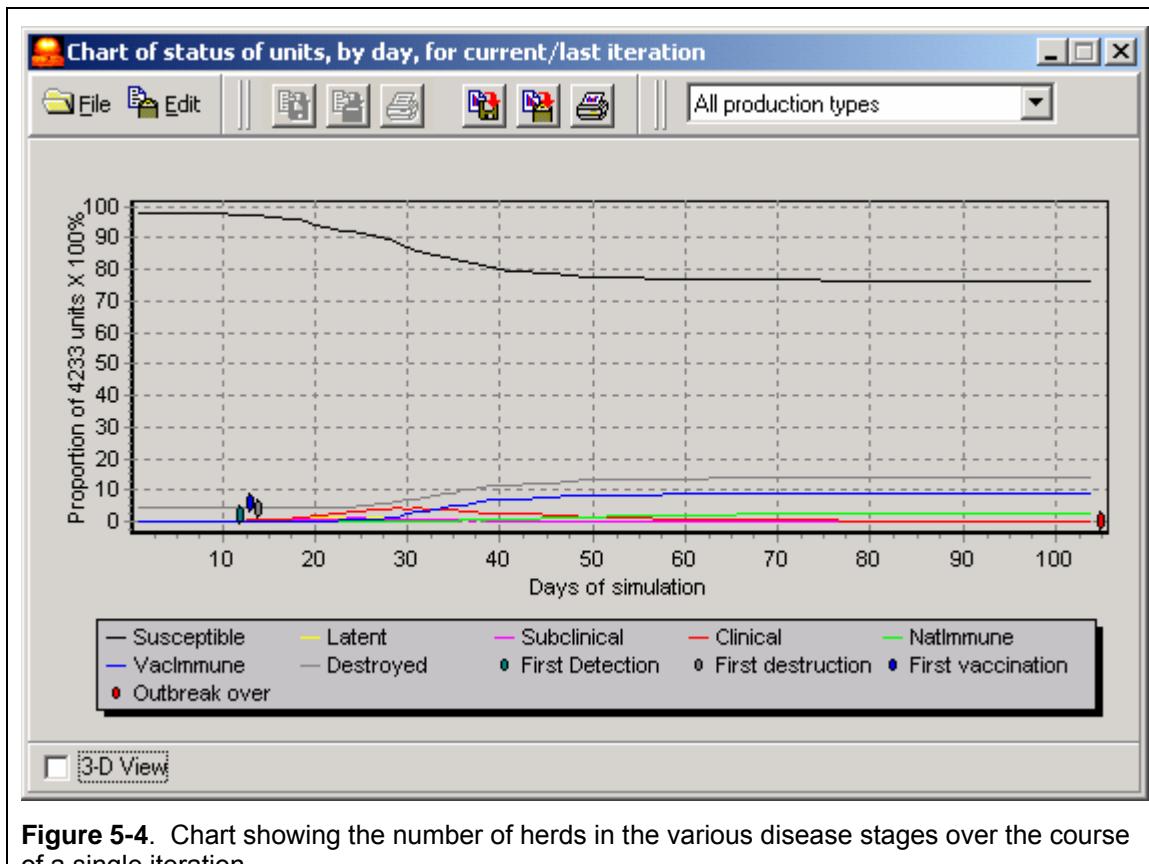


Figure 5-4. Chart showing the number of herds in the various disease stages over the course of a single iteration.

Several other selections are available on the **Output windows** menu. These items, which will be discussed in Section 10, are more useful when many iterations of a single scenario have been completed. The following sections of this guide will describe the kinds of parameters used to create or modify an *NAADSM* scenario, and will show how these parameters may be set by the user.

6. Types of input parameters

NAADSM accepts a multitude of parameters for scenario creation. Many of these parameters are required; others are optional, depending on the particular details associated with each possible scenario. This section provides an overview of the types of parameters required to set up or alter an *NAADSM* scenario. All input parameters and their uses are described in detail in Appendix A.

6.1. Basic types

There are six basic types of parameters used throughout *NAADSM*: yes/no values, integer values, floating point numbers, probabilities, probability density functions, and relational functions. The user interface provides hints as to which type of value is required for each parameter, and will not allow users to enter the wrong type.

These basic types are the building blocks of more complex elements of a complete *NAADSM* scenario, which will be examined in Section 6.2.

6.1.1. Yes/no values

Many scenario settings in *NAADSM* may be switched on or off: for example, a scenario may be set up with or without vaccination as a control measure. Yes/no values (sometimes called true/false or boolean values) like this are usually set via check boxes: a check in the box indicates that the option will be enabled.

6.1.2. Integer values

Various parameters require whole number values: examples include the delay in whole days before a destruction campaign is initiated, and the number of diseased herds that must be detected before a vaccination program begins. The user interface will not allow negative or floating point numbers in fields that require integers.

6.1.3. Floating point numbers

Floating point (sometimes called real) numbers are integer or non-integer values such as 1, 2.6, 9.34, etc. Several input parameters, such as the mean rate of herd shipments, are floating point numbers. Latitude and longitude are expressed as floating point numbers as well.

6.1.4. Probabilities

Quite a few model parameters are specified as probabilities, or floating point values between 0 and 1. An example is the probability of infection given exposure to an infected herd.

6. Types of input parameters

The user interface will not allow users to enter values outside of the range between 0 and 1 when a probability is required.

6.1.5. Probability density functions

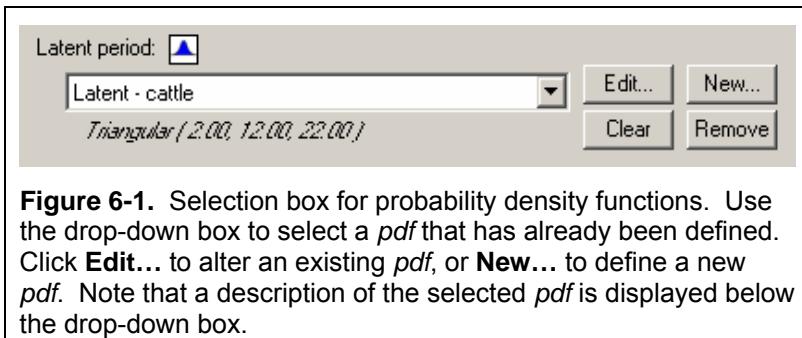
Probability density functions (*pdfs*) are distributions of values representative of the natural range of possible values for some parameter. The length of time that an infected herd is clinically infectious is one of many *pdf* parameters required by NAADSM. Values are drawn stochastically from these distributions as a simulation runs. Section 3 provides additional background information regarding the use of distributions in stochastic models.

6.1.5.1. Creating and editing *pdfs*

A new *pdf* may be created by entering parameters for a known mathematical probability function (normal, uniform, lognormal, triangular, *etc.*), by importing data, or by using the provided tools to create your own shape. Appendix D lists the *pdf* types supported by NAADSM.

6.1.5.1.1. The *pdf* selection box

Each *pdf* parameter in NAADSM is associated with a selection box like the one shown in Figure 6-1. The selection box shows a list of *pdfs* defined in each scenario, and provides buttons for editing, creating, and removing *pdfs*.



6.1.5.1.2. The *pdf* editor window

Clicking on either the **Edit...** or **New...** button will bring up a *pdf* editor window, similar to the one shown in Figure 6-2. The left side of the window shows the type of the selected *pdf* (*e.g.* triangular, normal, uniform, piecewise, *etc.*), the parameters required to define the selected type, and the values of those parameters. The right side of the window graphically depicts the selected *pdf*. Changing the parameters will cause the graphical depiction to change as well (Figure 6-3). If the parameters are changed to inappropriate values, an error message is displayed (Figure 6-4). The user must address the error problem in order to save the changes to the newly altered or created *pdf*.

6. Types of input parameters

Every new *pdf* must be assigned a unique name. Enter the name in the **Function name** text box in the upper right corner of the *pdf* editor window (Figure 6-2). If you are editing an existing *pdf*, you may rename it by changing the name entered in the **Function name** text box.

Once you have created or edited a *pdf*, you need to save it to apply the changes that you have made. This may be accomplished in either of two ways: you can use the **File → Save changes to function and exit** menu command or click on the  button.

If you wish to exit the *pdf* editor window without applying the changes you have made, use the **File → Exit without saving changes** menu command or click on the  button.

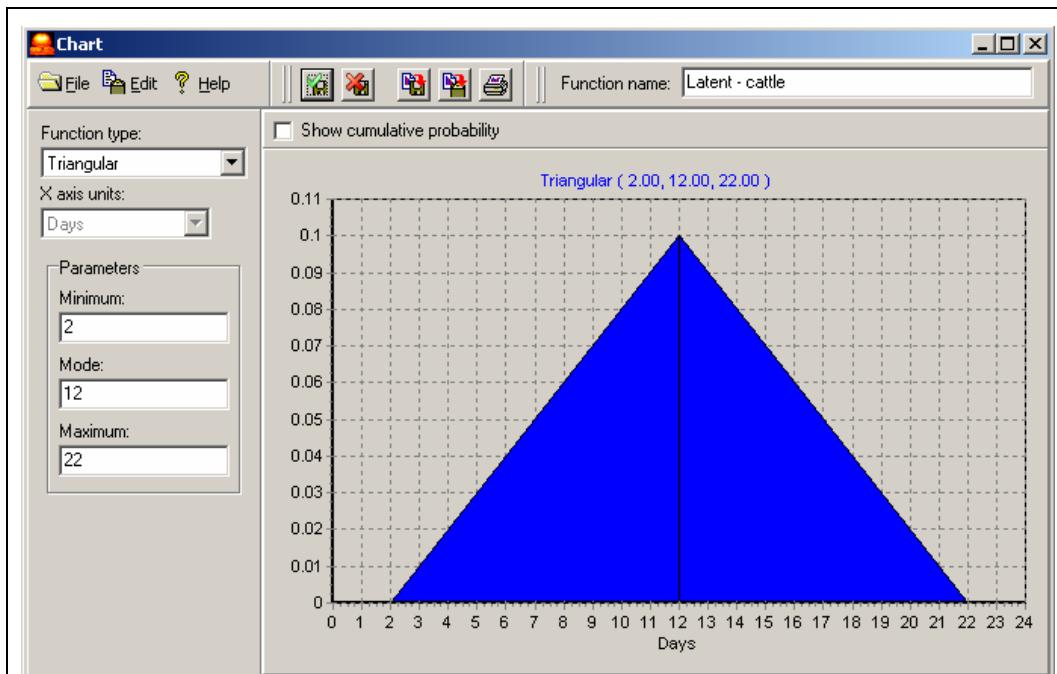


Figure 6-2. The *pdf* editor window. The type and parameters for the selected *pdf* are shown on the left. A graphical preview of the *pdf* is shown on the right. Note the name assigned to this particular *pdf*, shown in the upper right corner.

6. Types of input parameters

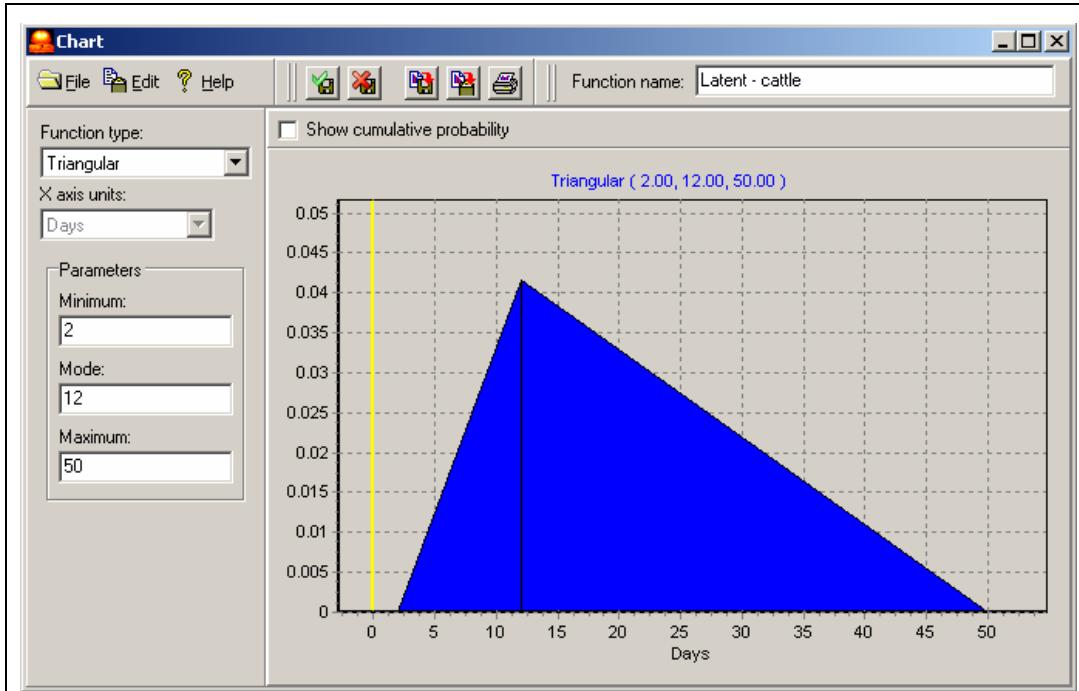


Figure 6-3. The parameters of the triangular *pdf* have changed. The graphical preview has been updated to correspond to the new parameters.

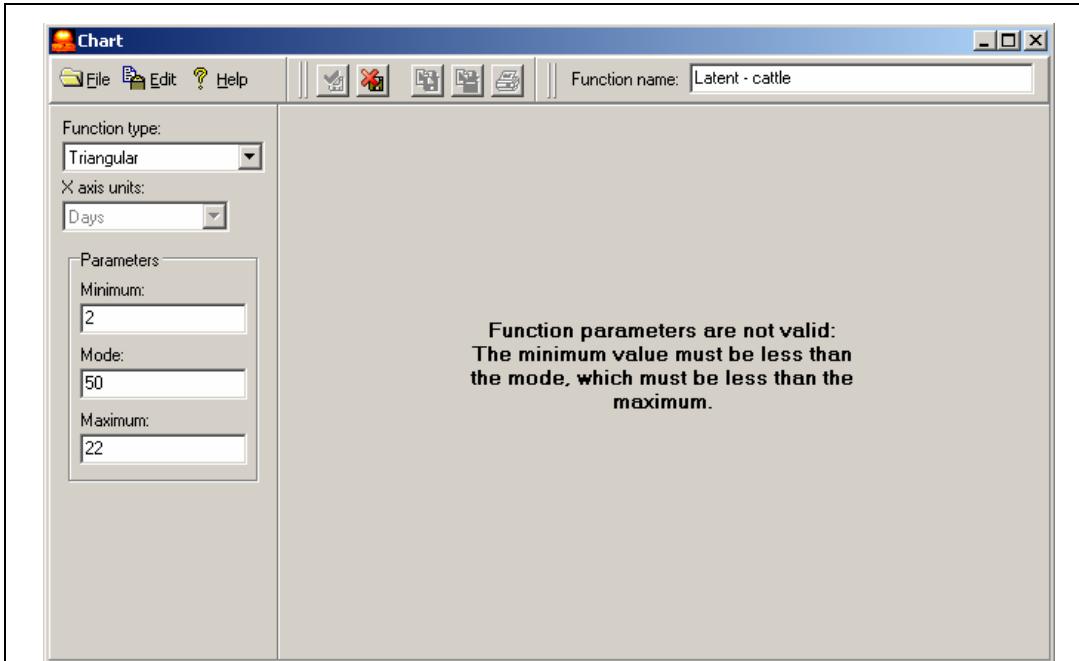


Figure 6-4. There is an error in the specified parameters. The source of the error is shown instead of the graphical preview.

6.1.5.1.3. Creating and editing piecewise *pdfs*

Piecewise *pdfs* can reflect nearly any shape that the user wishes, provided that the start and end points have an *y* value of 0. Figure 6-5 shows a piecewise *pdf* in the *pdf* editor window.

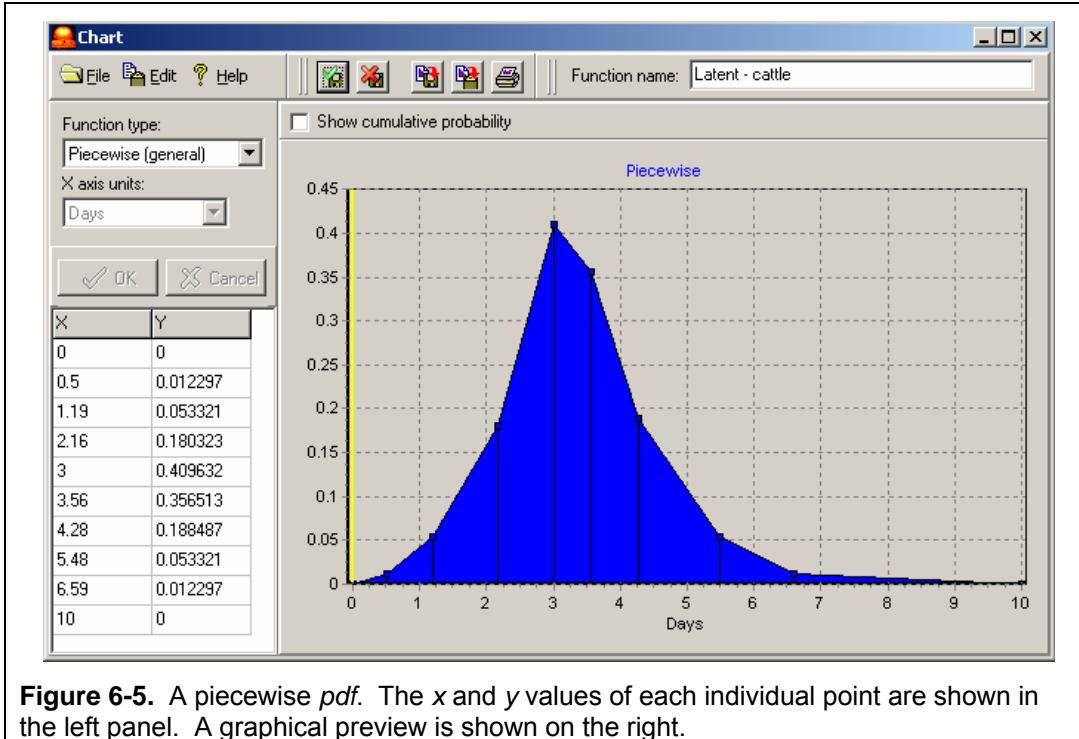


Figure 6-5. A piecewise *pdf*. The *x* and *y* values of each individual point are shown in the left panel. A graphical preview is shown on the right.

To create piecewise *pdf*, select **Piecewise (general)** from the list of function types. Three points are created automatically (Figure 6-6). You can alter the shape by moving or deleting existing points, or by adding new points. *NAADSM* will standardize the *pdf* so that the area under the curve is always 1. If you are editing an existing piecewise *pdf*, **Piecewise (general)** will already be selected, and the currently defined points will be displayed.

6. Types of input parameters

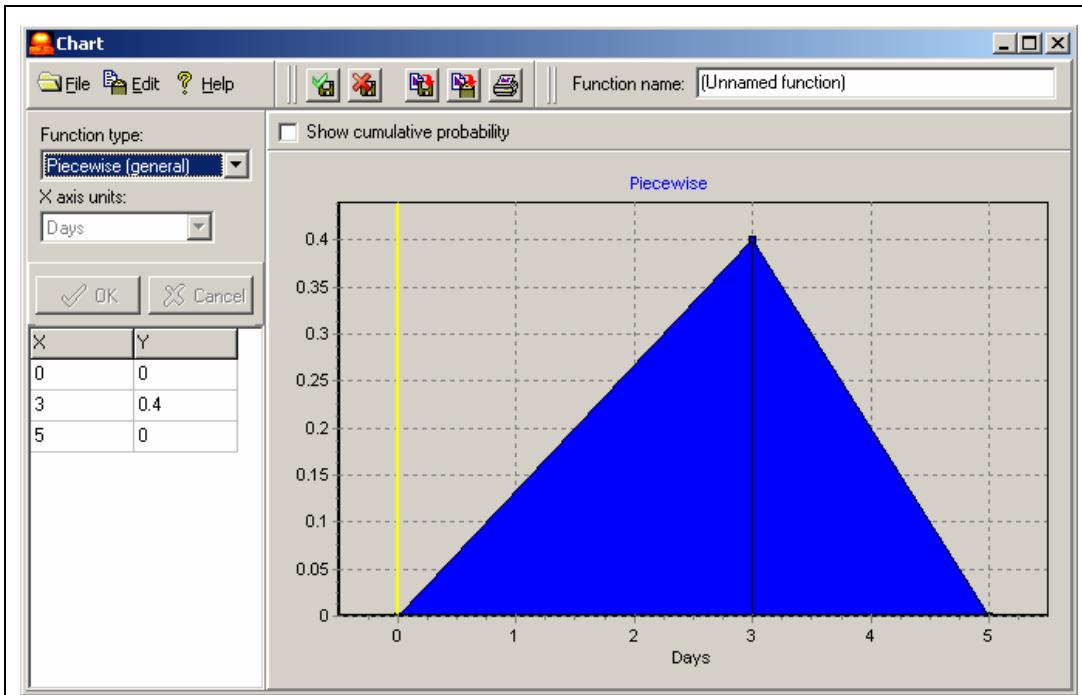


Figure 6-6. A new piecewise *pdf*. Three points are created by default. The values of these points may be changed directly in the left panel. Points may be added, removed, or dragged to new positions in the right panel. Note that the newly created function is unnamed. The user should assign a unique name to each new *pdf*.

Adding a new point

Move your pointer to a location on the chart where you would like an additional point. Right-click, then select **Add point** (Figure 6-7).

Moving an existing point

Left click on the point you wish to move. While holding the mouse button down, drag the point to its new position (Figure 6-8).

6. Types of input parameters

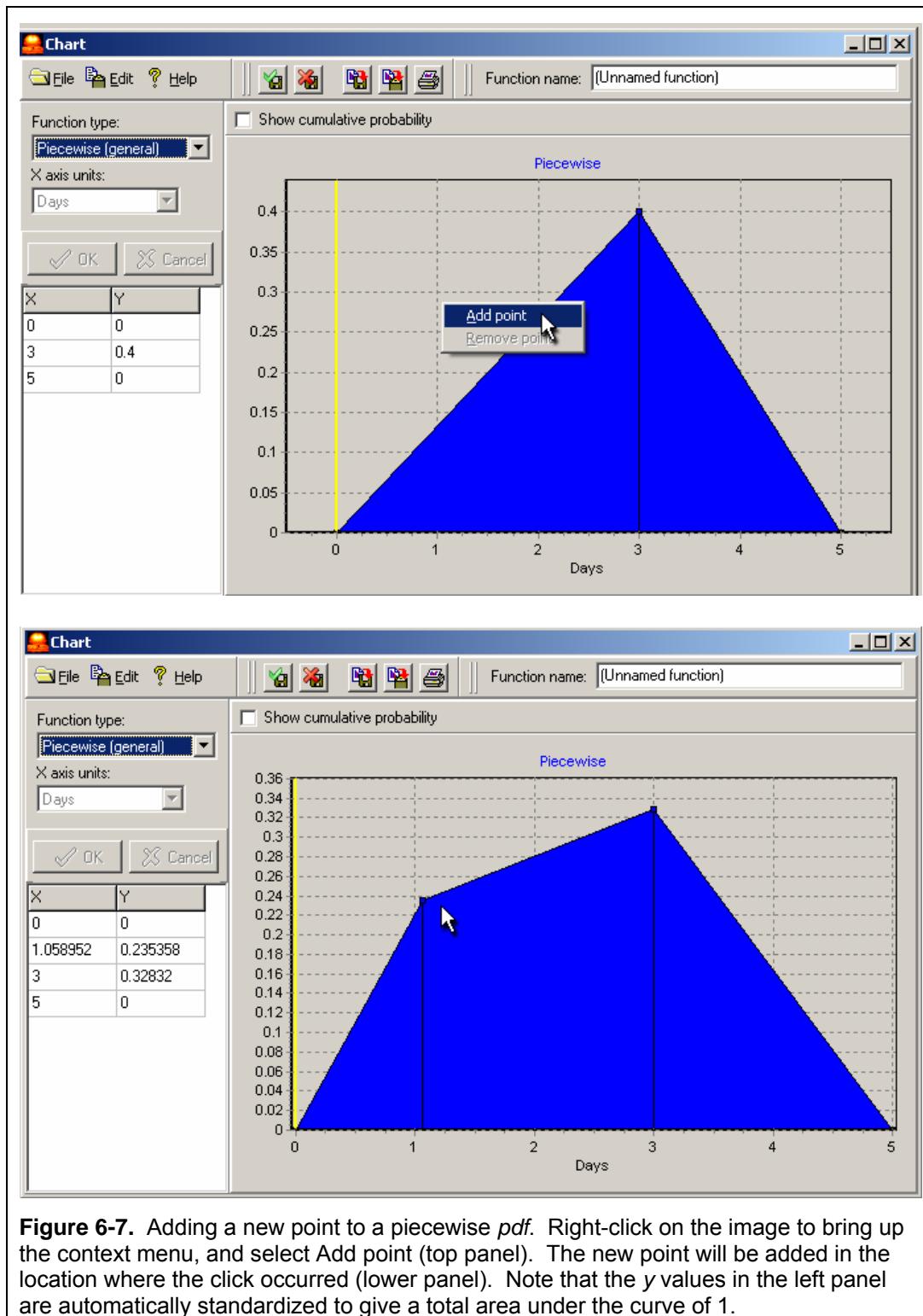


Figure 6-7. Adding a new point to a piecewise pdf. Right-click on the image to bring up the context menu, and select Add point (top panel). The new point will be added in the location where the click occurred (lower panel). Note that the y values in the left panel are automatically standardized to give a total area under the curve of 1.

6. Types of input parameters

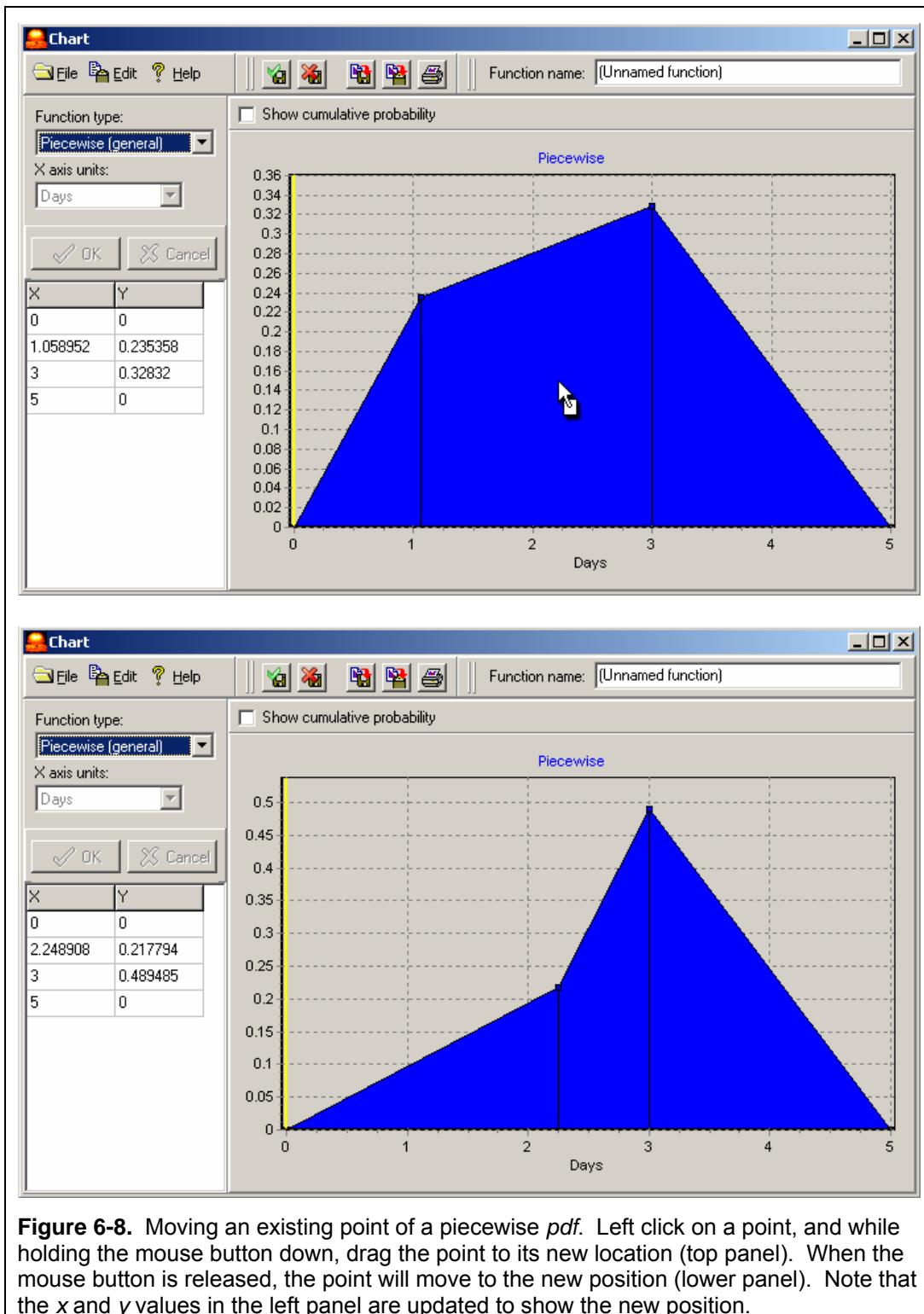


Figure 6-8. Moving an existing point of a piecewise pdf. Left click on a point, and while holding the mouse button down, drag the point to its new location (top panel). When the mouse button is released, the point will move to the new position (lower panel). Note that the x and y values in the left panel are updated to show the new position.

Changing the *x* axis

To increase the maximum value of the *x* axis, click on the rightmost point of your distribution, and drag it towards the right. When you lift your finger from the mouse button, the *x* axis will be rescaled. Similarly, to decrease the maximum value of the *x* axis, click on the rightmost point of your distribution, and drag it to the left. The same techniques can be used to alter the minimum value visible on the *x* axis.

The *x* axis can be altered with more precision by directly editing the point values, as described below.

Deleting an existing point

Right-click on the point you wish to delete, and select **Remove point**.

Directly editing point values

The *x* and *y* coordinates of each point (displayed in the left panel of the *pdf* editor window) may be changed directly by editing the values shown in the left panel. Simply enter the desired values, and click on the **OK** button to apply your changes.

There are several things to note when point values are edited directly: first, the leftmost and rightmost points must always have a *y* value of 0: attempts to assign these values to something other than 0 will not be successful. Second, the *y* values are automatically rescaled to give a graph with a total area of 1. If you change a *y* value, it will immediately be rescaled once you apply the changes so that it is appropriately proportional to the other values in the graph. If you need to adjust several *y* values, make all of your changes before clicking on the **OK** button to prevent immediate rescaling of each *y* value as you enter it.

6.1.5.1.4. Importing a piecewise *pdf*

If you have defined values for your piecewise *pdf* in another application (for example, in a spreadsheet or a statistical package), you may import the point values directly into the *pdf* editor window. Data should consist of two columns (*x* and *y* values) and should be in a comma-separated (*.csv) file. Values along the *x* axis should be in units appropriate for the *pdf* (for example, the *x* axis should be in days for a *pdf* describing the length of the latent period). Values along the *y* axis are in units of probability density such that the area under the curve of the entire function equals 1. *NAADSM* will adjust *y* axis values as needed to ensure that the area under the curve equals 1. Data can be imported from a file or from the clipboard. Figure 6-9 shows sample data suitable for importing into the *pdf* editor window. More details about the proper file format for this and other files used by *NAADSM* is given in Appendix C.

To import points from a file, use the menu command **File → Import points from file...** in the *pdf* editor window. To import points from the clipboard, use the menu command **Edit → Import points from clipboard**. Do not forget to assign a name to your new *pdf* in the **Function name** text box in upper right corner of the *pdf* editor window. Save the new *pdf* as usual, using either the **File → Save changes to function and exit** command or the  button.

6. Types of input parameters

x,	y
0,	0
0.5,	0.012198
1.19,	0.052892
2.16,	0.178874
2.72,	0.349649
3,	0.406341
3.56,	0.353648
4.28,	0.186973
5.48,	0.052892
6.59,	0.012198
10,	0

Figure 6-9. A properly formatted *.csv file for importing a piecewise *pdf*. Note that the first row is a header row, which identifies the x and y columns. (Columns shown here are evenly spaced to make them easier to read: uniform spacing is not required in the file to be imported.)

6.1.5.2. Exporting a piecewise *pdf*

You may want to export the coordinates a *pdf*. Exporting coordinates may be useful if you want to re-create the *pdf* in other software (*e.g.* in a statistical package), or if you want to import the *pdf* for use in another variable (for example, you may want to use the same *pdf* for both latent and infectious subclinical periods). You can export the points of a piecewise *pdf* to a file, or to the clipboard for pasting into another application.

To export to a file, use the **File → Export points to file...** menu command, and enter the name of a comma-delimited (*.csv) file to save. To export (copy) points to the clipboard, simply use the **Edit → Copy points to clipboard** menu command.

6.1.5.3. Saving, copying, or printing *pdf* images

The graphical image of any *pdf* may be saved to a file, copied to the clipboard, or printed.

Images are saved in the Windows metafile (*.wmf) format. To save a *pdf* image, use the menu command **File → Save chart image...** in the *pdf* editor window and enter a valid file name. Alternatively, you may use the  button.

To copy a *pdf* image to the clipboard, use the menu command **Edit → Copy chart image to clipboard** or use the  button. The copied image is suitable for pasting into *Microsoft Word*, *PowerPoint*, and many other applications.

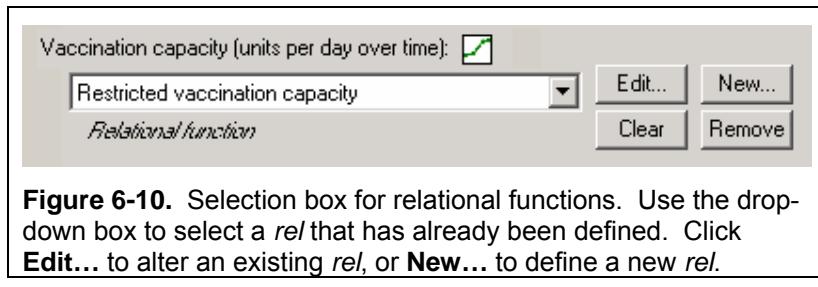
To print a *pdf* image, use the menu command **File → Print chart image...** and select a printer. Alternatively, you may use the  button to print the image directly to the currently selected system printer.

6.1.6. Relational functions

Relationships or relational functions (*rels*) are used in situations where one variable is a function of another: an example of a *rel* in *NAADSM* is the probability of detecting an infectious herd, which is a function of time since that herd itself was infected (see Section 3 and Figure 3-3 for a more detailed discussion of *rels*). In *NAADSM*, *rels* are created, edited, and handled in ways very similar to those described for *pdfs* (see Section 6.1.5).

6.1.6.1. The *rel* selection box

Each *rel* parameter in *NAADSM* is associated with a selection box like the one shown in Figure 6-10. The selection box shows a list of *rels* defined in each scenario, and provides buttons for editing, creating, and removing *rels*.



6.1.6.2. The *rel* editor window

Clicking on either the **Edit...** or **New...** button will bring up a *rel* editor window, similar to the one shown in Figure 6-11.

The right side of the window graphically depicts the selected *rel*. Changing the parameters will cause the graphical depiction to change as well. If the parameters are changed to inappropriate values, an error message is displayed. The user must address the error problem in order to save the changes to the newly altered or created *rel*.

Every new *rel* must be assigned a unique name. Enter the name in the **Function name** text box in the upper right corner of the *rel* editor window (Figure 6-11). If you are editing an existing *rel*, you may rename it by changing the name entered in the **Function name** text box.

Once you have created or edited a *rel*, you need to save it to apply the changes that you have made. This may be accomplished in either of two ways: you can use the **File → Save changes to function and exit** menu command or click on the button.

If you wish to exit the *rel* editor window without applying the changes you have made, use the **File → Exit without saving changes** menu command or click on the button.

6. Types of input parameters

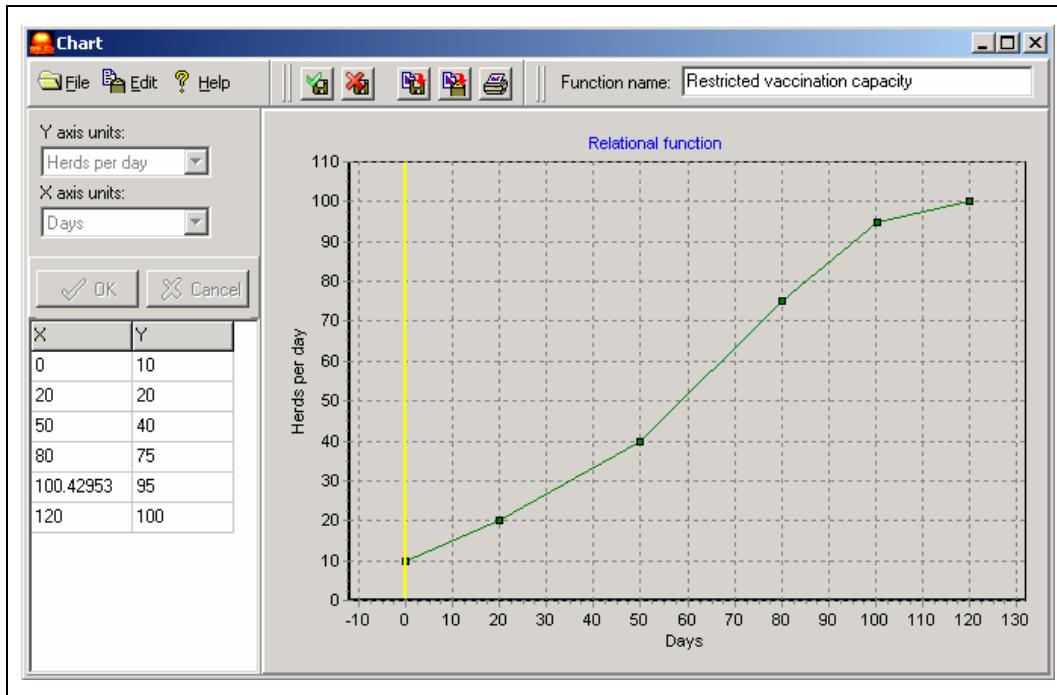


Figure 6-11. The *rel* editor window. The values of each point are displayed in the left panel. A graphical preview of the *rel* is shown on the right. Note the name assigned to this particular *rel*, shown in the upper right corner.

6.1.6.3. Creating and editing *rel* functions

Relational functions can reflect nearly any shape that the user wishes. When a new *rel* function is created, three points are created automatically (Figure 6-12). If you are editing an existing *rel* function, currently defined points will be displayed. You can alter the shape by moving or deleting existing points, or by adding new points.

Adding a new point

New points are added to *rels* in the same way that they are added to piecewise *pdfs* (see Section 6.1.5.1.3). Move your pointer to a location on the chart where you would like an additional point. Right-click, then select **Add point**.

Moving an existing point

Left click on the point you wish to move. While holding the mouse button down, drag the point to its new position (see Section 6.1.5.1.3).

Changing the axes

To increase the maximum value of either axis, click on the uppermost (for the *y* axis) or the rightmost (for the *x* axis) point of your function, and drag it toward the top (for the *y* axis) or the right (for the *x* axis). When you lift your finger from the mouse button, the axis will be

rescaled. Similarly, to decrease the maximum values, drag points downward or to the left. The same techniques can be used to alter the minimum value visible on either axis.

The axes can be altered with more precision by directly editing the point values, as described below.

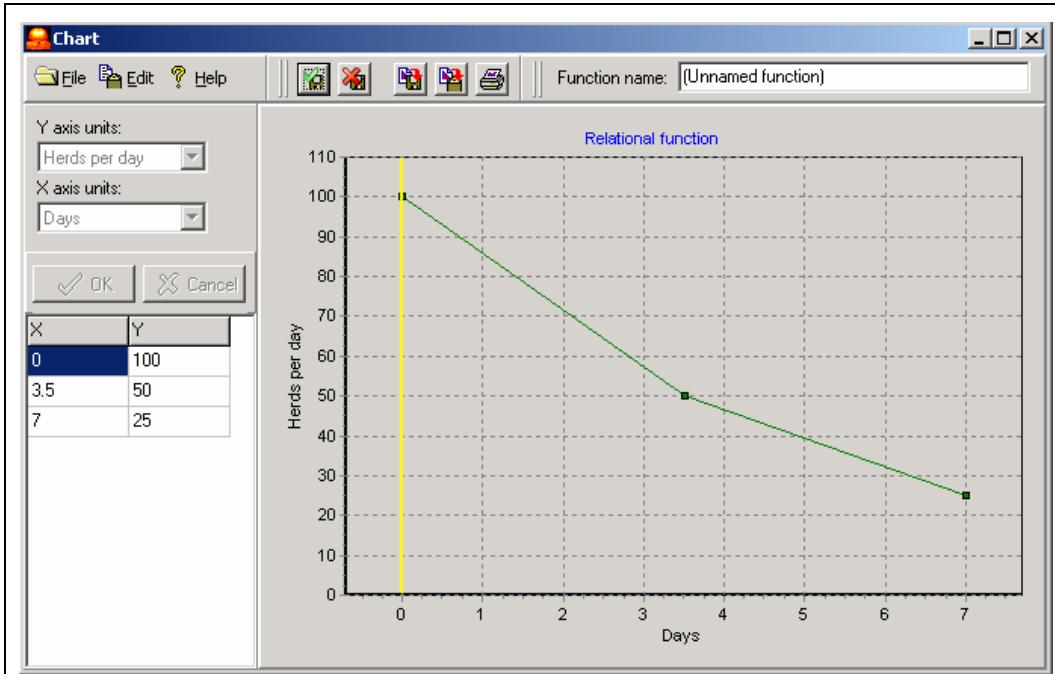


Figure 6-12. A new relational function. Three points are created by default. The values of these points may be changed directly in the left panel. Points may be added, removed, or dragged to new positions in the right panel. Note that the newly created function is unnamed. The user should assign a unique name to each new *rel*.

Deleting an existing point

Right-click on the point you wish to delete, and select **Remove point**.

Directly editing point values

The *x* and *y* coordinates of each point (displayed in the left panel of the *rel* editor window) may be changed directly by editing the values shown in the left panel. Simply enter the desired values, and click on the **OK** button to apply your changes.

6.1.6.4. Importing a *rel* function

If you have defined values for your relational function in another application (for example, in a spreadsheet or a statistical package), you may import the point values directly into the *rel* editor window. Data should consist of two columns (*x* and *y* values) and should be in a comma-separated (*.csv) file. Values along the *x* and *y* axes should be in units appropriate for the *rel*. Data can be imported from a file or from the clipboard. Figure 6-13 shows sample data

6. Types of input parameters

suitable for importing into the *rel* editor window. More detail about the proper file format for this and other *.csv files used by NAADSM is given in Appendix C.

To import points from a file, use the menu command **File → Import points from file...** in the *rel* editor window. To import points from the clipboard, use the menu command **Edit → Import points from clipboard**. Do not forget to assign a name to your new *rel* in the **Function name** text box in upper right corner of the *rel* editor window. Save the new *rel* as usual, using either the **File → Save changes to function and exit** command or the  button.

x,	y
0,	10
20,	20
50,	40
80,	75
100,	95
120,	100

Figure 6-13. A properly formatted *.csv file for importing a relational function. Note that the first row is a header row, which identifies the x and y columns. (Columns shown here are evenly spaced to make them easier to read: uniform spacing is not required in the file to be imported.)

6.1.6.5. Exporting a *rel* function

You may want to export the coordinates a *rel*. Exporting coordinates may be useful if you want to re-create the *rel* in other software (*e.g.* in *Microsoft Excel* or a dedicated statistical package), or if you want to import the *rel* for use in another variable. You can export the points of a piecewise *rel* to a file, or to the clipboard for pasting into another application.

To export to a file, use the **File → Export points to file...** menu command, and enter the name of a comma-delimited (*.csv) file to save. To export (copy) points to the clipboard, simply use the **Edit → Copy points to clipboard** menu command.

6.1.6.6. Saving, copying, or printing images of *rel* functions

The graphical image of any *rel* may be saved to a file, copied to the clipboard, or printed.

Images are saved in the *Windows* metafile (*.wmf) format. To save a *rel* image, use the menu command **File → Save chart image...** in the *rel* editor window and enter a valid file name. Alternatively, you may use the  button.

To copy a *rel* image to the clipboard, use the menu command **Edit → Copy chart image to clipboard** or use the  button. The copied image is suitable for pasting into *Microsoft Word*, *PowerPoint*, and many other applications.

To print a *rel* image, use the menu command **File → Print chart image...** and select a printer. Alternatively, you may use the  button to print the image directly to the currently selected system printer.

6.2. Complex types

There are several complex types used throughout *NAADSM*: herds (also called units), production types, and production type combinations.

6.2.1. Herds or units

The smallest unit in an *NAADSM* scenario is a herd (recall that *NAADSM* simulates disease at the herd level, rather than at the level of the individual animal). Within *NAADSM*, the more generic term “unit” is used instead of “herd” to refer to a collection of animals. These two terms are used interchangeably in this guide.

Each unit has the following attributes, which must be specified by the user:

- Geolocation, specified by latitude and longitude
- Number of animals in the unit (unit size)
- Production type (see Section 6.2.2)
- Initial disease state

As described in Section 8.5 and Appendix A, there are five disease states, as well as two more possible unit states that act somewhat like disease states. These are as follows:

- Susceptible to infection
- Latently infected
- Infectious but not showing clinical signs of disease
- Infectious and showing clinical signs
- Naturally immune to infection
- Vaccine immune to infection
- Destroyed

If a unit is not specifically assigned an initial disease state, it is assumed to be susceptible to disease.

At least one unit in a scenario must have an initial disease state of latent, subclinical, or clinical: otherwise, there will be no potential source of infection, and the simulation will not run.

6.2.2. Production types

Every unit in a simulation has a particular production type. A production type defines a group of herds with similar disease transmission probabilities, disease manifestation, disease detection probabilities, and control strategies. Production types are typically based on animal species and/or management practices applied to particular types of livestock operations. For example, beef cattle would likely be a separate production type from swine, since the biological activity of most diseases will be different in cattle than in swine. Beef cattle might also be a separate production type from dairy cattle: although the disease dynamics may be quite similar within one species, very different management practices might influence the spread or detection of disease.

6. Types of input parameters

Users may define as few or as many production types as they wish. Depending on the level of desired specificity -- as well as the quality of the data available -- production types may be quite broad (*e.g.*, cattle, swine, poultry) or very narrow (*e.g.*, beef sale yard, beef cow-calf operation <50 head, beef cow-calf operation >50 head, beef feedlot).

Production types form the basis for many parameters in an *NAADSM* scenario. Each production type has the following attributes:

- Duration of the latent, subclinical, clinical, and naturally immune disease stages (probability density functions: see Section 6.1.5)
- Duration of the vaccine immune stage (a probability density function: see Section 6.1.5)
- Detection and surveillance parameters, some of which are relational functions (see Section 6.1.6)
- Control (vaccination and destruction) measures, some of which are relational functions (see Section 6.1.6)
- Direct costs associated with control measures

6.2.3. Production type combinations

Disease is spread from one unit to another based on contact rates and probabilities of disease transfer, which are set for each pair of production types. The spread of disease from cattle to cattle, for example, might involve a much higher contact rate than spread from cattle to swine. Similarly, the probability of disease transfer from swine to cattle may be different from the probability of transfer from cattle to swine.

All disease spread parameters are set for each pair of production types in a scenario. Spread may occur between some or all of these production type pairs.

7. Basic file operations: **File** menu commands

The **File** menu includes tools for managing scenario files, as well as the means to create, open, and save files. Scenario files are saved as *Microsoft Access*-compatible (*.mdb) files. These files store all information included in a scenario, such as the geolocation, size, and production type of each herd; the disease spread and control parameters for each production type; and output generated by simulating the scenario.

NOTE: NAADSM scenario files and Microsoft Access

An NAADSM scenario file may be opened and viewed with any application capable of opening *Microsoft Access* databases, such as *Microsoft Access* itself, a utility like *JetSQLConsole* (available from <<http://www.reevesdigital.com/jetsql/console>>), or a custom script or application. Other applications, for example, might use output generated by NAADSM for analyses not built into NAADSM. Because all of this data is available in a commonly used format, data import or export from or to other sources should be relatively simple.

A complete description of the NAADSM database schema is provided in Appendix E.

7.1. **New scenario file**

The **File → New scenario file** command has two options: **Empty scenario** creates a completely empty scenario file: the user must enter all disease and control parameter values, and must import or create an animal population to which these parameters apply. The second option, **US Midwest scenario**, creates the sample scenario which we saw earlier in Section 5. This sample scenario contains a reasonably realistic scenario (complete with an animal population, production types, and disease and control measures) for an outbreak of foot and mouth disease in the midwestern United States. In some cases, it may be easier to modify this sample scenario than to create a completely new scenario from scratch.

7.2. **Open scenario file...**

This option opens an existing scenario file. Only one scenario may be open at a time: you will have to close any currently open scenario file (see Section 7.6) in order to use a different scenario.

If the scenario file was created with an earlier version of NAADSM, you may see an on-screen notification. In most cases, files from older versions will be updated without further action on the part of the user.

In (hopefully rare) instances, new versions of NAADSM will correct errors found in earlier versions of the application. In these cases, it may be desirable to re-run your scenario

7. Basic file operations

The File menu

with the corrected version. Should this situation arise, please contact a member of the *NAADSM* development team (see Appendix G) for more information about the sources of and solutions for these problems.

NOTE: Compacting NAADSM scenario files

Users of *Microsoft Access* may be familiar with the **Tools → Database Utilities → Compact and Repair Database...** command in Access. Under ordinary circumstances, it is not necessary to compact and repair *NAADSM* scenario files: these tasks are carried out automatically as needed by the *NAADSM* application when a scenario file is opened.

7.3. Export scenario...

As noted in Section 2 and described in Appendix F, implementations of *NAADSM* exist for *Microsoft Windows*-based PCs (*NAADSM/PC*) as well as *Linux/Unix*-based parallel computing platforms (*NAADSM/SC*). The **File → Export scenario...** command converts a scenario from the relational database format used in *NAADSM/PC* to the plain-text extensible markup language (***.xml**) format used by *NAADSM/SC*. Please see Appendix F for more information about *NAADSM/SC*.

7.4. Save scenario file

Use this menu item to save the current scenario with any changes that you have made or results that may have been generated.

NOTE: Using an NAADSM scenario file simultaneously in several applications

As noted in Section 7, *NAADSM* scenario files may be opened with other applications, such as *Microsoft Access*. It is not advisable to have a scenario file open in multiple applications simultaneously if you are making changes to it: errors will result if you attempt to save changes to the scenario file in one application while it is already open in another application.

If you need to make changes to an *NAADSM* scenario file, make sure that it is open in only one application. Once changes are saved, you may then re-open the file in another application.

7.5. Save As...

This option saves the current scenario under a different file name. Developing and populating a scenario from scratch is time-consuming: in most cases, it is more efficient to open

an existing scenario, alter parameters as necessary, and save the scenario under a different file name.

7.6. **Close scenario file**

Use the **File** → **Close scenario file** command to close an open scenario. If changes have been made to the scenario since it was last saved, you will be prompted to save the changes or close the file without saving changes.

7.7 **Exit**

The **Exit** command closes the open scenario if there is one, and exits the *NAADSM* application. If changes have been made to an open scenario since it was last saved, you will be prompted to save the changes or close the file without saving changes.

8. Setting up a scenario: The **Scenario parameters** menu

The commands of the **Scenario parameters** menu are the most heavily used in *NAADSM*. Options in this menu allow you to create or modify an animal population, specify parameters for disease spread and control, enter costs associated with control measures, and select optional output data. All user interaction with the scenario database file occurs via these commands.

8.1. Using the wizard screens

NAADSM parameters are specified with a series of sequential “wizard” screens. Most of the commands of the **Scenario parameters** menu are used to select one of these wizard screens (Figure 8-1). Each of the 20 or so screens shows the set of options associated with a particular aspect of a disease scenario. Depending on your choices on a particular screen, you will be guided to the next appropriate screen to enter any related parameters.

If you have not already done so, this is a good time to review the description of the conceptual *NAADSM* model (Appendix A). The scenario parameters screens and their various input options correspond very closely to the sections of the model description.

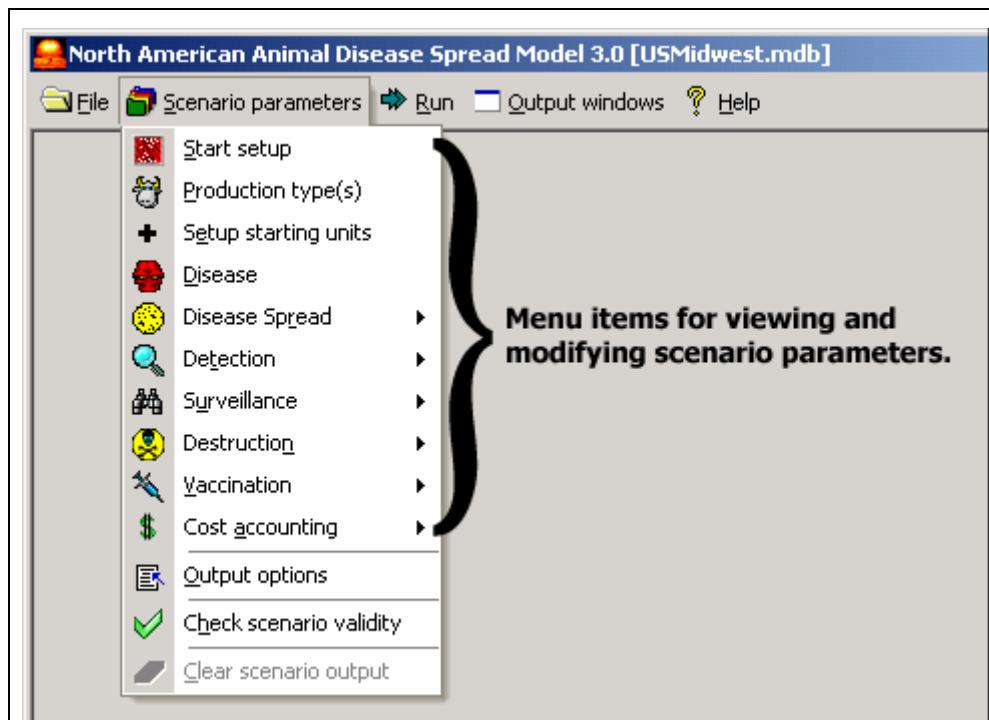
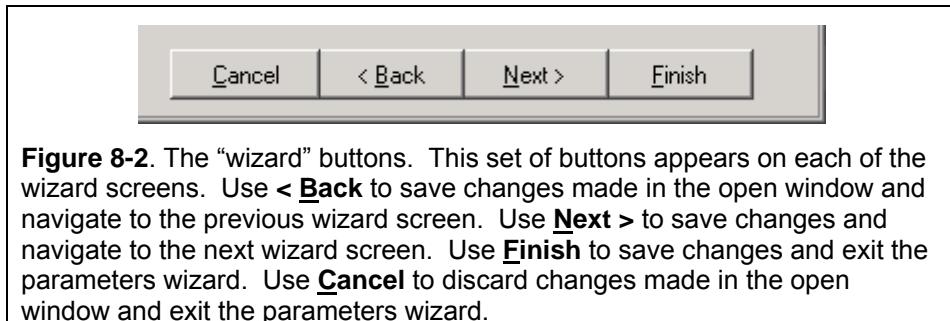


Figure 8-1. The “wizard” commands of the **Scenario parameters** menu. Each of these menu commands activates a wizard screen, used to view or set scenario parameters associated with the indicated aspect of the scenario.

8. Setting up a scenario

The **Scenario parameters** menu

You can navigate linearly through the wizard screens, using the **< Back** or **Next >** buttons in the lower right corner of each screen (Figure 8-2). Alternatively, you may select one of the other items from upper section of the **Scenario parameters** menu (Figure 8-1) to skip directly to the parameters of interest. Each of the **Scenario parameters** commands and their associated wizard screens will be discussed sequentially in the following subsections.



Most of the screen shots displayed in the following sections show the sample US Midwest scenario (see Section 5).

8.2. The Start setup window

The **Start setup** command and wizard screen (Figure 8-3) display several basic options associated with a scenario.

Scenario description: the scenario description is used to store comments concerning the particular scenario. This description is completely optional, but potentially very useful.

Number of iterations: the number of iterations is the number of times that you want to run a particular scenario. Recall from Section 3 that each time a stochastic model is run (each iteration), a different result will be produced. The more iterations you run, the better distribution you obtain for your results. You can then determine which results occur commonly and which are unusual, allowing you to make more informed decisions.

One to five iterations are typically sufficient for scenario development and testing: you can watch the iteration in progress (see Section 10.2) to get a “feel” for the way a scenario might progress.

For analytical purposes, the current standard in stochastic modeling is 100 – 10,000 iterations, though that range may be impractical for large or complex data sets. Ten iterations may be a good starting point.

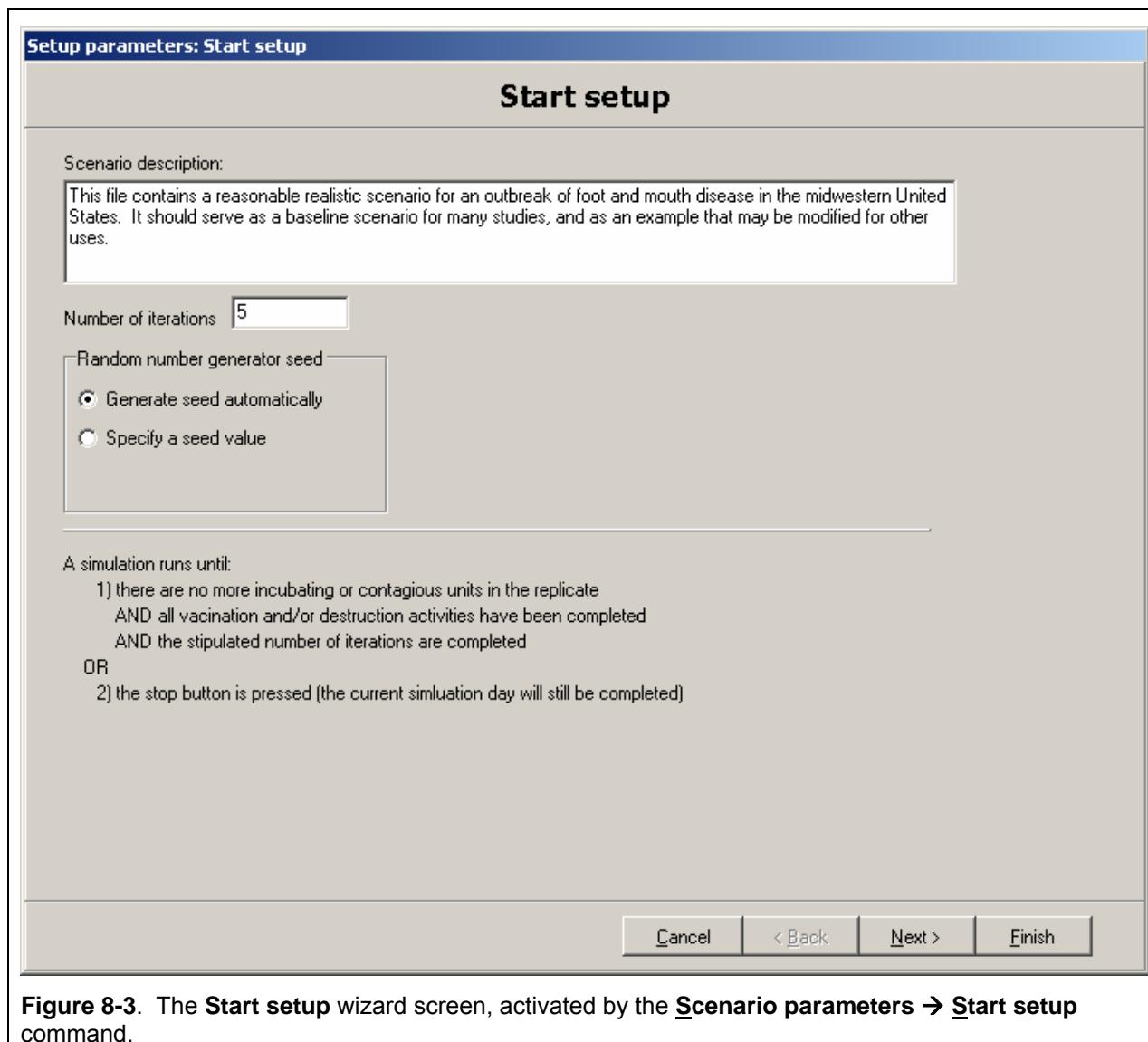


Figure 8-3. The **Start setup** wizard screen, activated by the **Scenario parameters → Start setup** command.

Random number generator seed: this option is used to specify a particular seed value for the random number generator, if desired.

Computers do not generate truly random numbers: instead, they produce a pseudorandom sequence of numbers. The sequence of numbers produced depends on the “seed value” used to initiate the so-called random number generator. If the same seed value is used over and over again, a random number generator will produce exactly the same sequence of “random” numbers.

For testing purposes, it may be helpful to specify a seed value: with a specified seed value, a particular scenario will produce the same results time after time. For analytical purposes, automatic seed generation is typically used.

8. Setting up a scenario
The **Scenario parameters** menu

8.3. The Production type(s) window

The **Production type(s)** command and wizard screen (Figure 8-4) are used to create (**Add production type**), rename (**Modify selected production type**), or delete (**Remove selected production type(s)**) production types (see Section 6.2.2).

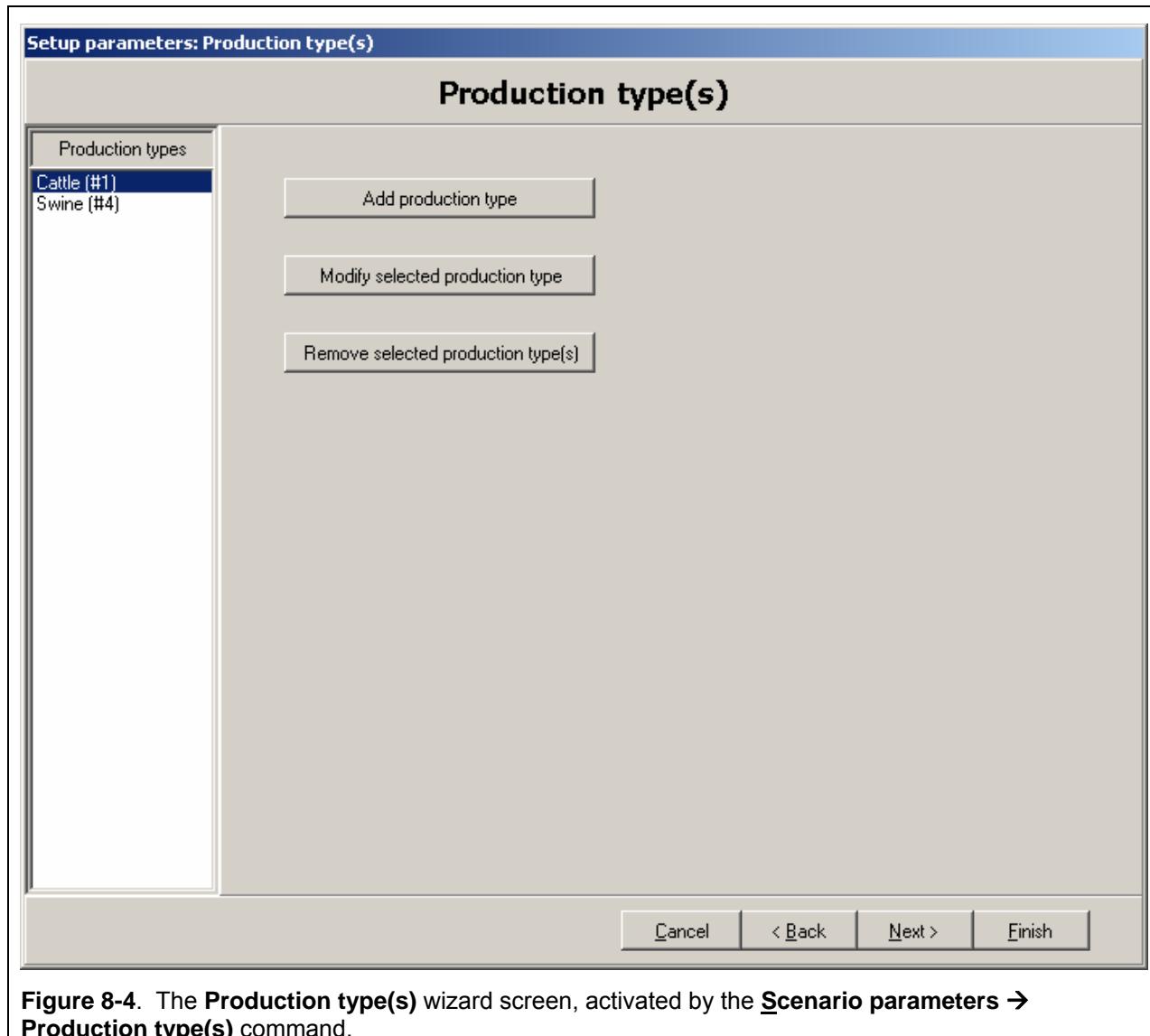


Figure 8-4. The **Production type(s)** wizard screen, activated by the **Scenario parameters → Production type(s)** command.

Once a production type has been created, subsequent wizard screens may be used to set all of the pertinent disease, contact, and control parameters.

WARNING: Use care when removing production types

When a production type is removed, all herds of that type will be deleted from the scenario as well.

NOTE: Production type ID numbers

When a production type is added to a scenario file, it is automatically assigned an ID number. Most of the time, these ID numbers are used entirely behind the scenes. On occasion, however, they are used to refer to production types (for example, when population data is imported: see Section 8.4.3). For convenience, the **Production type(s)** window displays these ID numbers, but is it not necessary (or possible) to manually enter ID numbers: NAADSM application will assign them as needed.

8.4. Setting up a population of herds: The View/edit starting units window

Scenario parameters → Set up starting units is used to activate the **View/edit starting units** wizard screen (Figure 8-5). Use this screen to import, edit, and/or export your herd population data.

NOTE: Editing large herd data sets

The population editor in NAADSM will work for up to approximately 65,000 herds. If you have more than 65,000 herds in your population data set, you will need to edit the herds in the scenario database file with a different application.

Microsoft Access could be used for this purpose. Even better, a text-based console application like JetSQLConsole (available from <<http://www.reevesdigital.com/jetsqlconsole>>) provides a very efficient way to work with large database files: many records can be updated with just a few commands.

8. Setting up a scenario
 The **Scenario parameters** menu

Setup parameters: View/edit starting units

View/edit starting units

Order	ID	Production type	Herd size	Latitude	Longitude	Status	Days left in status
31	31	Swine (#4)	37	41.28056	-92.18361	Susceptible	Unspecified
898	898	Cattle (#1)	40	41.28139	-92.04056	Susceptible	Unspecified
3810	3810	Cattle (#1)	15	41.28139	-92.01222	Susceptible	Unspecified
4452	4452	Swine (#4)	12	41.28333	-91.53611	Susceptible	Unspecified
135	135	Swine (#4)	4	41.28556	-91.71389	Susceptible	Unspecified
2939	2939	Cattle (#1)	29	41.28833	-92.49639	Susceptible	Unspecified
1403	1403	Cattle (#1)	41	41.29083	-92.02583	Susceptible	Unspecified
1924	1924	Cattle (#1)	126	41.29083	-92.76778	Susceptible	Unspecified
4451	4451	Swine (#4)	46	41.29111	-91.81639	Susceptible	Unspecified
4450	4450	Swine (#4)	10	41.29194	-91.81528	Susceptible	Unspecified
2499	2499	Cattle (#1)	8	41.29222	-92.53027	Susceptible	Unspecified
3966	3966	Cattle (#1)	71	41.295	-91.73889	Susceptible	Unspecified
1049	1049	Cattle (#1)	64	41.29528	-92.07166	Susceptible	Unspecified
1	1	Swine (#4)	19	41.30028	-91.96638	Susceptible	Unspecified
570	570	Cattle (#1)	8	41.30056	-93.275	Susceptible	Unspecified

4520 units

Cancel | **< Back** | **Next >** | **Finish**

Figure 8-5. The **View/edit starting units** screen, activated by the **Scenario parameters → Set up starting units** command.

8.4.1. Editing the attributes of a herd

8.4.1.1. Changing basic attributes

Recall from Section 6.2.1 that each herd has a production type, a geolocation (specified by latitude and longitude), a size (the number of animals in the herd), and an initial disease state (susceptible to infection, latent, infectious but not showing clinical signs of disease, infectious and showing clinical signs, naturally immune to infection, vaccine immune to infection, or destroyed: see Section 8.5). These attributes may be altered for any herd directly in the spreadsheet-like herd list window (Figure 8-5): simply click inside the cell that you want to change, and enter a new value. Production type and disease status are changed by selecting a

8. Setting up a scenario
The **Scenario parameters** menu

different option from the dropdown menu that appears when you click on a cell in columns labeled **Production type** or **Status** (Figure 8-6). For latitude, longitude, or herd size, simply type a number in the cell.

(a)

Order	ID	Production type
31	31	Swine (#4)
898	898	Cattle (#1) Swine (#4)

(b)

Order	ID	Production type	Herd size	Latitude	Longitude	Status
30	30	Cattle (#1)	48	42.935	-92.63167	Susceptible
31	31	Cattle (#1)	37	41.28056	-92.18361	Susceptible
32	32	Swine (#4)	195	41.64417	-90.72222	Latent
33	33	Swine (#4)	14	43.04194	-96.14722	Subclinical
34	34	Swine (#4)	41	42.77833	-95.81333	Clinical

Figure 8-6. Changing the production type and initial disease status of a herd. **(a)** The unit with ID 31 is being changed from a production type “Swine” to production type “Cattle”. The dropdown menu automatically lists the production types that have been defined for the scenario. Note the display of the production type ID numbers (in this case, 1 for “Cattle”, 4 for “Swine”), as described in Section 8.3. **(b)** The initial disease state of unit 31 is being set to “Subclinical”, i.e., infectious but without clinical signs of disease.

NOTE: Herd order and ID

You may have noticed two additional columns in Figures 8-5 and 8-6, labeled **Order** and **ID**. These two numbers may be used for sorting and filtering if desired (see Section 8.4.3), but otherwise are not significant. Each herd is assigned a unique ID number when it is incorporated into a scenario database. ID numbers are generally but not always sequential, and do not necessarily start at 1. The **Order** column shows the order in which herds are actually stored in the database, which always starts at 1 and is always sequential.

8.4.1.2. Changing the number of days left in status

Figure 8-5 shows one additional column, labeled **Days left in status**. Recall from Section 6.2.1 that every herd in a scenario is assigned an initial disease state. Normally, the duration of each of these disease stages will be automatically determined by *NAADSM* as a scenario runs (see Section 6.2.1 and Appendix A for more details).

In some cases (for example, if you want to begin a simulation in a later stage of a disease outbreak rather than at the start of the outbreak), you may wish to specify the number of days that a particular herd or herds has in its initial disease state before it makes a transition to the

8. Setting up a scenario

The **Scenario parameters** menu

next stage. This can be done by setting the value of **Days left in status** to something other than **Unspecified**. Simply enter a whole number in this field to change the initial setting. If the value of **Days left in status** is either -1 or **Unspecified**, NAADSM will determine the duration of the initial stage according to the disease stage distributions (see Section 6.2.1 and Section 8.6).

Note that a value other than **Unspecified** for a herd which is initially susceptible will have no effect: herds are susceptible indefinitely, unless they become infected, are vaccinated, or are destroyed during the simulation.

8.4.2. Sorting and filtering the list of herds

A lengthy list of herds may be difficult to edit. To simplify the process, the list of herds may be sorted into a particular order, or may be filtered to display only herds that match a particular criterion.

8.4.2.1 Sorting herds

Herds may be sorted in ascending or descending order by any column. Use the **Sort by** dropdown menu to select the column to use for sorting, and select a sort direction (Figure 8-7a). Alternatively, click inside a column header to sort by the selected column. Click on the column header again to reverse the sort direction. A sort direction indicator (a small triangular arrow) will appear in the column header to indicate the sorted column and the direction of the sort (Figure 8-7).

(a)

Order	ID	Production type	Herd size	Latitude	Longitude	Status	Days left in status
1	4355	Cattle (#1)	359	43.1275	-94.08527	Susceptible	Unspecified
2	4356	Cattle (#1)	23	42.62806	-93.08722	Susceptible	Unspecified
1	1	Swine (#4)	19	41.30028	-91.96638	Susceptible	Unspecified
2	2	Swine (#4)	15	43.28833	-96.31611	Susceptible	Unspecified

(b)

Order	ID	Production type	Herd size	Latitude	Longitude	Status	Days left in status
4355	4355	Cattle (#1)	359	43.1275	-94.08527	Susceptible	Unspecified
4356	4356	Cattle (#1)	23	42.62806	-93.08722	Susceptible	Unspecified
1	1	Swine (#4)	19	41.30028	-91.96638	Susceptible	Unspecified
2	2	Swine (#4)	15	43.28833	-96.31611	Susceptible	Unspecified

4520 units

Cancel < Back Next > Finish

Figure 8-7. Sorting herds by production type. **(a)** Selecting production type from the **Sort by** dropdown menu. Note that, prior to sorting, herds are sorted in ascending order by **Order**, as shown by the sort direction indicator (the small gray arrow) in the **Order** column. **(b)** After sorting by production type. The sort direction indicator now appears in column **Production type**.

8.4.2.2. Filtering herds by simple criteria

You can display a subset of your population data in the **View/edit starting units** window by using a filter. Filters may be created based on the same herd attributes used for sorting. Use the **Filter by** dropdown menu to select an attribute to use for a filter (Figure 8-8a). Next, enter a value to use as the filter criterion. Depending on the selected filter attribute, the filter criterion might be entered in another dropdown menu or in a text box (Figure 8-8b shows an example of the latter).

8. Setting up a scenario

The **Scenario parameters** menu

(a)

Order	Type	Herd size	Latitude	Longitude	Status	Days left in status
1	Herd size	19	41.30028	-91.96638	Susceptible	Unspecified
2	Geographic range	15	43.28833	-96.31611	Susceptible	Unspecified
3	Specific latitude	20	42.98333	-96.13111	Susceptible	Unspecified
4	Specific longitude	3	42.9025	-91.45111	Susceptible	Unspecified
	Status					
	Days left in status					

(b)

Filter by: Herd size Unit size: 19

(c)

Setup parameters: View/edit starting units

View/edit starting units

File Edit ||

Sort by: Order Sort order: Ascending Descending

Filter by: Herd size Unit size: 19

Order	ID	Production type	Herd size	Latitude	Longitude	Status	Days left in status
4300	4300	Cattle (#1)	19	42.69416	-96.06805	Susceptible	Unspecified
4315	4315	Cattle (#1)	19	42.48444	-91.51334	Susceptible	Unspecified
4358	4358	Swine (#4)	19	43.20778	-92.23833	Susceptible	Unspecified
4417	4417	Swine (#4)	19	42.05833	-92.37305	Susceptible	Unspecified
4449	4449	Swine (#4)	19	41.30583	-91.48666	Susceptible	Unspecified

60 units shown (of 4520 units total)

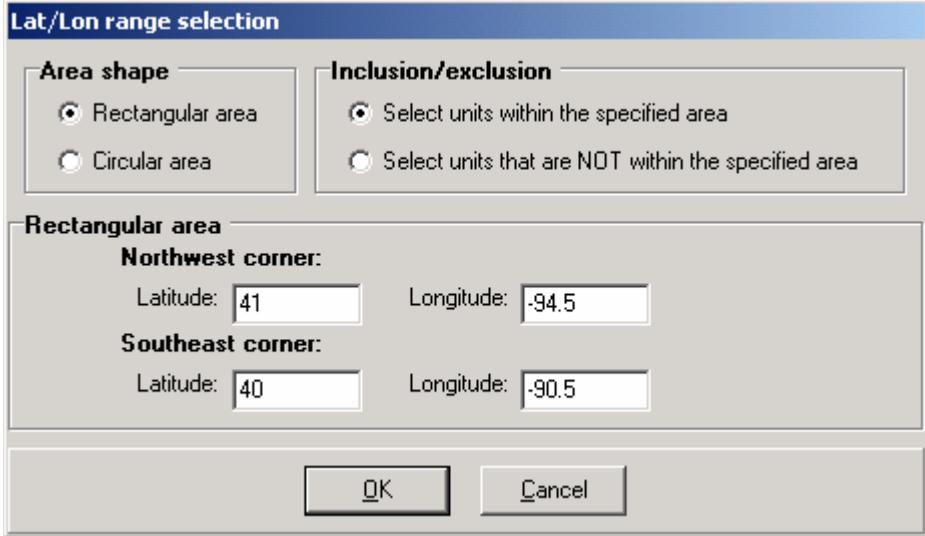
Cancel < Back Next > Finish

Figure 8-8. Filtering herds by herd size. **(a)** Selecting **Herd size** from the **Filter by** dropdown menu. **(b)** Specifying a herd size. Click on the button or press the **Enter** key to apply the filter criterion. **(c)** The filtered list. Note the message in the lower left portion of the window, displaying the number of herds which meet the filter criterion. Also note that the sort order is unchanged.

8.4.2.3. Filtering herds by geographic range

When editing herds, it may be useful to restrict the displayed list to herds contained within a specific geographic range. The **View/edit starting units** window will allow you to filter herds based on rectangular or a circular area. For a rectangular area, specify the latitude and longitude for the northwest and southeast corners of the bounding region (Figure 8-9).

8. Setting up a scenario
The **Scenario parameters** menu


(a)

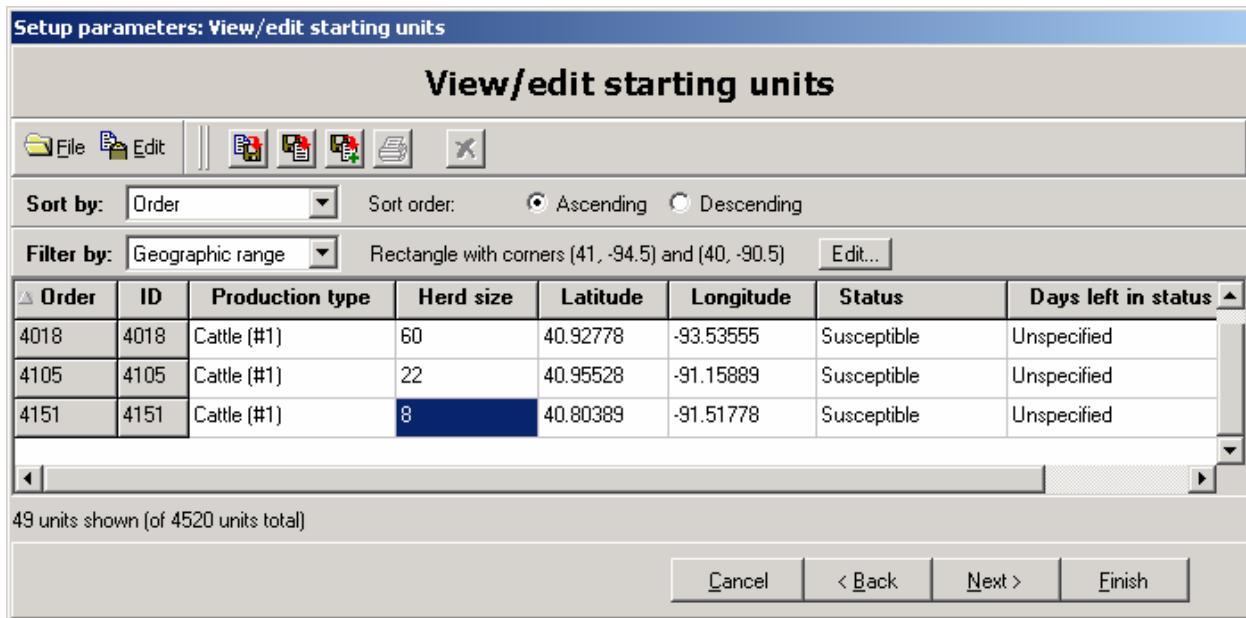

(b)

Figure 8-9. Filtering herds by rectangular geographic range. **(a)** Specifying the northwest and southeast corners of the bounding region. The coordinates initially displayed for these points will encompass the entire study area. Change these coordinates to specify a smaller area. **(b)** The results of the filter. Click on the **Edit...** button to change the geographic range.

For a circular area, specify the center of the region and a radius in kilometers (Figure 8-10).

8. Setting up a scenario

The **Scenario parameters** menu

Use the **Inclusion/exclusion** options to indicate whether you want to display the herds that are within the specified region, or the herds that are not within the specified region.

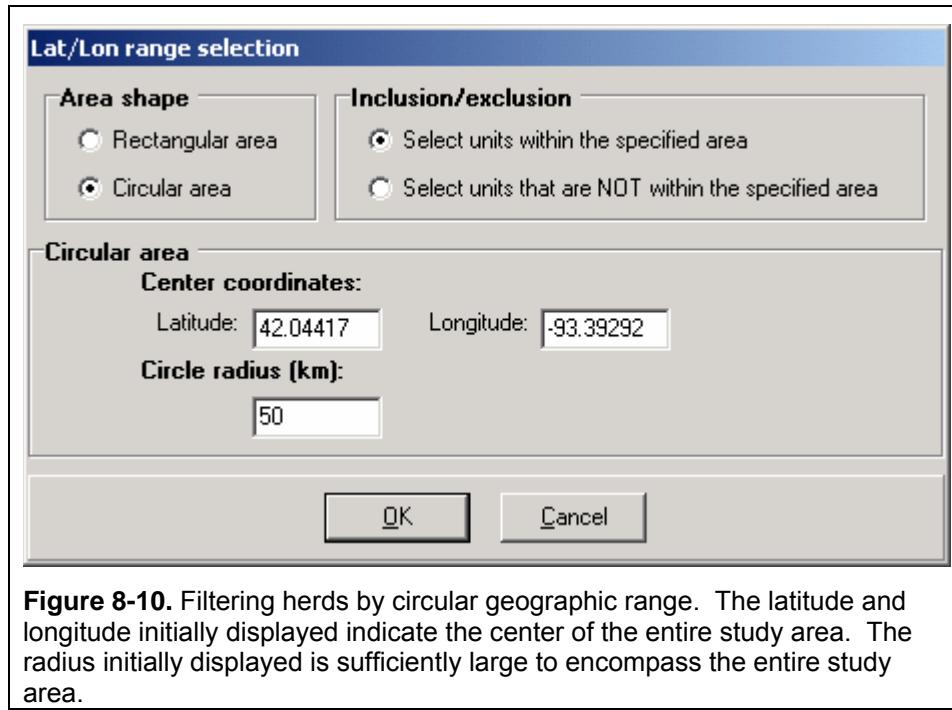


Figure 8-10. Filtering herds by circular geographic range. The latitude and longitude initially displayed indicate the center of the entire study area. The radius initially displayed is sufficiently large to encompass the entire study area.

8.4.3. Importing and exporting herds

If you have your own herd data, you may want to import it into an *NAADSM* scenario. There are several commands on the **File** menu of the **View/edit starting units** window that will allow you to import your own herd data. (Note this is not the **File** menu of the main *NAADSM* application window, which was covered in Section 7). Another **File** menu command will allow you to export a list of herds from a scenario into a plain text file suitable for use in other applications.

8.4.3.1. File formats for import and export of herd data

NAADSM will import herd data from properly formatted plain text comma-delimited (*.**csv**) or extensible markup language (*.**xml**) files. Both of these file formats are plain text formats, which should be easy to generate from any of a number of other applications. *NAADSM* will also export herd data to either of these formats.

To be successfully imported into *NAADSM*, your data file must include a unique herd ID number, a production type (usually indicated by name), the herd size, the initial disease state, and the latitude and longitude for each herd. Files without all of these fields cannot be imported. Optionally, your file may include a field for days left in status.

Column headers must be included in your *.**csv** file for import, as shown in Table 8-1. Columns may be given in any order.

Table 8-1. Fields in NAADSM *.csv files.

Field name	Field description
<i>HerdID</i>	Unique integer identifier for each unit. ID must be greater than 0.
<i>ProductionType</i>	Identifier for the unit's production type (see note below).
<i>HerdSize</i>	Integer indicating the number of animals in the unit.
<i>Lat</i>	Real (floating point) number indicating the latitude of the unit. Values must be between -90 and 90, inclusive.
<i>Lon</i>	Real (floating point) number indicating the longitude of the unit. Values must be between -180 and 180, inclusive.
<i>Status</i>	Code indicating the unit's disease transition state at the beginning of the simulation (see Table 8-2).
<i>DaysLeftInStatus</i>	(Optional) Integer indicating the number of days the unit has remaining in its initial status (see Section 8.4.1.2).

NOTE: Production types in imported herd files

As mentioned above, production type names are usually required in plain text files containing herds for import. Production types must be created (see Section 8.3) prior to attempted to import herds.

Production type names are not case sensitive, but otherwise they must appear in a text file in exactly the same way that they are given on the **Production type(s)** window (see Section 8.3). For example, "Swine" and "swine" are equivalent, but "BeefCattle" (without a space) and "Beef Cattle" (with a space) are not. Error messages will be displayed if NAADSM attempts to import a herd file that contains production type names that do not appear in the scenario.

Production types may also be specified by ID number: see Appendix C for details.

Initial disease state is usually indicated by a single upper case character, as shown in Table 8-2.

Table 8-2. Character codes used for disease transition states

Transition state	Single character code
Susceptible	S
Latent	L
Subclinical infectious	B
Clinical infectious	C
Naturally immune	N
Vaccine immune	V
Destroyed	D

8. Setting up a scenario

The Scenario parameters menu

Figure 8-11 shows a few herds from a *.csv file suitable for import into NAADSM. A detailed description of the herd file *.csv format may be found in Appendix C. A description of the *.xml file formats used by NAADSM is beyond the scope of this guide, but resources are listed in Appendix F.

```
HerdID,ProductionType,HerdSize,Lat,Lon,Status,DaysLeftInStatus
1,"Swine",19000,42.9,-94.899999,L,10
2,"Swine",15000,42.8,-94.581999,L,-1
3,"Swine",20000,42.7,-94.187899,S,-1
4,"Swine",3000,42.6,-93.663299,S,-1
5,"Swine",63000,42.5,-92.866699,S,-1
6,"Swine",52000,42.4,-91.177999,S,-1
7,"Swine",45000,42.3,-91.299999,S,-1
8,"Swine",4000,43.2,-95.499999,S,-1
9,"Swine",28000,41.1,-90.699999,S,-1
```

Figure 8-11. Part of a *.csv file suitable for import into an NAADSM scenario. Recall from Section 8.4.1.2 that a value of -1 for DaysLeftInStatus is equivalent to **Unspecified**.

8.4.3.2. Import and replace existing herds

If your NAADSM scenario file contains a herd population that you no longer want, you can use the command **File → Import and replace existing unit list...** or, equivalently, click on  in the **View/edit starting units** toolbar (Figure 8-12). Then select the *.csv file that you wish to import.

If you replace an existing unit list, the herd ID numbers in your *.csv file (see Section 8.4.3.1) will be preserved: contrast the situation when new herds are appended to an existing list, described in Section 8.4.3.3.

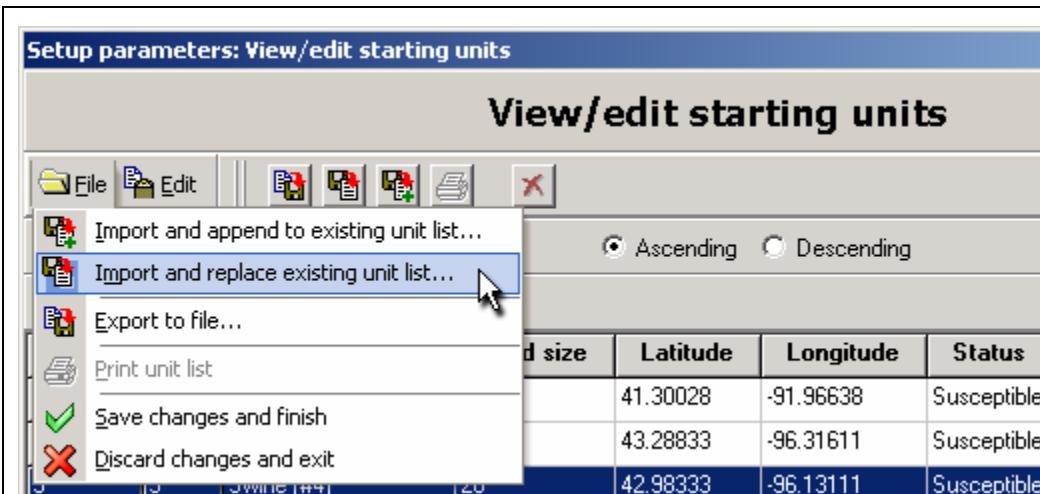


Figure 8-12. The **View/edit starting units** toolbar. This tool bar contains the **File** menu (open in the figure), the **Edit** menu, and a collection of useful buttons.

In the event that an error occurs, *NAADSM* will display a (hopefully) helpful error message. Make sure that your file format corresponds to the description in Appendix C, that all production types exist in your scenario exactly as they appear in your ***.csv** file, and that your file is not already open in another application.

8.4.3.3. Import and append to existing herd list

If you want to add new herds to the population data already stored in your scenario file, use the command **File → Import and append to existing unit list...** or, equivalently, click on  in the **View/edit starting units** toolbar (Figure 8-12). Then select the ***.csv** file that you wish to import.

If you append new units to an existing unit list, the herd ID numbers in your ***.csv** file cannot be preserved: new herds will automatically be given new ID numbers, which do not conflict with ID numbers that already exist in the scenario file.

8.4.3.4. Exporting herds

Use the **File → Export to file...** command or the  button to export a herd list to a plain text (***.csv** or ***.xml**) file. A window like the one shown in Figure 8-13 will be displayed. Using the options shown in Figure 8-13 will produce a ***.csv** file in the format discussed above (Section 8.4.3.1). Other options are described in Appendix C. Simply enter a file name and click **OK** to save your herd list as a file.

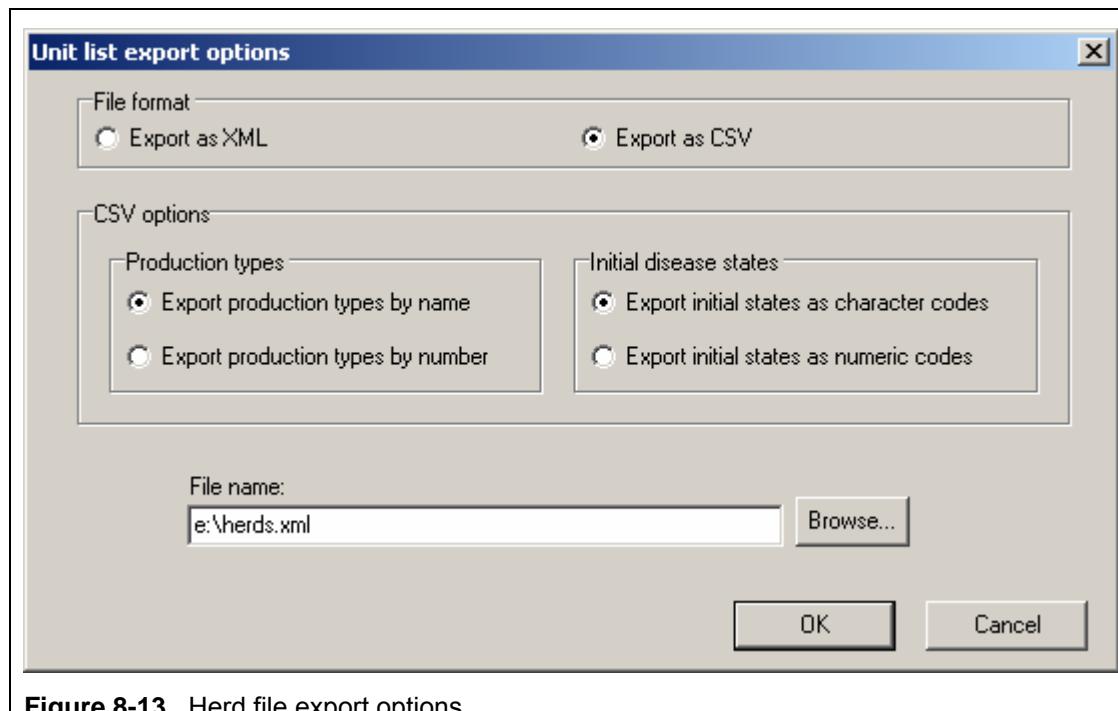


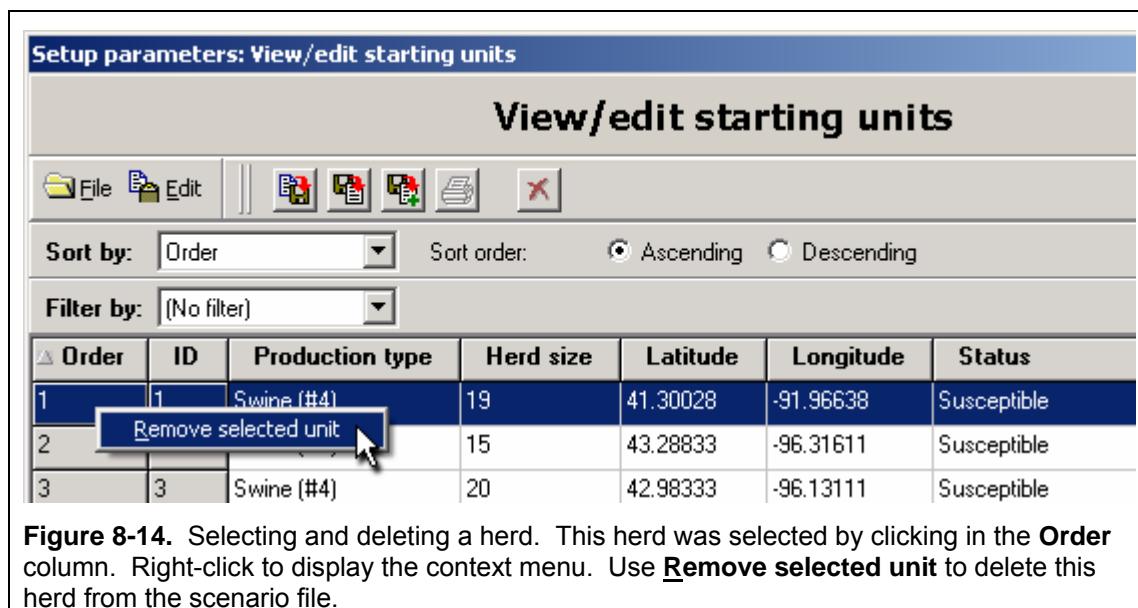
Figure 8-13. Herd file export options.

8. Setting up a scenario
The **Scenario parameters** menu

8.4.4. Deleting herds

To remove an existing herd, click on the **Order** or **ID** column in the row that contains the herd you wish to delete (Figure 8-14). You may then delete the herd by right-clicking and selecting **Remove selected unit** from the context menu; by using **Edit → Remove selected unit(s)**, or by clicking on  in the toolbar. (These options are available only when one or more herds are selected.)

To select multiple units at once, hold the **Shift** key while clicking in the **Order** or **ID** column.



8.4.5. Finalizing changes to starting units

As in all of the wizard screens described in Section 8, clicking on the **Back >**, **Next >**, or **Finish** buttons will save your changes to the current NAADSM scenario database. Clicking on the **Cancel** button will discard all changes made since starting the screen.

If you have changed, modified, or removed quite a few units, you may need to be a little patient: each unit is a separate record that must be updated in the scenario database file, and this process can take a minute or two. NAADSM will show a progress indicator if extensive database changes are required.

8.5. Modeling disease progression

8.5.1. Concepts of disease progression in NAADSM

NAADSM is a state transition model. Each unit (herd) in a simulated outbreak exists in one of the several disease or disease-like states described in Table 8-3. All animals within a herd are assumed to exist in the same disease state.

Table 8-3. Disease and disease-like transition states used in *NAADSM*.

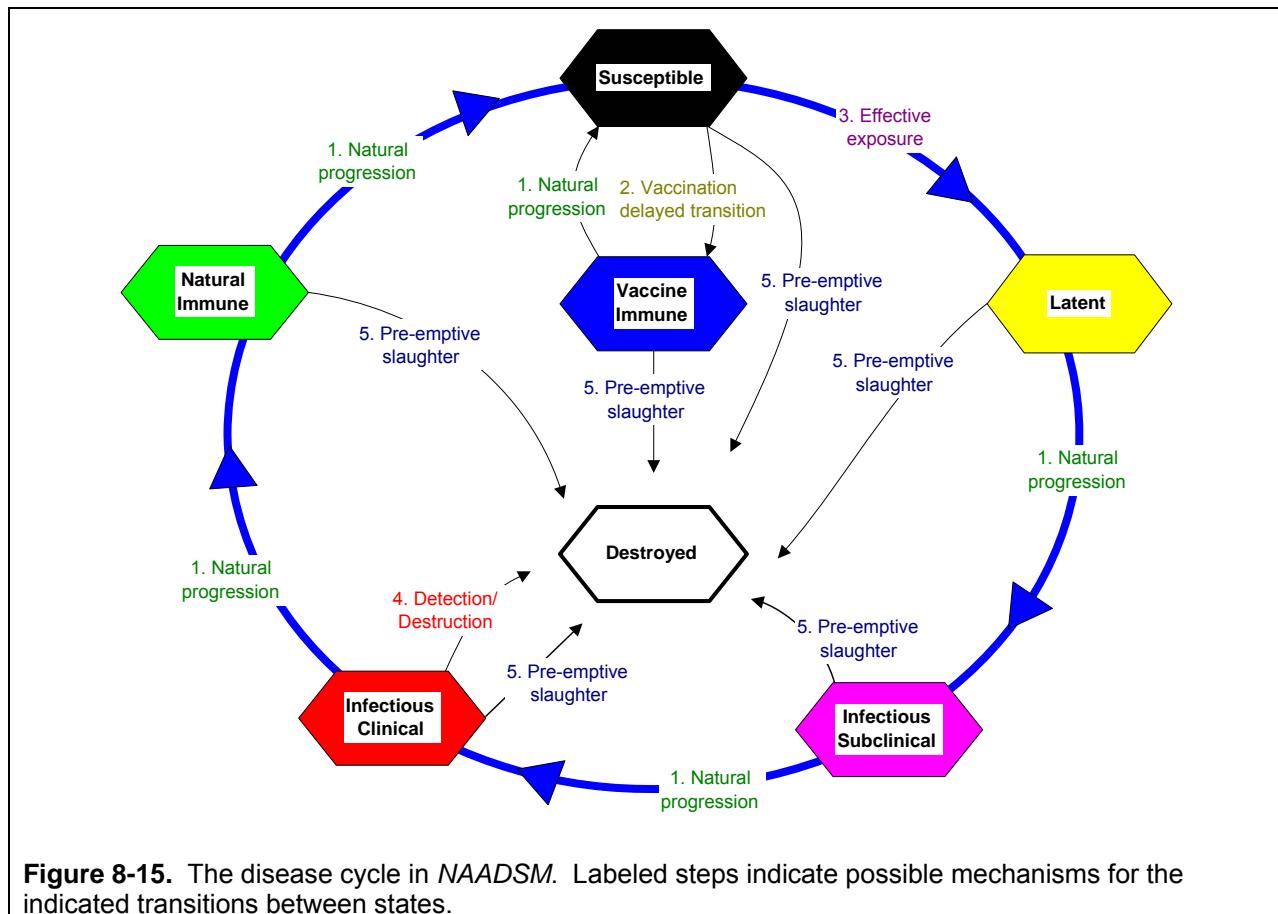
Transition state	Single character code	Description
Susceptible	S	Healthy herds without immunity to infection. Susceptible herds become infected upon effective contact.
Latent	L	Herds which are infected, but not yet shedding the disease agent.
Subclinical	B	Herds which are infected and shedding the disease agent, but not yet showing clinical signs of disease.
Clinical	C	Herds which are infected, shedding the disease agent, and showing clinical signs.
Naturally immune	N	Herds which have progressed through the disease cycle and are immune from further infection.
Vaccine immune	V	Herds which are immune by virtue of vaccination.
Destroyed	D	Herds which have been destroyed (depopulated).

Without intervention, a typical newly infected herd will progress from the latent stage through the subclinical, clinical, and immune stages before again becoming susceptible to infection. If susceptible units are vaccinated (see Section 8.10), they can become vaccine immune for a certain period of time. Herds in any stage may be subject to destruction (depopulation). Figure 8-15 shows all of the disease states in *NAADSM* and all possible transitions from one state to another.

For each production type, the user defines the length in days of all of these periods except the susceptible and destroyed stages. Units are susceptible indefinitely: there is no fixed time period for the susceptible state. Units cease to be susceptible upon infection, destruction, or the onset of vaccine immunity. There is also no fixed time period for units in the destroyed state: once a unit has been destroyed in a particular iteration, it is never “repopulated” during that iteration, and remains in the destroyed state for the remainder of the iteration. The remaining five stages are associated with a time period determined from an appropriate probability density function. The length of time each unit spends in these stages is based on the probability density functions for these stages for the unit’s given production type, as shown below.

8. Setting up a scenario

The **Scenario parameters** menu



8.5.2. Setting disease parameters.

The **Disease** window (Figure 8-16) is used to set the duration of each disease stage (latent, subclinical, clinical, and immune) for each production type (see Sections 6.2.1 and 6.2.2 and Appendix A). Activate the **Disease** window with the **Scenario parameters → Disease** command.

Disease stage parameters should be specified for each production type. Select a production type in the left panel of the **Disease** window, and then adjust parameters as described below.

Transition the disease in units of this production type: select this option if disease occurs in the production type. This option must be selected for at least one production type in a scenario in order for the scenario to run. If the box is checked for the selected production type, the following variables will appear:

Latent period: this variable is a probability density function (*pdf*) defining the duration of the latent period for herds of this production type. Select or create a function with the *pdf* selection box (see Section 6.1.5).

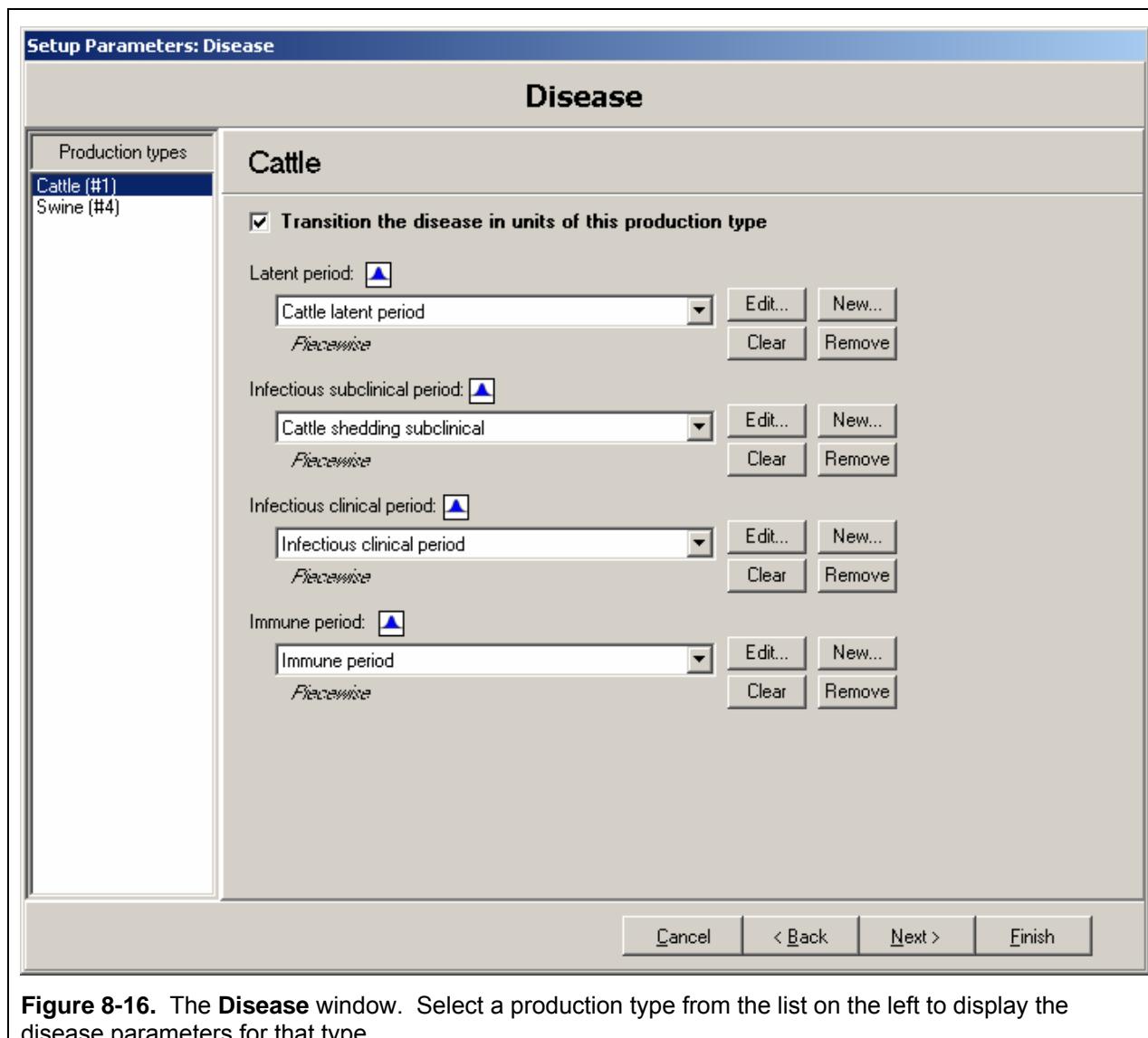


Figure 8-16. The **Disease** window. Select a production type from the list on the left to display the disease parameters for that type.

Infectious subclinical period: this variable is a probability density function (*pdf*) defining the duration of the period when herds of this production type are infectious, yet not clinically ill. Select or create a function with the *pdf* selection box (see Section 6.1.5).

Infectious clinical period: this variable is a probability density function (*pdf*) defining the duration of the period when herds of this production type are clinically ill. Select or create a function with the *pdf* selection box (see Section 6.1.5).

Immune period after infection: this variable is a probability density function (*pdf*) defining the duration of immunity following natural infection. Select or create a function with the *pdf* selection box (see Section 6.1.5).

8. Setting up a scenario

The **Scenario parameters** menu

After describing disease stage durations for a specific production type, move on to the next production type, or use one of the wizard buttons (Figure 8-2).

8.6. Modeling disease spread

A series of four windows is used to set disease spread parameters. These windows are displayed using the commands on the **Scenario parameters → Disease spread** menu (Figure 8-17). Depending on your selections in the **Spread options** window, one or more of the other menu commands may be disabled.

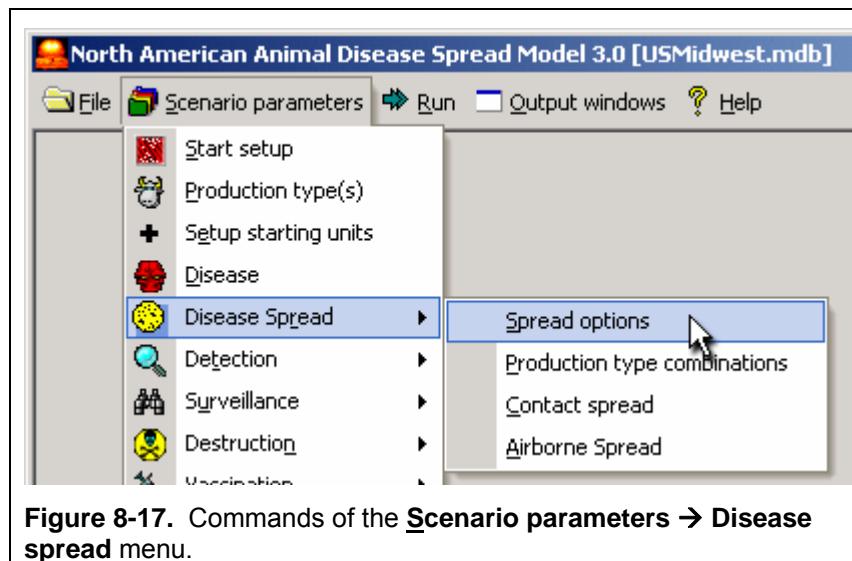


Figure 8-17. Commands of the **Scenario parameters → Disease spread** menu.

8.6.1. The **Spread options** window

The **Spread options** window asks the question **What type of SPREAD would you like to model during simulation runs?** Select whether you want NAADSM to simulate spread via **Contact** with an infected herd (described in Section 8.6.3), **Airborne** spread (described in Section 8.6.4), **Both airborne and contact** spread, or **No spread** at all. Then use one of the wizard buttons (Figure 8-2) to move to the next screen.

NOTE: No spread

An NAADSM scenario will still run when the **No spread** option is selected, but the results of the simulation will be quite dull.

If you choose to simulate airborne spread, you will also be asked whether to use a linear or an exponential decline in the probability of disease spread with increasing distance from a source herd: this option is discussed in more detail in Section 8.6.4.

8.6.2. The **Production type combinations** window

Recall from Section 6.2.3 that disease spread parameters are set for pairs of production types. If contact and/or airborne spread were selected in the **Spread options** window (Section 8.6.1), the **Production type combinations** window will be displayed (Figure 8-18). Use this window to select which production types can spread disease to which other production types. For example, if one of your production types is “feedlot” and one is “beef cow-calf operation”, you may opt to allow disease to spread from a cow-calf operation to another cow-calf operation and from a cow-calf operation to a feedlot, but not from a feedlot to another feedlot.

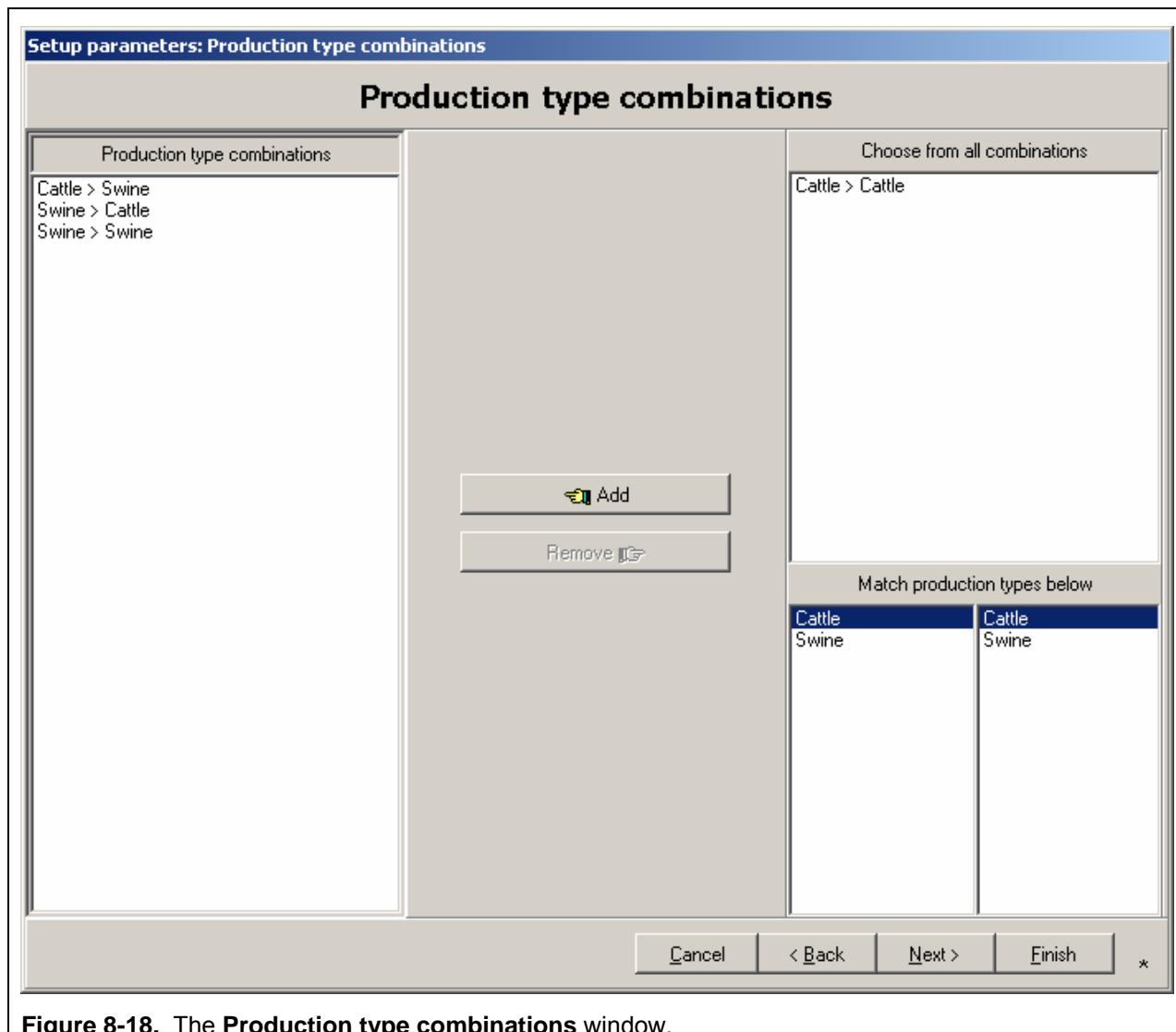


Figure 8-18. The **Production type combinations** window.

The frame on the left side of the window (**Production type combinations**) shows the selected production type combinations which are capable of disease spread. For example, for the item **Cattle > Swine** shown in Figure 8-18, the source of disease is **Cattle** and the destination is **Swine**. In other words, disease can be spread from units (herds) of the production type “Cattle”

8. Setting up a scenario

The **Scenario parameters** menu

to units of the production type “Swine. To remove an item from this frame, select the item, then click the **Remove** button. Hold down the **Ctrl** key to select multiple items. Items can be added to this frame as described below.

The frame in the upper right corner of the window (**Choose from all combinations**) shows all possible combinations of your production types. To add an item from this frame to the **Production type combinations** frame, select the item, then click the **Add** button. Hold down the **Ctrl** key to select multiple items.

The two frames in the lower right corner of the window are labeled **Match production types below**. Each frame has a list of all production types in the current scenario. Select a source production type (one from which disease can be spread) in the left frame, and a recipient production type (one that can be infected by the source production type) in the right frame. Click on the **Add** button, and this production type combination will be added to the list in the panel on the left.

8.6.3. The **Contact spread** window

If contact spread was selected in the **Spread options** window (Section 8.6.1), the **Contact spread** window will be displayed (Figure 8-19). Use this window to modify parameters describing direct and indirect contact spread of disease for each production type combination.

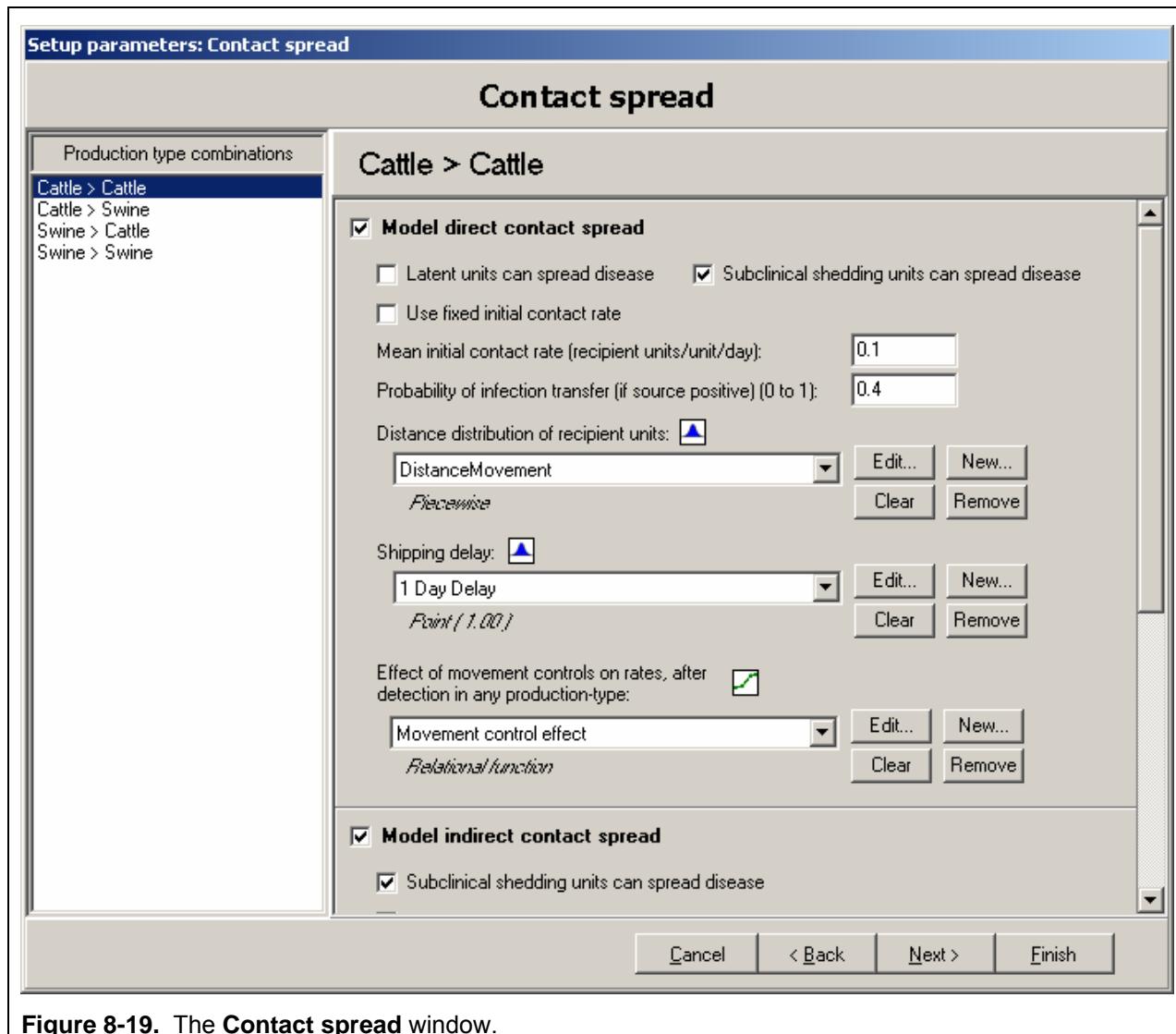


Figure 8-19. The Contact spread window.

8.6.3.1. Direct versus indirect contact spread

Direct contact involves the transfer of one or more animals from one herd (unit) to another. Indirect contact typically involves the movement of people, materials, vehicles, equipment, animal products, *etc.*, among units. The parameters for direct and indirect contact are similar, but independent from one another.

8.6.3.2. Contact rates

A contact rate is used to indicate the average number of contacts (shipments of animals in the case of direct contact, or movements of people, equipment, *etc.*, for indirect contact) that are generated by each herd on each day. Consider the following (very hypothetical) example. Suppose that a single dairy herd typically generates six indirect contacts with other dairy herds

8. Setting up a scenario

The **Scenario parameters** menu

every day: these might be movements of dairy farm workers, milk trucks, or veterinarians. This particular herd would have a contact rate of six other herds per day. If the contact pattern is similar for all other dairy herds, that is, for all units of production type “Dairy”, the contact rate would be six recipient units per source unit per day.

By contrast, suppose that our first dairy herd was unusually active, and that we have a second dairy herd that typically generates only two contacts per day. If these two herds represent the entire “Dairy” production type, we would recalculate the contact rate by taking an average. The mean contact rate for units of production type “Dairy” would be four recipient units per source unit per day.

Some other movements might be relatively rare: suppose that our “average” dairy herd ships animals to another herd only once every 20 days. The mean direct contact rate for production type “Dairy”, then, would be 0.05 recipient herds per source herd per day.

NAADSM typically generates contacts among herds on each day of a simulation based on this mean daily contact rate. Specifically, this mean contact rate is used to define a Poisson distribution (see Appendix D). The actual number of contacts generated on a particular day is then drawn from this distribution. The actual number of contacts on any given day may be far more or far less than the mean contact rate, but over time, the average number of contacts generated on each day will closely approximate the specified mean contact rate.

After the detection of disease, a common control measure would be the implementation of movement restrictions to limit the number of contacts between infectious and susceptible herds. The contact rate required by *NAADSM* is the initial, or unrestricted, mean contact rate. Movement restrictions are modeled in *NAADSM* by the reduction over time of the initial mean contact rate. Movement control effects are described in more detail in Appendix A.

NOTE: Fixed initial contact rates

NAADSM also offers the option of specifying a fixed number of daily contacts. This feature may be particularly useful for testing purposes: if you want to generate exactly five contacts for each source herd, you can provide a fixed contact rate of five recipient units per source unit per day. Like the mean initial contact rate, a fixed initial contact rate is affected by movement controls.

Note that the number of fixed contacts generated on each day will always be an integer: “ $\frac{1}{2}$ of a contact” is not possible. In the current version of *NAADSM*, it is not possible to use a fixed contact rate to generate exactly one contact every two days.

For virtually all analytical or experimental purposes, the use of a mean contact rate and the Poisson distribution it defines is better than the use of a fixed contact rate.

8.6.3.3. Setting contact spread parameters

Select a production type combination from the list on the left side of the **Contact spread** window (Figure 8-19), and then adjust parameters as described below.

8.6.3.3.1. Direct contact spread parameters

Model direct contact spread: check this box to include disease spread by direct contact in this production type combination in your simulation. If this box is checked, the remaining options for direct contact will appear.

Latent units can spread disease: check this box if direct contact with a latently infected herd can spread disease.

Subclinical shedding units can spread disease: check this box if direct contact with a subclinically affected herd that is shedding the disease agent can spread disease.

Use fixed initial contact rate: check this box if you wish to specify a fixed initial contact rate (see the note in Section 8.6.3.2). Most of the time, leave this box unchecked.

Mean initial contact rate (recipient units/unit/day): enter the unrestricted contact rate, as described in Section 8.6.3.2. (If **Use fixed initial contact rate** is checked, this option will be called **Fixed initial contact rate (recipient units/unit/day)**).

Probability of infection transfer (if source positive): if a herd of the source production type is infectious, and it directly contacts a herd of the recipient production type, how likely is it that the contact herd will become infected? Enter a probability between 0 (0%) and 1 (100%).

Distance distribution of recipient units: this variable is a probability density function (*pdf*) defining the distance between a herd of the source production type, and its direct contacts with herds of the recipient production type. Select or create a function with the *pdf* selection box (see Section 6.1.5).

Effect of movement controls on rates, after detection in any production type: after detection of disease, movement of livestock may be restricted. Use this variable to simulate the effect of movement restrictions on the number of contacts between units. This variable describes animal movement between these production types as a proportion of pre-outbreak animal movement, *i.e.*, direct contact rate. The variable is a relational function (see Section 6.1.6) of time (in days) since disease was detected (see Section 8.7). Select or create a function with the *rel* selection box (see Section 6.1.6).

Shipping delay: this variable is a probability density function (*pdf*) defining the length of time (in days) that it takes for each contact initiated by a herd of the source production type, to reach herds of the recipient production type. Select or create a function with the *pdf* selection box (see Section 6.1.5).

WARNING: Shipping delay

The behavior of the shipping delay parameter is not well defined in the current version of NAADSM, and it is unrelated to the shipping distance. It is strongly recommended that point functions of 0 or 1 day (see Appendix D) be used for the shipping delay. It is possible to use a more complex distribution, but results may be somewhat unpredictable.

8. Setting up a scenario
The **Scenario parameters** menu

8.6.3.3.2. Indirect contact spread parameters

Model indirect contact spread: check this box to include disease spread by indirect contact in this production type combination in your simulation (see Section 8.6.3.1). If this box is checked, the remaining options for indirect contact will appear.

All other parameters for indirect contact spread are specified in the same was as those for direct contact spread (see Section 8.6.3.3.1), with one exception: latent units cannot spread disease by indirect contact.

8.6.4. Airborne spread

NAADSM offers the option of simulating airborne transmission of disease, *e.g.*, by dispersal of virus particles or other causal agents. The probability of airborne disease spread is related to the distance between a source and recipient herd, the prevailing wind current direction, and the direction from a source to a recipient.

8.6.4.1. Linear versus exponential decline in probability of spread

The probability of infection decreases with greater distance between a potential source and a potential recipient herd. *NAADSM* offers two algorithms (linear and exponential) for modeling this decline. In both cases, the size of the source and recipient herd influences the probability of spread: a large source herd is more likely to transmit disease by airborne spread; likewise, a large recipient herd is more likely to become infected by airborne spread.

Both algorithms require that the user to specify the probability of spread between two herds (units) of average size located 1 kilometer apart from one another. This probability is then adjusted by the selected algorithm to simulate the probability of spread between herds that are closer than 1 km (resulting in a higher probability) or farther apart (resulting in a lower probability).

For a linear decrease in probability, a maximum distance of spread (which must be greater than 1 km) must also be specified. The probability of disease spread declines linearly until this maximum distance is reached, where the probability drops to zero. The formula used to calculate the probability of spread at a particular distance with linear drop-off is as follows:

$$\begin{aligned} & (\text{Probability of infection at 1 km}) \times \\ & [(\text{Maximum distance of spread}) - (\text{Distance between herds})] / [(\text{Maximum distance of spread}) - 1] \\ & \times (\text{Adjustment factor for size of source herd}) \times (\text{Adjustment factor for size of recipient herd}) \end{aligned}$$

Please see Appendix A for details regarding the size adjustment factors.

For an exponential decrease, it is not necessary to specify this maximum distance: the maximum distance is determined based on the distance at which the probability of spread drops to a value very close to zero. The formula used to calculate to probability of spread at a particular distance with exponential drop-off is:

8. Setting up a scenario The **Scenario parameters** menu

$$(Probability\ of\ infection\ at\ 1\ km)^{(Distance\ between\ herds)} \times (Adjustment\ factor\ for\ size\ of\ source\ herd) \times (Adjustment\ factor\ for\ size\ of\ recipient\ herd)$$

Figure 8-20 shows the decline in probability of disease spread by airborne transmission between two herds in the sample US Midwest database. For these calculations, the source herd has 367 animals, the recipient herd has 35 animals, and the probability of spread between “average” herds 1 km apart is 0.02. For linear decrease, the distance of maximum spread is set to 3 km. Note that the two algorithms result in dramatically different probabilities of spread, particularly for units located less than 1 km apart.

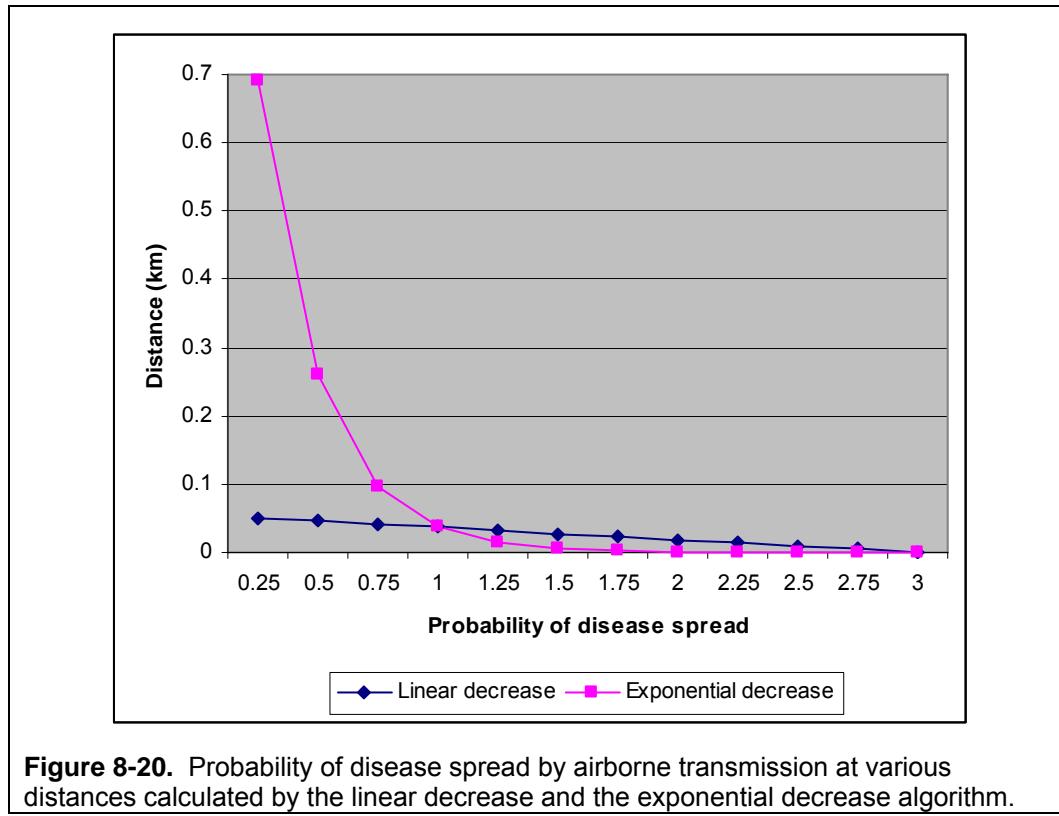


Figure 8-20. Probability of disease spread by airborne transmission at various distances calculated by the linear decrease and the exponential decrease algorithm.

8.6.4.2. The **Airborne spread** window

If airborne spread was selected in the **Spread options** window (Section 8.6.1), the **Airborne spread** window will be displayed (Figure 8-21). Use this window to modify parameters describing airborne spread of disease for each production type combination.

Model airborne spread between these production types: check this box to include airborne spread in this production type combination in your simulation.

Probability of spread/contg. day, at 1 km, average unit sizes: if you had one herd of each production type in your combination, and these herds were of average size and located 1 kilometer apart, and the source herd was infected, what is the probability that the recipient herd

8. Setting up a scenario

The **Scenario parameters** menu

would become infected within one day of the source herd becoming contagious? Enter a probability between 0 (0%) and 1 (100%).

Maximum distance spread under these conditions: if a herd was infected and shedding disease, how far from that herd can disease be spread via the air? The value must be at least one kilometer. (Recall from the previous section that this parameter applies only to the linear algorithm: this option will not be displayed if the exponential algorithm is selected.)

Wind direction: these parameters describe the directions in which disease can spread by air. Allowable range of wind direction is described by degrees (range: 0 to 360) The circle gives a visual representation of the range you specify: blue indicates wind, and gray indicates no wind. A range from 0 to 360 will color the entire circle blue. Smaller ranges (*e.g.*, 315 to 45, as shown in Figure 8-21) will restrict wind direction and thus leave part of the circle gray.

Airborne transport delay: this variable is a probability density function (*pdf*) defining the length of time (in days) that it takes for disease agents (*e.g.* virus particles) from a herd of the source production type to reach herds of the recipient production type. Select or create a function with the *pdf* selection box (see Section 6.1.5).

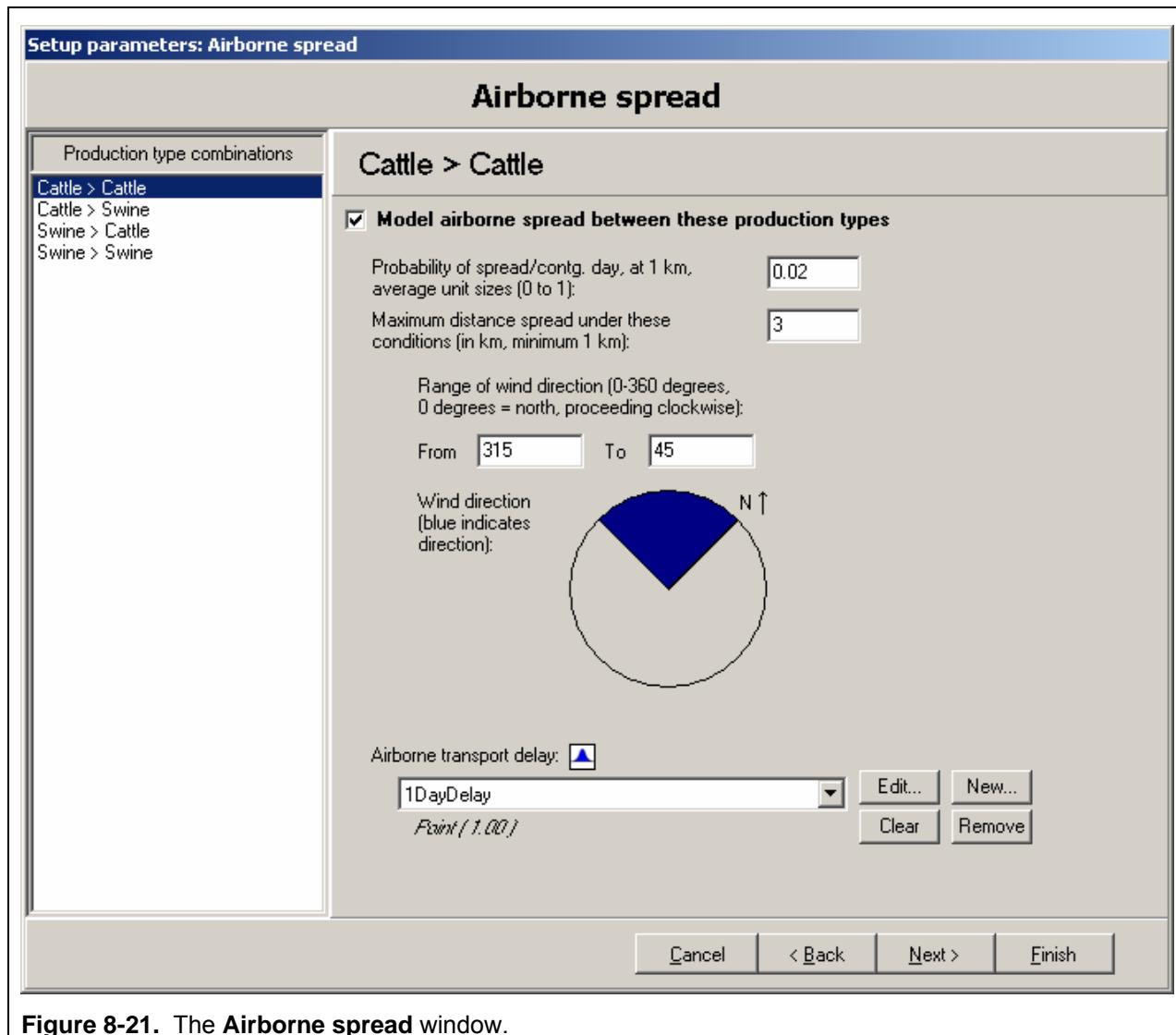


Figure 8-21. The Airborne spread window.

8.7. Modeling disease detection

8.7.1. Description of disease detection

Disease detection in *NAADSM* refers to the identification and reporting of infected units based on the appearance of clinical signs. Once disease has been detected, various control measures, such as movement restriction (Section 8.6.3.2), destruction (Section 8.9) or vaccination (Section 8.10) can be implemented. Detection parameters are given separately for each production type, to account for the possibility that signs of disease may be more obvious in some production types than in others.

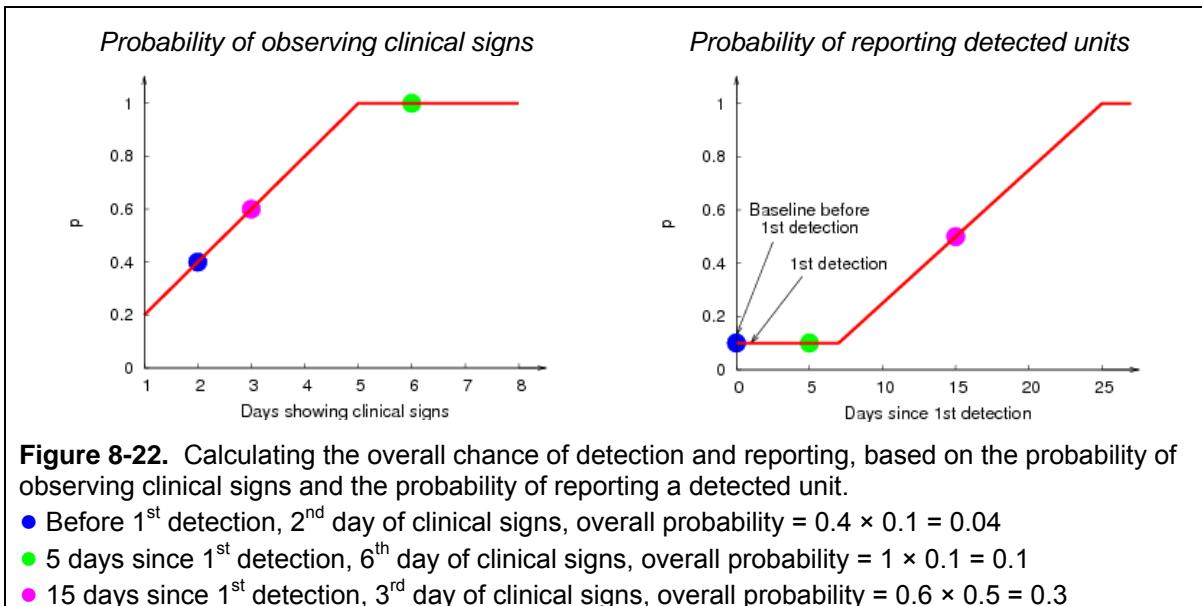
Two probabilities affect the overall chance that an infected herd (unit) will be detected: the probability of actually observing clinical signs in a herd, and the probability that a herd will

8. Setting up a scenario

The **Scenario parameters** menu

be reported once clinical signs have been observed. The overall chance of detection is the product of these two probabilities.

Both probabilities are given as relational functions (see Section 6.1.6), to account for changes to the probability of observing clinical signs and the probability of reporting over time during the course of an outbreak. For example, the probability of observing clinical signs might increase over time, as more animals in a unit become ill. Similarly, as public awareness of an outbreak increases over time, the probability of reporting infected units might change. Figure 8-22 demonstrates the interaction of these two relational functions.



NOTE: Baseline probability of reporting disease

The probability of reporting detected units is based on the number of days that have passed since the first unit was detected: for example, Figure 8-22 demonstrates a situation where the probability of reporting increases over time as public awareness of an outbreak increases. This is dependent upon an initial detection/report actually taking place.

If the initial or baseline probability of reporting is 0%, infected units will never be reported. Consequently, “detection” in the context used by NAADSM can never occur, and control measures can never be implemented. NAADSM will warn you if you have a baseline probability of reporting of 0%.

8.7.2. Setting disease detection parameters

Two windows are used to set disease detection parameters. These windows are displayed using the commands on the **Scenario parameters → Detection** menu (Figure 8-23). Depending on your selection in the **Detection options** window, the **Production type settings for detection** menu command may be disabled.

8. Setting up a scenario
The **Scenario parameters** menu



Figure 8-23. Commands of the **Scenario parameters→Detection** menu.

The **Detection options** window asks the question **Would you like to include DISEASE DETECTION in simulation runs?** If you select **Yes, include detection**, the **Production type settings for detection** menu command and **Detection** wizard screen will be enabled (Figure 8-24).

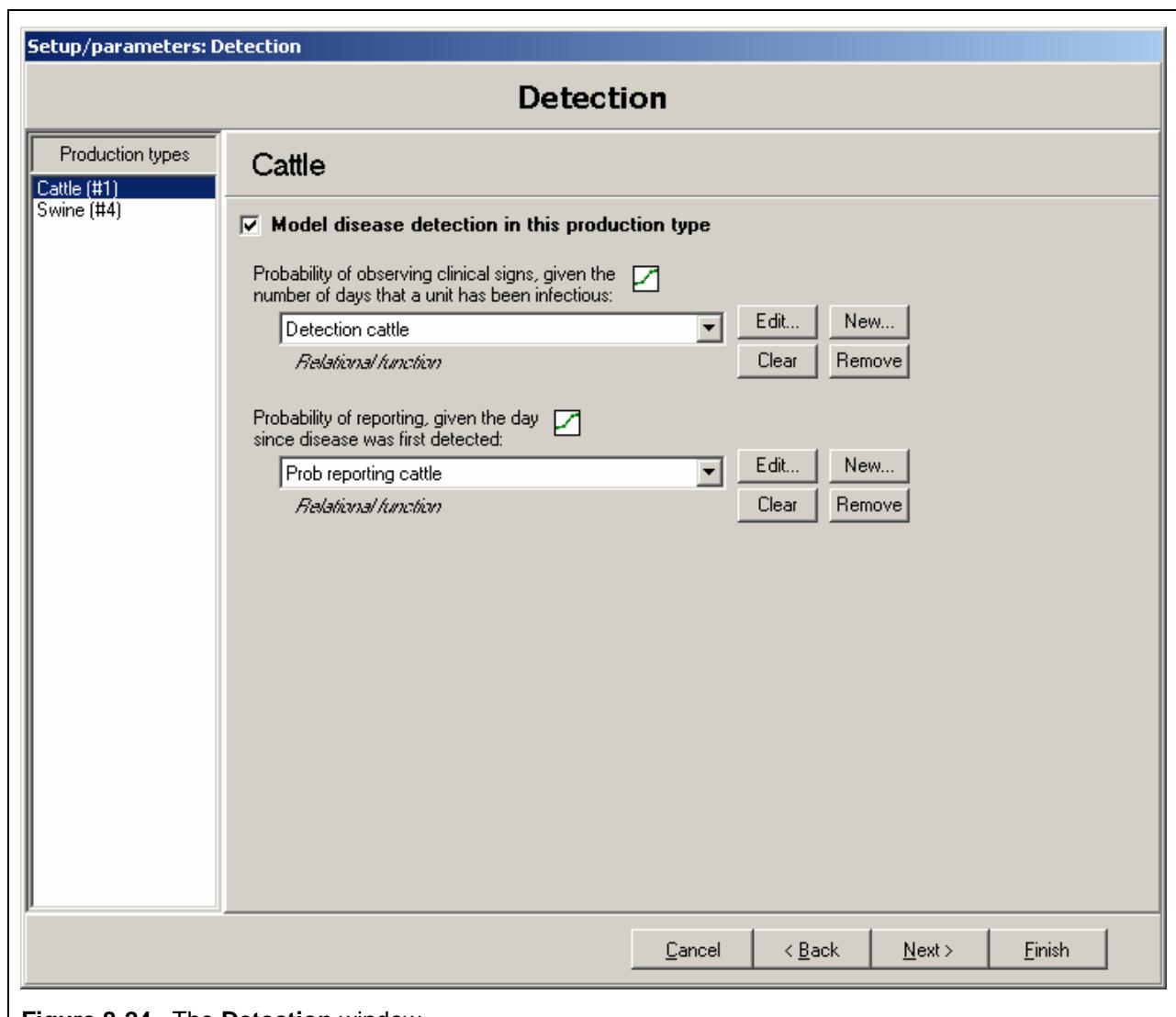


Figure 8-24. The **Detection** window.

Model disease detection in this production type: check this box to model disease detection for this type. If checked, the remaining parameters will displayed:

8. Setting up a scenario

The **Scenario parameters** menu

Probability of observing clinical signs, given the number of days that a unit has been infectious: see Section 8.7.1. The variable is a relational function (see Section 6.1.6) of time (in days) that a unit has been infectious. Select or create a function with the *rel* selection box (see Section 6.1.6).

Probability of reporting, given the day since disease was first detected: see Section 8.7.1. The variable is a relational function (see Section 6.1.6) of time (in days) since disease was detected. Select or create a function with the *rel* selection box (see Section 6.1.6).

8.8. Modeling surveillance

8.8.1. Description of surveillance

Surveillance refers to the process of identifying units (herds) at high risk for disease based upon exposure or (potentially) proximity to infected, detected units. Units identified by surveillance will be quarantined and thus can no longer spread disease by direct contact (see section 8.6.3). In the current version of NAADSM, surveillance does not affect disease detection: that is, units subject to surveillance which become infected are no more likely to be detected than other units of the same production type.

When a diseased unit is detected, tracing may be carried out to attempt to identify potential sources of the infection. Any unit that had prior contact with the diseased unit within a specified time period may be identified by trace-back investigations and quarantined. Optionally, units identified by trace surveillance may be preemptively destroyed (see Section 8.9).

8.8.2. Setting surveillance parameters

Two windows are used to set surveillance parameters. These windows are displayed using the commands on the **Scenario parameters → Surveillance** menu (Figure 8-25). Depending on your selection in the **Global surveillance options** window, the **Production type settings for surveillance** menu command may be disabled.

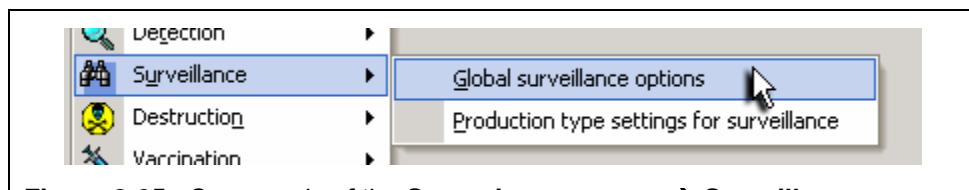


Figure 8-25. Commands of the **Scenario parameters → Surveillance** menu.

If the **Use surveillance for some or all production types** box is checked in the **Global surveillance options** window, then the **Production type settings for surveillance** menu command and **Surveillance** wizard screen will be enabled (Figure 8-26).

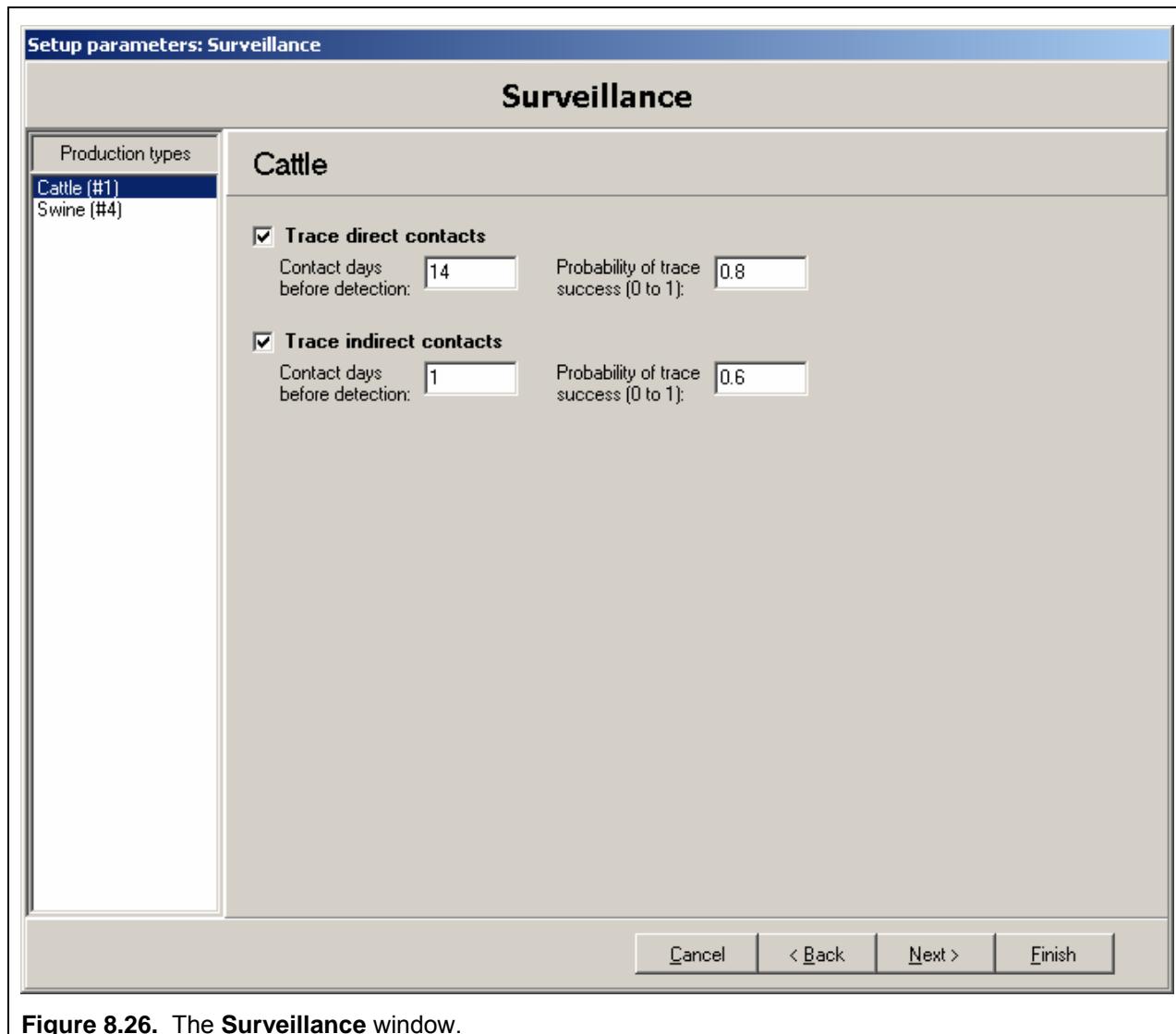


Figure 8.26. The Surveillance window.

Trace direct contacts: check this box to trace units which have had prior direct contact with an infected unit.

Contact days before detection: how far back in time should contacts be traced in an attempt to find the source of infection? Enter an integer value.

Probability of trace success: the probability that a contact will be successfully identified by tracing. Enter a value between 0 (0%) and 1 (100%).

Trace indirect contacts: check this box to trace units which have had prior indirect contact with an infected unit. The remaining parameters for tracing of indirect contact are equivalent to those described for tracing of direct contacts.

8. Setting up a scenario
- The **Scenario parameters** menu

8.9. Modeling destruction for disease control

Destruction (stamping out or depopulation) may be modeled as a method of disease control. Units (herds) with detected infection may be destroyed. Preemptive destruction of uninfected or undetected herds may also be carried out, based on exposure or proximity to infected, detected units.

A series of three windows is used to set destruction parameters. These windows are displayed using the commands on the **Scenario parameters → Destruction** menu (Figure 8-27). Depending on your selections in the **Global destruction options** window, the remaining menu commands may be disabled.

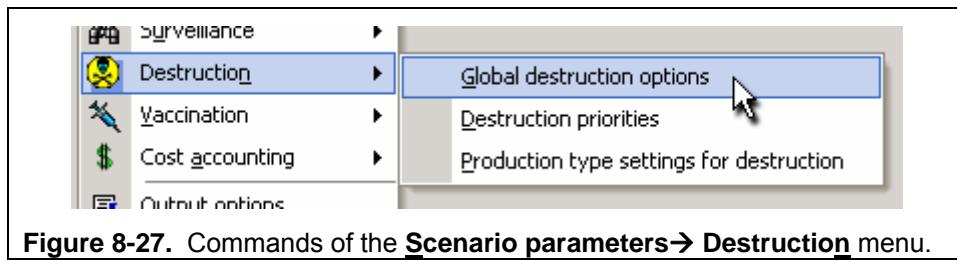


Figure 8-27. Commands of the **Scenario parameters→ Destruction** menu.

8.9.1. Global destruction options

Some settings affect destruction regardless of the production types involved: these settings are made in the **Global destruction options** window (Figure 8-28), and are described below.

Use destruction for disease control for some or all production types: if this box is checked, destruction will be modeled. Otherwise, destruction will not be modeled, and all remaining destruction options will be disabled.

Delay before implementing destruction program: after detection of the first infected herd, how long is it before a destruction program can be initiated? Specify a number of days, or 0 if a destruction program can begin immediately.

Destruction capacity: it may take time for a destruction program to ramp up to full capacity. Destruction capacity is modeled as relational function of the number of units (of any production type and of any size) that can be destroyed per day versus the number of days since the first detection. Select or create a function with the *rel* selection box (see Section 6.1.6).

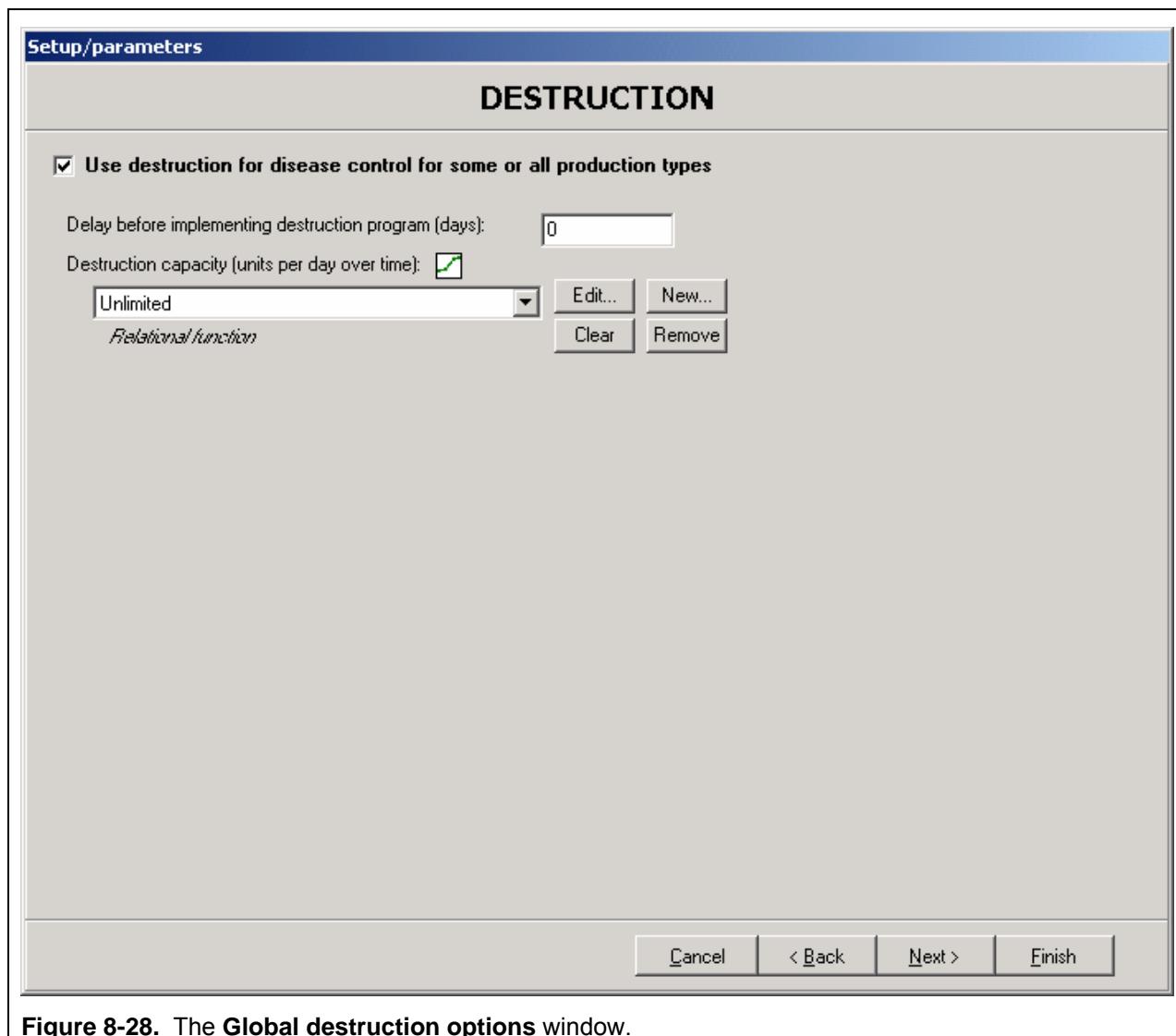


Figure 8-28. The **Global destruction options** window.

8.9.2. Destruction priorities

If destruction capacity is limited, the priority with which units are destroyed may be depend on the reason for destruction (*e.g.*, units known to be infected might be destroyed before any units are preemptively destroyed), production type (*e.g.*, animals of some species might be more likely to spread disease and thus should be destroyed first), or the length of time that a herd has been in queue for destruction.

8. Setting up a scenario
- The **Scenario parameters** menu

8.9.2.1. Setting destruction priorities

The **Destruction priorities** window (Figure 8-29) allows you to specify the destruction priorities that you would like to model.

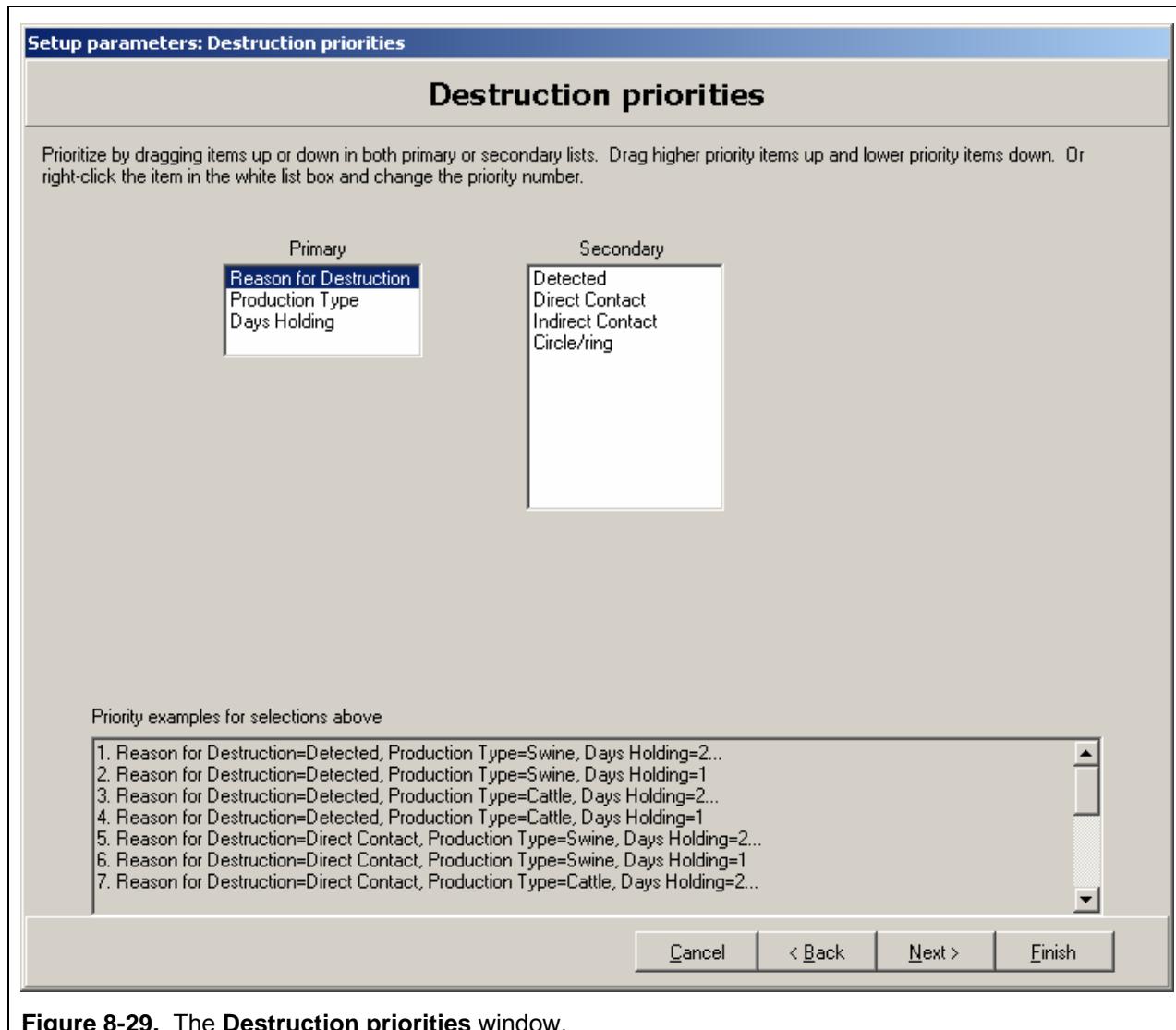


Figure 8-29. The **Destruction priorities** window.

Primary priorities: this frame lists broad categories for prioritizing destruction. Items listed include **Production type**, **Days holding**, and **Reason for Destruction**. **Days holding** is the number of days the herd has been listed for destruction. Reasons for destruction include all of the different reasons that a herd could have marked for destruction: detection of disease (Section 8.7); identification of a direct or an indirect contact with a detected unit by trace-back investigation (Section 8.8); or proximity to a detected unit within a specified radius (circle/ring destruction: see section 8.9.3). Change priority order by dragging items up or down in the list.

Secondary priorities: this frame lists the options within the category selected under primary priorities. Change priority order by dragging items up or down in the list.

Priority examples for selections above: this frame shows the order in which herds will be destroyed, based on your sequence of primary and secondary priorities. More detailed examples are presented in Section 8.9.2.2.

8.9.2.2. Examples of destruction priorities

Consider the following examples, using these four units which have been designated for destruction:

- Unit A. Cattle herd, detected infection, holding for 3 days
- Unit B. Cattle herd, indirect contact, holding for 5 days
- Unit C. Swine herd, direct contact, holding for 1 day
- Unit D. Swine herd, within circle/ring, holding for 5 days

Example 1:

With the following destruction priorities:

Days holding > production type (swine > cattle) > destruction reason (detected > direct > indirect > circle/ring)

The four herds are destroyed in the following order:

D, B, A, C

Example 2:

Priorities: production type (cattle > swine) > destruction reason (detected > direct > indirect > circle/ring) > days holding:

Destruction order: A, B, C, D

Example 3:

Priorities: production type (cattle > swine) > days holding > destruction reason (detected > direct > indirect > circle/ring):

Destruction order: B, A, D, C

Example 4:

Priorities: destruction reason (detected > circle/ring > direct > indirect) > production type (cattle > swine) > days holding:

Destruction order: A, D, C, B

8. Setting up a scenario
The **Scenario parameters** menu

8.9.3. Production type settings for destruction

The **Destruction** window (Figure 8-30) is used to set production type-specific parameters for modeling destruction.

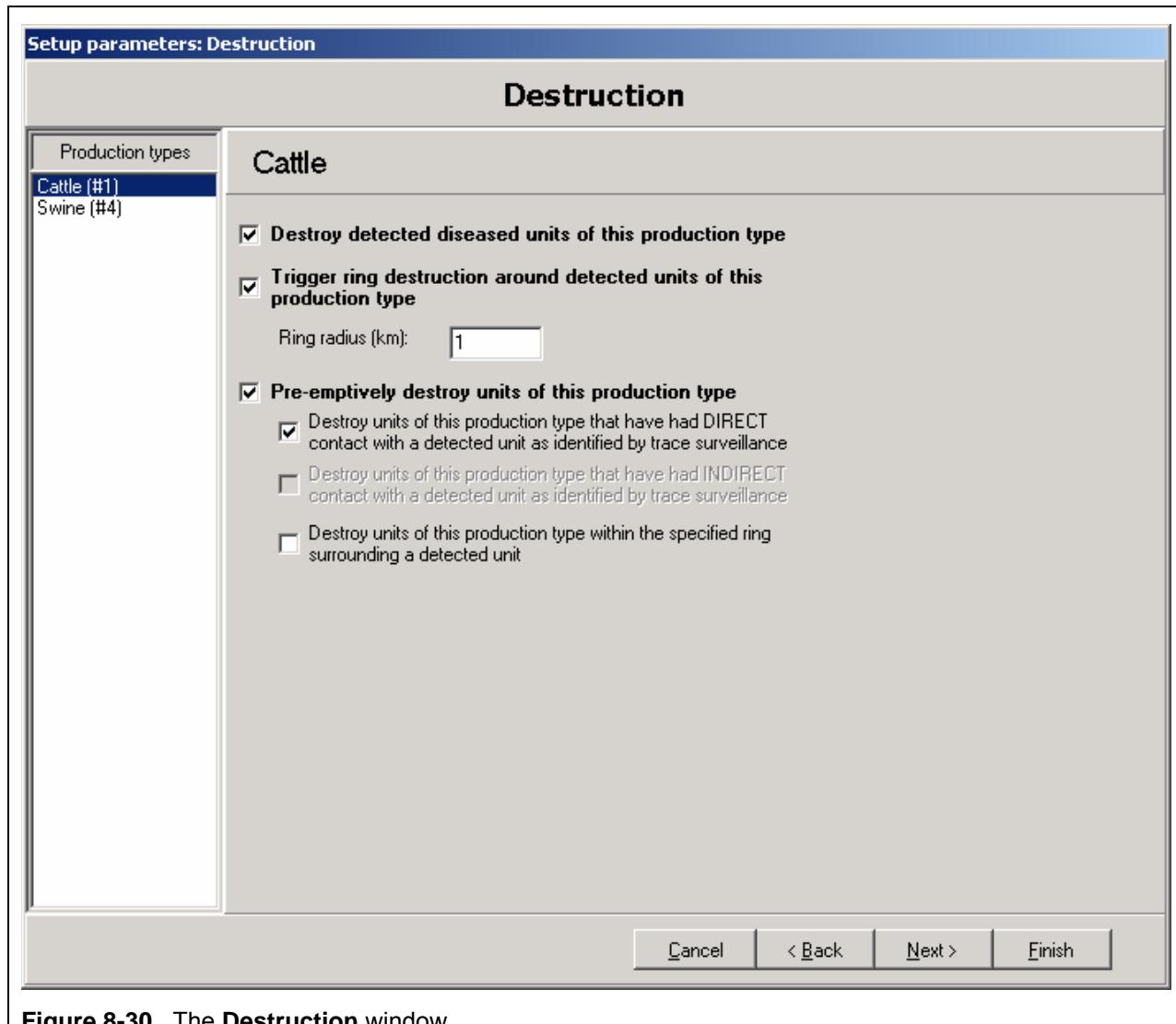


Figure 8-30. The **Destruction** window.

The parameters in this window address two distinct aspects of destruction as a disease control measure: 1) should units (herds) of a particular production type be destroyed, and 2) should disease detection in a unit of a particular production type lead to the destruction of other units? In the first case, units are the *targets* of destruction; in the second, they are the *triggers* for destruction. A production type may be a target, a trigger, a target and a trigger, or neither, depending on the scenario that you want to model. Please keep this distinction in mind when setting the following parameters.

8. Setting up a scenario

The **Scenario parameters** menu

Destroy detected diseased units of this production type: checking this box will cause units of this production type to be marked for destruction when disease is detected (*i.e.*, units of this type will be targets of destruction when detected).

Trigger ring destruction around detected units of this production type: checking this box may cause *other* units located around an infected unit of this production type to be marked for destruction when the infected unit is detected. Those other units will be marked for destruction only if the production types of those units are subject to preemptive destruction (see below). If this box is checked, you must also specify a radius (in kilometers) of the destruction ring.

Preemptively destroy units of this production type: check this box if units of a particular production type should be subject to preemptive destruction (*i.e.*, units of this type will be targets of preemptive destruction). If this box is checked, three more boxes may be activated. You may choose to preemptively destroy units that have had either direct or indirect contact with a detected unit as determined by tracing (these two options will be available only if surveillance is being carried out for the specified type: see Section 8.8). You may also choose to preemptively destroy units that are located within a destruction ring (see above).

8.10. Modeling vaccination

A vaccination campaign may also be modeled as a method of disease control. This consists of vaccinating units (herds) within a specified distance of the detected units, in circles or rings around detected units. After vaccination, a herd becomes immune from infection for some specified period of time (Section 8.10.3). In the current version of *NAADSM*, vaccination is always 100% effective and conveys complete immunity to the vaccinated herd.

A series of three windows is used to set vaccination parameters. These windows are displayed using the commands on the **Scenario parameters → Vaccination** menu (Figure 8-31). Depending on your selections in the **Global vaccination options** window, the remaining menu commands may be disabled.

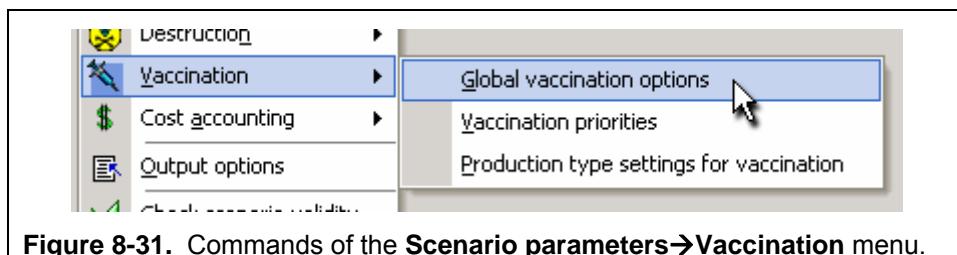


Figure 8-31. Commands of the **Scenario parameters→Vaccination** menu.

8.10.1. Global vaccination options

Some settings affect vaccination regardless of the production types involved: these settings are made in the **Global vaccination options** window (Figure 8-32), and are described below.

8. Setting up a scenario
The **Scenario parameters** menu

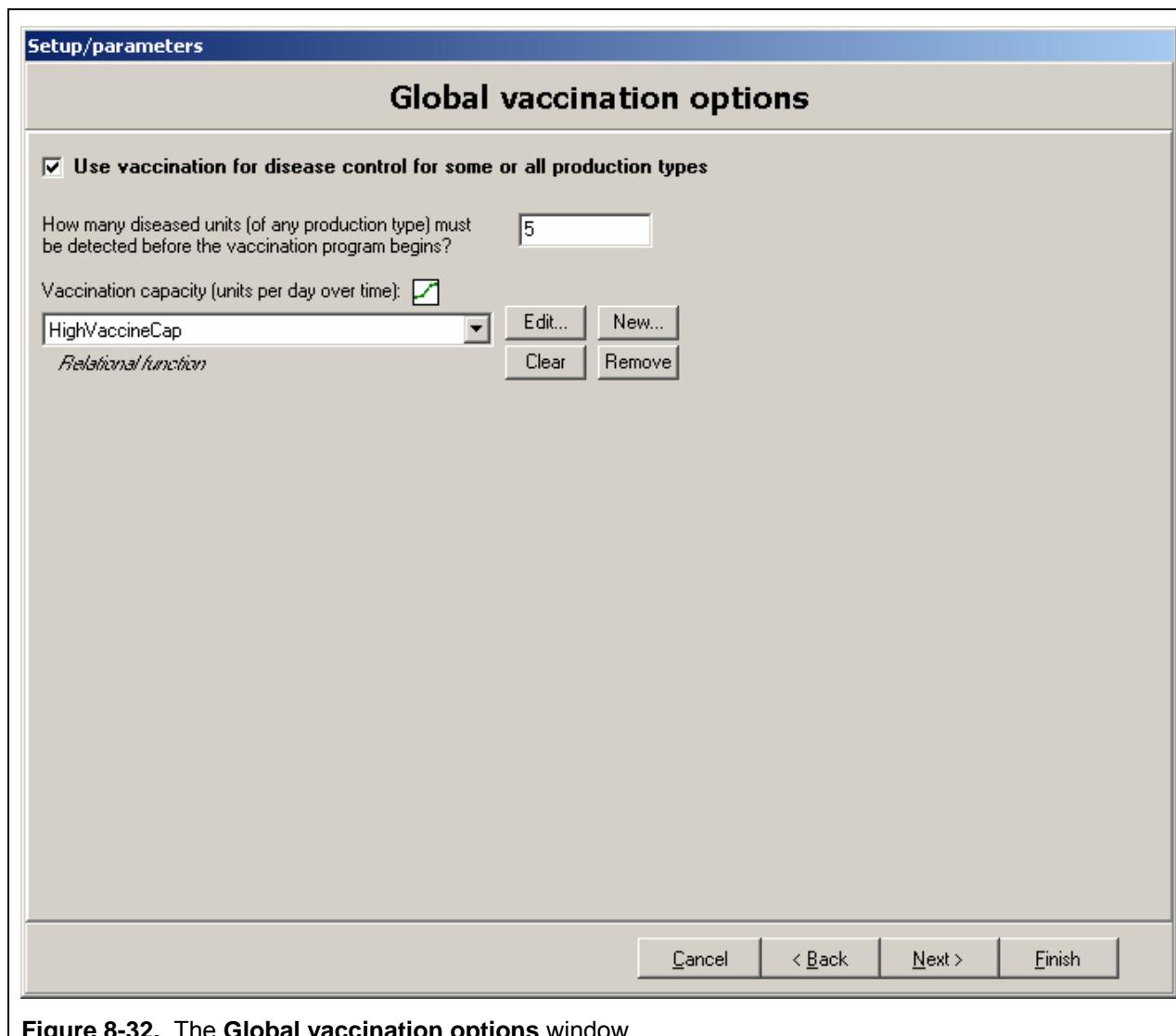


Figure 8-32. The **Global vaccination options** window.

Use vaccination for disease control for some or all production types: if this box is checked, vaccination will be modeled. Otherwise, vaccination will not be modeled, and all remaining vaccination options will be disabled.

How many diseased units (of any production type) must be detected before the vaccination program begins?: how many detections must occur before the vaccination campaign is initiated? Specify a number of units, or 0 if a destruction program can begin immediately. [Note that this delay before initiating vaccination is slightly different from the delay before initiating destruction (Section 8.9.1): vaccination starts after the specified number of detections, while destruction starts the specified number of days after the first detection.]

Vaccination capacity: it may take time for a vaccination program to ramp up to full capacity. Capacity is modeled as relational function of the number of units (of any production type and of any size) that can be vaccination per day versus the number of days since the first detection. Select or create a function with the *rel* selection box (see Section 6.1.6).

8.10.2. Vaccination priorities

If vaccination capacity is limited, the priority with which units are vaccinated may be depend on the production type or the length of time that a herd has been in queue for vaccination. Vaccination priorities are set in much the same way as destruction priorities (Section 8.9.2).

8.10.2.1. Setting vaccination priorities

The **Vaccination priorities** window (Figure 8-33) allows you to specify the vaccination priorities that you would like to model.

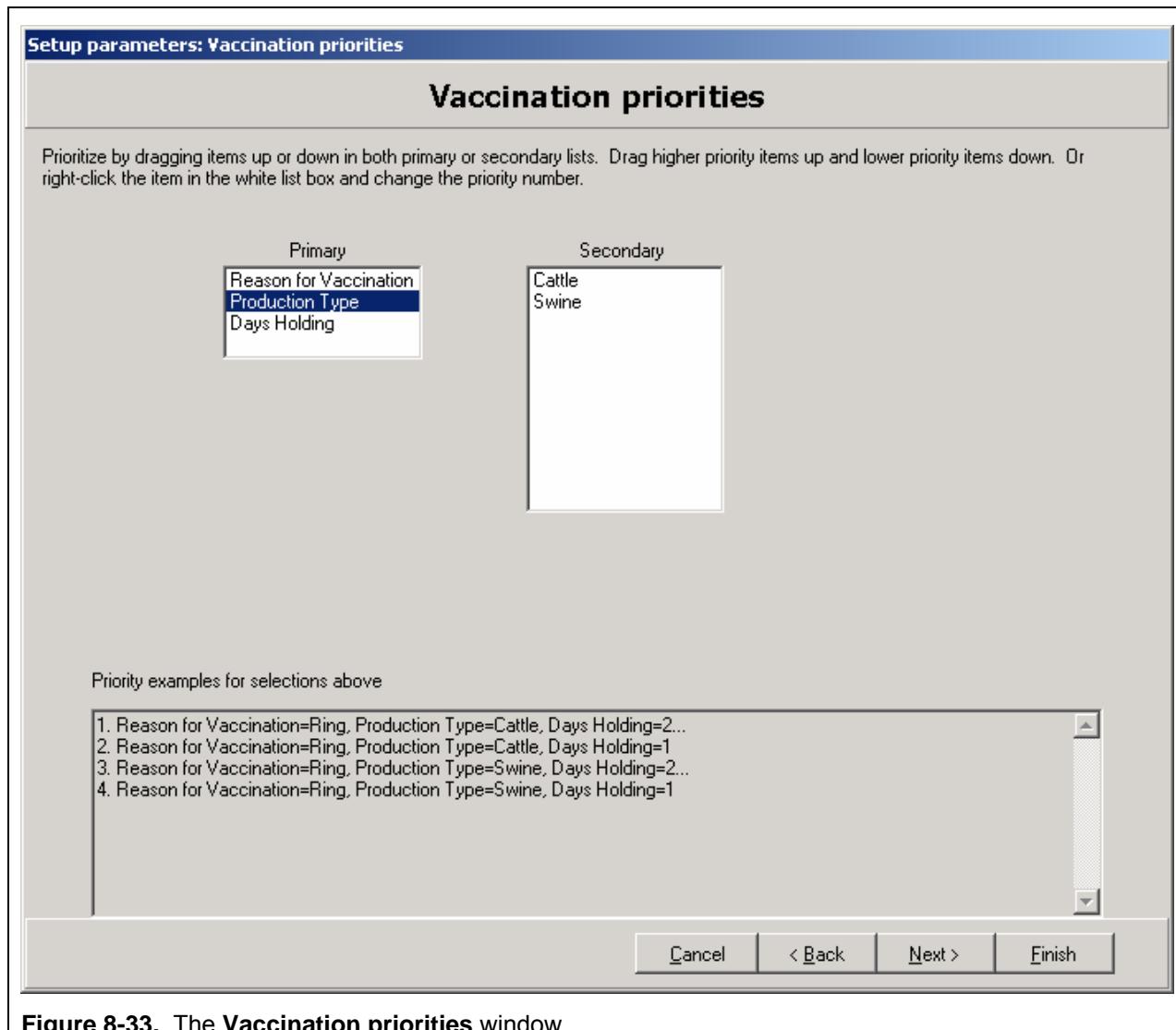


Figure 8-33. The **Vaccination priorities** window.

Primary priorities: this frame lists broad categories for prioritizing vaccination. Items listed include **Production type**, and **Days holding**. **Days holding** is the number of days the herd has been listed for vaccination. Change priority order by dragging items up or down in the list.

8. Setting up a scenario

The **Scenario parameters** menu

NOTE: Reason for vaccination

The **Vaccination priorities** window actually shows a third primary priority, called **Reason for Vaccination**. While future versions of NAADSM may include multiple reasons for vaccination, the current version models offers only one (ring vaccination). Consequently, changing the priority of **Reason for Vaccination** in the **Primary** frame has no effect.

Secondary priorities: this frame lists the options within the category selected under primary priorities. Change priority order by dragging items up or down in the list.

Priority examples for selections above: this frame shows the order in which herds will be vaccinated, based on your sequence of primary and secondary priorities.

8.10.3. Production type settings for vaccination

The **Vaccination** window (Figure 8-34) is used to set production type-specific parameters for modeling vaccination.

The parameters in this window address two distinct aspects of vaccination as a disease control measure: 1) should units (herds) of a particular production type be vaccinated, and 2) should disease detection in a unit of a particular production type lead to the vaccination of other units? In the first case, units are the *targets* of vaccination; in the second, they are the *triggers* for vaccination. A production type may be a target, a trigger, a target and a trigger, or neither, depending on the scenario that you want to model. Please keep this distinction in mind when setting the following parameters.

Vaccinate units of this production type as part of disease control efforts: check this box if you want herds of this production type to be vaccinated (*i.e.*, if units of this type should be targets of vaccination). Vaccination is 100% effective if administered when the herd is susceptible, unless the herd is infected before the vaccine takes effect (see below).

Vaccine immune period: this variable is a probability density function (*pdf*) defining the duration of immunity following vaccination. Select or create a function with the *pdf* selection box (see Section 6.1.5).

Delay in unit immunity following vaccination: this variable defines the fixed number of days following vaccination that the herd remains susceptible. If a herd is vaccinated but exposed to disease before the specified number of days has passed, the herd will become infected and enter the disease cycle (Section 8.5).

Minimum time between vaccinations: this is the number of days which must pass before a unit may be revaccinated. Once the specified number of days has passed, a unit may be revaccinated if vaccination of that unit is triggered again (see Section 8.10.3.1).

8. Setting up a scenario
The **Scenario parameters** menu

Trigger a vaccination ring upon disease detection in units of this production type:
checking this box may cause *other* units located around an infected unit of this production type to be marked for vaccination when the infected unit is detected. If this box is checked, you must also specify a radius (in kilometers) of the vaccination ring.

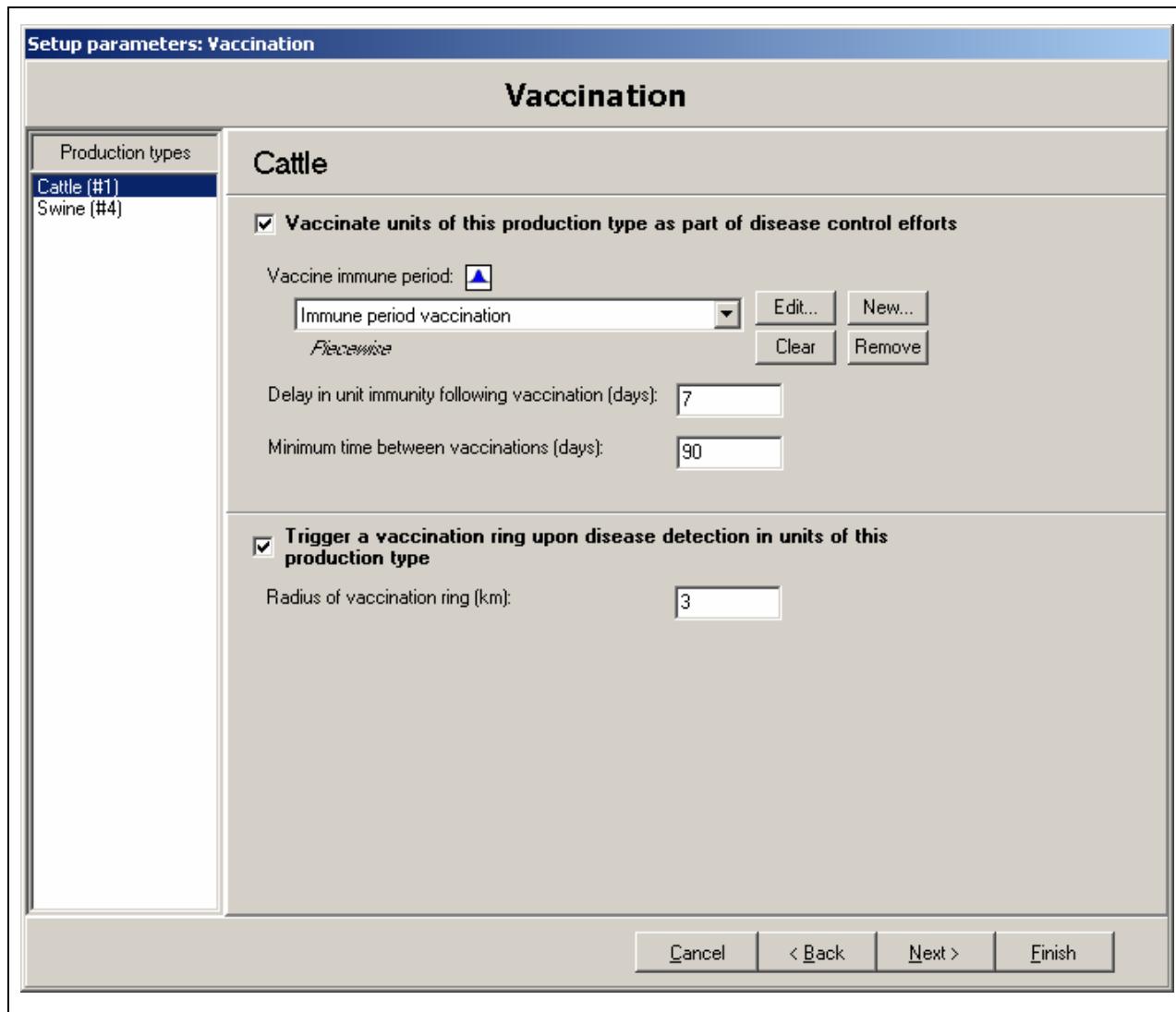


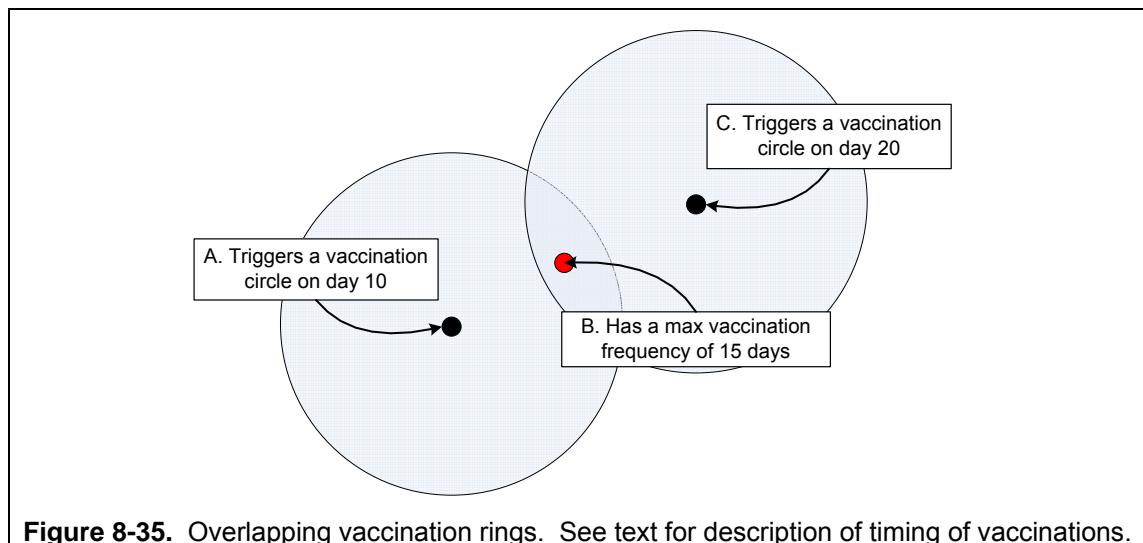
Figure 8-34. The Vaccination window.

8. Setting up a scenario
- The **Scenario parameters** menu

8.10.3.1. Minimum time between vaccinations

The minimum time between vaccinations is the number of days which must pass before a unit may be revaccinated. Once the specified number of days has passed, a unit may be revaccinated if vaccination of that unit is triggered again.

Consider the simple situation involving units *A*, *B*, and *C* as shown Figure 8-35. Disease is detected in unit *A* ten days before disease is detected in unit *C*. Both detections trigger vaccination circles as shown.



Unit *B* is within vaccination circles triggered by detection of units *A* and *C*, and will be added twice to the queue of units to be vaccinated. If there is no waiting period for vaccination (*i.e.* vaccination capacity is not reached), unit *B* will receive only one vaccination: its minimum time between vaccinations will not have been reached before it comes to the head of the queue the second time.

If vaccination capacity has been reached, unit *B* will receive two vaccinations only if the elapsed time between the first and second scheduled vaccinations exceeds the unit's minimum time between vaccinations. This subsequent vaccination resets the vaccine-immune period for unit *B*. If the elapsed time is less than the unit's maximum vaccination frequency, unit *B* will not be revaccinated.

8.11. Cost accounting

NAADSM can calculate the direct costs associated with disease control (destruction and vaccination) during an outbreak. Two windows are used to set cost parameters. These windows are displayed using the commands on the **Scenario parameters→Cost accounting** menu (Figure

8. Setting up a scenario
The **Scenario parameters** menu

8-36). Depending on your selection in the **Cost accounting options** window, the **Production type settings for cost accounting** command may be disabled.

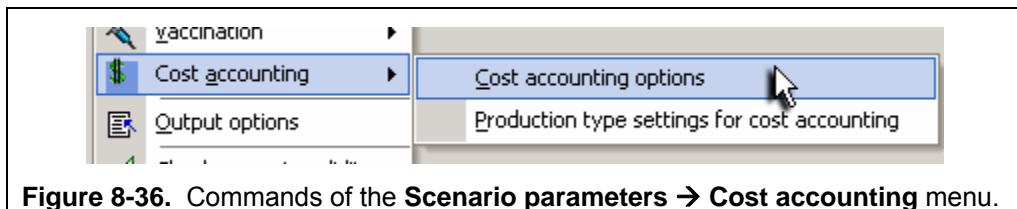


Figure 8-36. Commands of the **Scenario parameters** → **Cost accounting** menu.

The **Cost accounting options** window asks the question **Would you like to track DIRECT COSTS during simulation runs?** If you select **Yes, include cost accounting**, the **Production type settings for cost accounting** menu command and **Cost accounting** wizard screen will be enabled (Figure 8-37).

A screenshot of the 'Setup parameters: Cost accounting' window, specifically the 'Cost accounting' section for 'Cattle'. The window has a title bar 'Setup parameters: Cost accounting' and a main title 'Cost accounting'. On the left, there is a sidebar labeled 'Production types' with 'Cattle (#1)' selected. The main area shows 'Cattle' and has several input fields and checkboxes:

- Track direct costs for this production type:** A checked checkbox.
- Destruction costs:** Five input fields for 'Cost of appraisal (per unit)', 'Cost of cleaning and disinfection (per unit)', 'Indemnification (per animal)', 'Euthanasia (per animal)', and 'Carcass disposal (per animal)'. Each field starts with a '\$' sign.
- Vaccination costs:** Three input fields for 'Cost of site setup (per unit)', 'Baseline vaccination cost (per animal)', and 'Additional cost for each animal vaccinated beyond the threshold (per animal)'. Each field starts with a '\$' sign.
- Number of animals that may be vaccinated before the cost increases:** An input field containing a number.
- Additional cost for each animal vaccinated beyond the threshold (per animal):** An input field containing a number.

At the bottom of the window are four buttons: 'Cancel', '< Back', 'Next >', and 'Finish'.

Figure 8-37. The **Cost accounting** window.

8. Setting up a scenario

The **Scenario parameters** menu

Check the **Track direct costs for this production type** box if you wish to track costs for the selected type. When this box is checked, the remaining options for destruction costs (Section 8.11.1) and vaccination costs (8.11.2) are enabled.

8.11.1. Costs associated with destruction

There is a fixed cost associated with **appraisal** of each destroyed unit, regardless of the number of animals in the unit. The cost associated with **cleaning and disinfection** each unit is also fixed regardless of the number of animals in each unit.

Beyond these fixed per-unit costs, the per-animal costs for **euthanasia**, **carcass disposal**, and **indemnification** apply.

The total cost of destruction *for each unit* of a particular production type is calculated as follows:

$$\begin{aligned} & (\text{Appraisal cost} + \text{Cleaning and disinfection cost}) \\ & + [(\text{Number of animals in the unit}) \times (\text{Cost of euthanasia} + \text{Cost of indemnification} + \text{Cost of disposal})] \end{aligned}$$

The total cost of destruction *for each production type* is calculated as:

$$\begin{aligned} & (\text{Number of units destroyed}) \times (\text{Appraisal cost} + \text{Cleaning and disinfection cost}) \\ & + [(\text{Total number of animals destroyed}) \times (\text{Cost of euthanasia} + \text{Cost of indemnification} \\ & \quad + \text{Cost of disposal})] \end{aligned}$$

8.11.2. Costs associated with vaccination

There is a fixed cost associated with vaccination **site set up** for each vaccinated unit, regardless of the number of animals in the unit. The cost of vaccination of each animal in the unit is added to this fixed unit cost.

The cost of vaccination of each animal will depend on the total number of animals vaccinated. For each animal up to a specified threshold, only a **baseline vaccination** cost applies. For each animal over this threshold, an **additional cost** applies.

The total cost of vaccination *for each production type* is calculated as follows:

If the threshold is not reached:

$$\begin{aligned} & [(\text{Number of units vaccinated}) \times (\text{Cost of site setup})] \\ & + [(\text{Total number of animals vaccinated}) \times (\text{Baseline cost per animal})] \end{aligned}$$

If the threshold is reached:

$$\begin{aligned} & [(\text{Number of units vaccinated}) \times (\text{Cost of site setup})] \\ & + [(\text{Threshold level}) \times (\text{Baseline cost per animal})] \\ & + [(\text{Total number of animals vaccinated} - \text{Threshold level}) \times (\text{Baseline cost per animal} \\ & \quad + \text{Additional cost per animal})] \end{aligned}$$

8.12. Specifying output options

NAADSM will automatically save all of model outputs which are useful for analysis. The optional outputs described in this section are primarily intended for testing and troubleshooting.

Recall from Section 3 that the results of a single iteration of a stochastic model are not particularly useful: the power of stochastic models comes from having many iterations. In an *NAADSM* simulation, one iteration represents one possible outcome of a scenario. One iteration might run for 50 simulated days, while another might run for 250 simulated days. The events that occur on any particular simulated day in any particular iteration are of virtually no value by themselves. In some instances, however, particularly during the design and testing of a new scenario, it might be helpful to run a single iteration and track individual events for every day of that iteration. An examination of events on this “microscopic” level might provide some insight into how *NAADSM* works, or (perish the thought!) identify a previously undiscovered bug in the *NAADSM* application.

The options available on the **Output options** window (Figure 8-38) will allow you to track the daily events that occur in the model.

Write a plain text file containing daily states for all units: if this box is checked, a plain text file similar to the one shown in Figure 8-39 will be produced when an iteration is run. Each row in the file represents a single simulated day. Each column represents a single unit. An examination of this file will show exactly how the disease or immunity states of each unit changed over time during the iteration. Note that such a file quickly become impractical if a scenario contains more than a handful of units. If this box is checked, you must specify a valid file name for the plain text file.

8. Setting up a scenario
The **Scenario parameters** menu

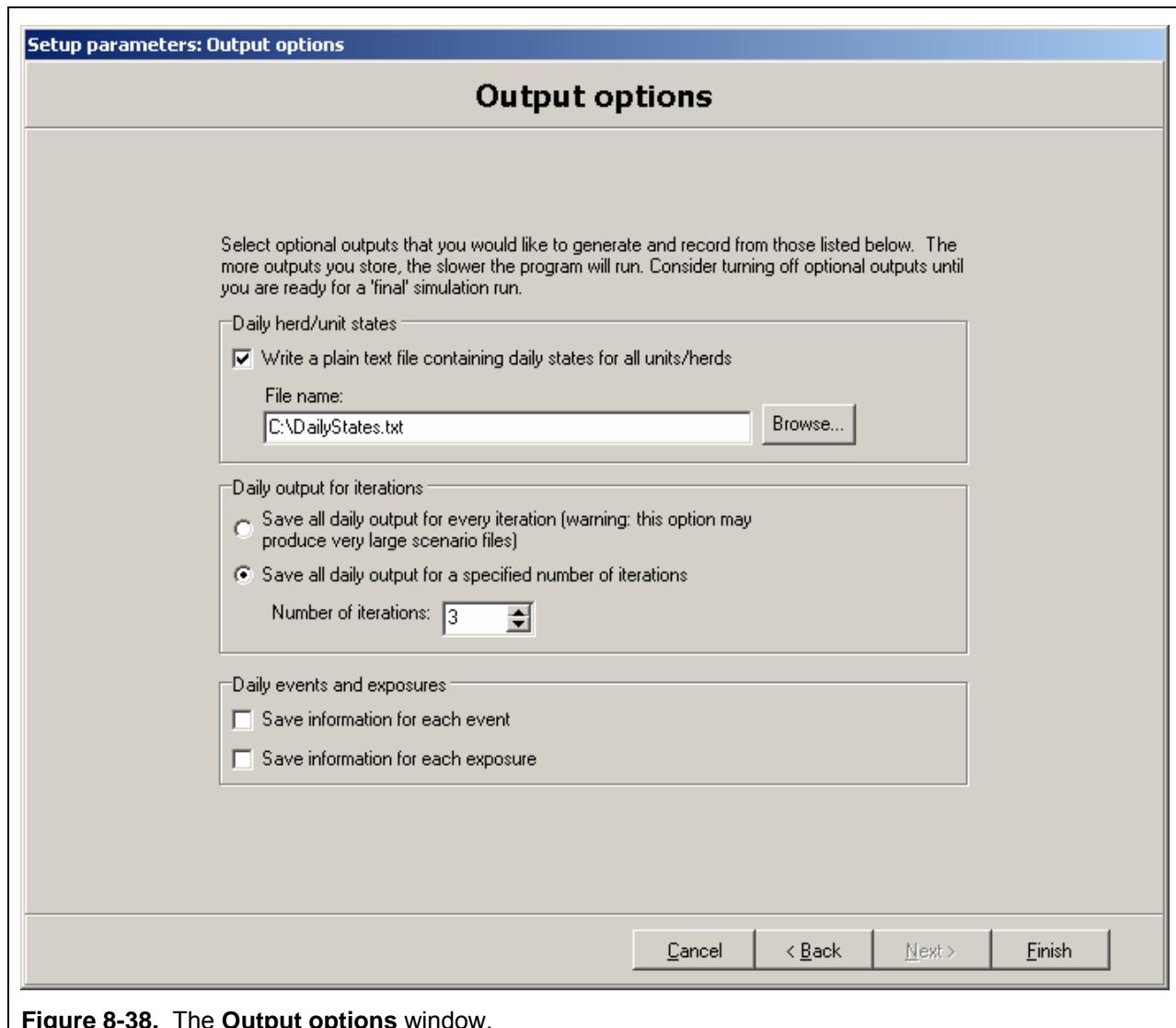


Figure 8-38. The **Output options** window.

```
Iteration 1
N N N N S S S S S L S S S S N N N
N N N N S S S S S S B S S S S N N N
N N N N S S S S S S B S S S S N N N
N N N N S S S S S S C S S S S N N N
N N N N S S S S S S C S S S L N N N
N N N N S S S S S S C S S B N N N
N N N D S S S S S D S S S C N N N
N N N D S S S S S D S S S C N N N
N N N D S S S S S D S L D N N N
N N N D S S S S S D V B D N N N
N N N D S S S S S D V C D N N N
N N N D S S S S S D V C D N N N
N N N D S S S S S D V D D N N N
N N N D S S S S S D V D D N N N
N N N D S S S S S D V D D N N N
```

Figure 8-39. A sample plain text file showing daily herd (unit) states. Disease states are indicated with the single character code shown in Table 8-2.

Daily output for iterations: by default, *NAADSM* will save daily outputs for the last three iterations for the current scenario. Ordinarily, this is more than sufficient. To save daily outputs for more iterations, you may choose to **Save all daily output for every iteration** or increase the number of iterations for which daily outputs are saved.

WARNING: Saving all daily output

If you are positive that you need to save all daily output for every iteration, be sure to have plenty of hard disk space. The *NAADSM* scenario file will be huge if you have very many iterations.

Daily events and exposures: check these boxes if you wish to save information about all daily events (*e.g.*, infections, disease state transitions, destructions, vaccinations, *etc.*) or exposures (by contact or airborne spread). Event and exposure data are discussed in more detail in Section 10.3.

8.13. Using the **Check scenario validity** command

Alert readers of this guide will have noticed by this point that a detailed *NAADSM* scenario will have many parameters. The process of setting each of these parameters presents many opportunities for mistakes to be made, or for important variables to be overlooked. Using **Scenario parameters → Check scenario validity** will help to reduce the potential for such errors. If *NAADSM* detects errors in a scenario, a detailed error message with the location and description of the problem will be displayed. If no errors are identified, you should be able to run your scenario (see Section 9).

8. Setting up a scenario

The **Scenario parameters** menu

NOTE: Attempting to run an invalid scenario

If you attempt to run a scenario (with one of the commands discussed in Section 9) when the scenario contains errors, no damage will be done. You will simply see the same error message that would have been displayed had you used the **Scenario parameters → Check scenario validity** command, and *NAADSM* will refuse to launch the simulation. Use of this command is not required, but it will save you some time.

8.14. Using the **Clear scenario output** command

Recall from Section 7 that an *NAADSM* scenario file contains all input parameters as well as all output for that scenario when it is run. If you wish you restore your scenario file to pristine condition without any output, use the **Scenario parameters → Clear scenario output** command. Input parameters will remain just as they were, but outputs will be deleted and the scenario file will be ready to run again.

WARNING: Clearing scenario output

Use the **Scenario parameters → Clear scenario output** command with care: once output has been cleared, it gone (unless you practice good computing habits and have a backup copy of all of your essential data!).

9. Launching a simulation: The Run menu

The commands of the Run menu allow you to launch a simulation. Each iteration will start with the initial population (see Section 8.4) and will proceed until the ending condition you select (see below) is met. Once all iterations are complete, an on-screen notification will be displayed, and summary output for all iterations (see Section 10.4) will be available.

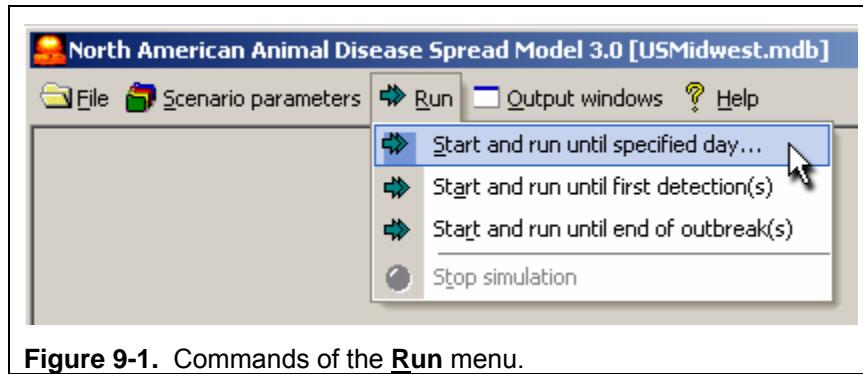


Figure 9-1. Commands of the Run menu.

Start and run until specified day...: selecting this option will stop all iterations of the simulation a specified number of days after the start of the iteration. Enter a value greater than 0, then click OK.

Start and run until first detection(s): selecting this option will stop all iterations of the simulation on the day when infection is first detected in a herd of any production type.

Start and run until end of outbreak(s): selecting this option will allow all iterations of the simulation to run until the outbreak is over, i.e. no herds remain infected, and no herds remain wait-listed for vaccination or destruction.

An outbreak is considered to have ended when there are no more latent, subclinical, or clinically infectious units left, and when all vaccination and/or destruction activities are complete.

Stop simulation: use this command to prematurely interrupt a simulation in progress. This command is available only when a simulation is in progress. If **Stop simulation** is selected, *NAADSM* will complete the simulation day currently in progress, and then halt the simulation. It may take a moment or two for *NAADSM* to complete the current day.

10. Viewing model output: The **Output windows** menu

A great deal of work goes into the creation of a scenario and the execution of a simulation. The commands of the **Output windows** menu (Figure 10-1) let us determine whether all of that effort was worthwhile. These commands activate windows which will display the day-to-day and the summary results of a single iteration (Sections 10.2 and 10.3), as well as statistics based on many iterations of a scenario (Section 10.4). The commands of the **Output windows** menu act as toggle switches: select the command once to display the designated window, and select it again to hide the window.

Nearly all of the output windows have the same capabilities and use the same toolbar: see Section 10.5 for a description of these options.

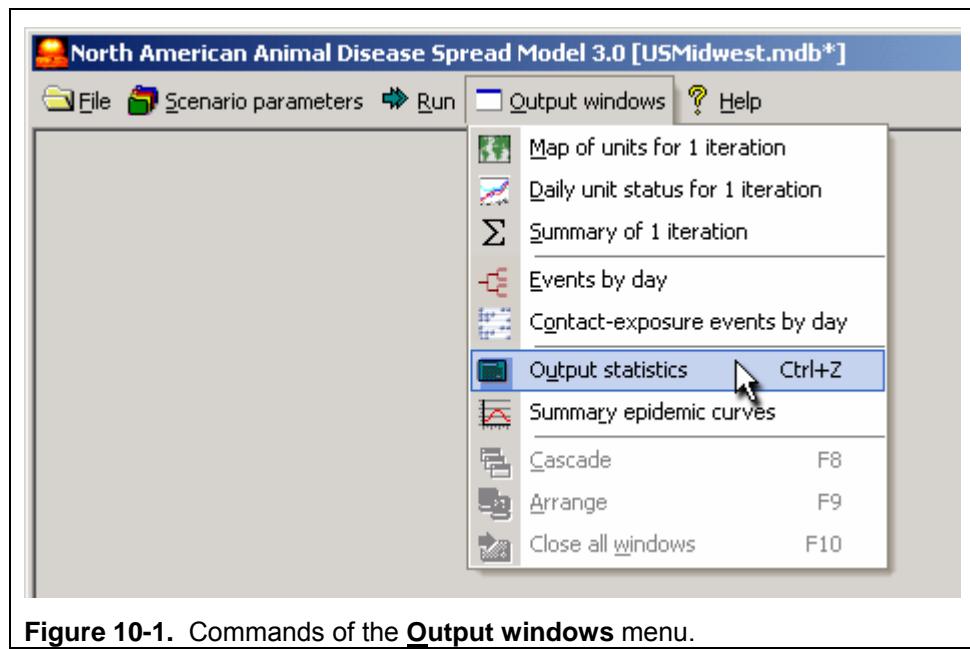


Figure 10-1. Commands of the **Output windows** menu.

NOTE: Obtaining raw data for simulation outputs

All simulation output data is stored in the *NAADSM* scenario file. In the event that raw data generated by *NAADSM* is needed for a calculation not directly supported in the *NAADSM* application, these values can be obtained directly from this *Microsoft Access*-compatible (*.mdb) file. Appendices B and E contain information that should be useful to anyone who needs direct access to outputs stored in the scenario file.

10. Viewing model output

The **Output windows** menu

10.1. “Actual” versus “apparent” events

Several of the outputs discussed below make a distinction between “actual” and “apparent” events. *NAADSM* is omnipotent, at least concerning the events that occur within a simulation. The application “knows” and records the actual disease status of every herd, the occurrence of every successful and unsuccessful exposure, and the true source of every new infection. These are examples of *actual* events, which may or may not be noticed by an observer.

The application also models the incomplete state of knowledge about an outbreak that exists while that outbreak is in progress. *Apparent* events are those that someone “on the ground” during the simulated outbreak would observe. During an outbreak, for example, herds may have been infected for quite a while before clinical signs are ever detected. Similarly, a herd may be vaccinated after it has already been infected but before it shows clinical signs, causing the vaccination to be ineffective. In cases like these, there will be a disparity between the actual and the apparent events. In an extreme example, an entire disease outbreak might occur, but if disease is never detected, the outbreak will not be apparent.

Detections, destructions, vaccinations, and the outcome of trace-back investigations are examples of apparent or observable events. Transitions between disease states, the onset of vaccine immunity, and exposures are examples of actual events.

10.2. Dynamic output windows for a single iteration

While developing a new scenario, or while attempting to get a “feel” for the ways that a particular scenario might play out, it may be helpful to evaluate a single iteration of that scenario. Several output windows in *NAADSM* allow users to follow the course of a single iteration. These windows may be used while an iteration is in progress to provide a dynamic, “real time” view of simulated events as they occur in the model. Alternatively, they may be viewed at the end of an iteration to display a complete picture of the outbreak simulated by that iteration.

10.2.1. Map of units for 1 iteration

The **Output windows → Map of units for 1 iteration** option opens a map with latitude and longitude coordinates displaying all herds. (This window is displayed by default when a scenario file is opened.) The map can be used to display all herds, or only those of a particular production type. Herds can be coded by disease status, by herd size, or by method of exposure. Options are described below.

When a simulation is in progress, the map will show the population as it exists on the current simulated day (Figure 10-2). The map is dynamically updated as disease progresses within herds, as disease spreads among herds, and as control measure are applied.

When a simulation is not running, the map will display either the starting population (if no output from previous simulation runs is available) or the population in its final state after the last completed iteration.

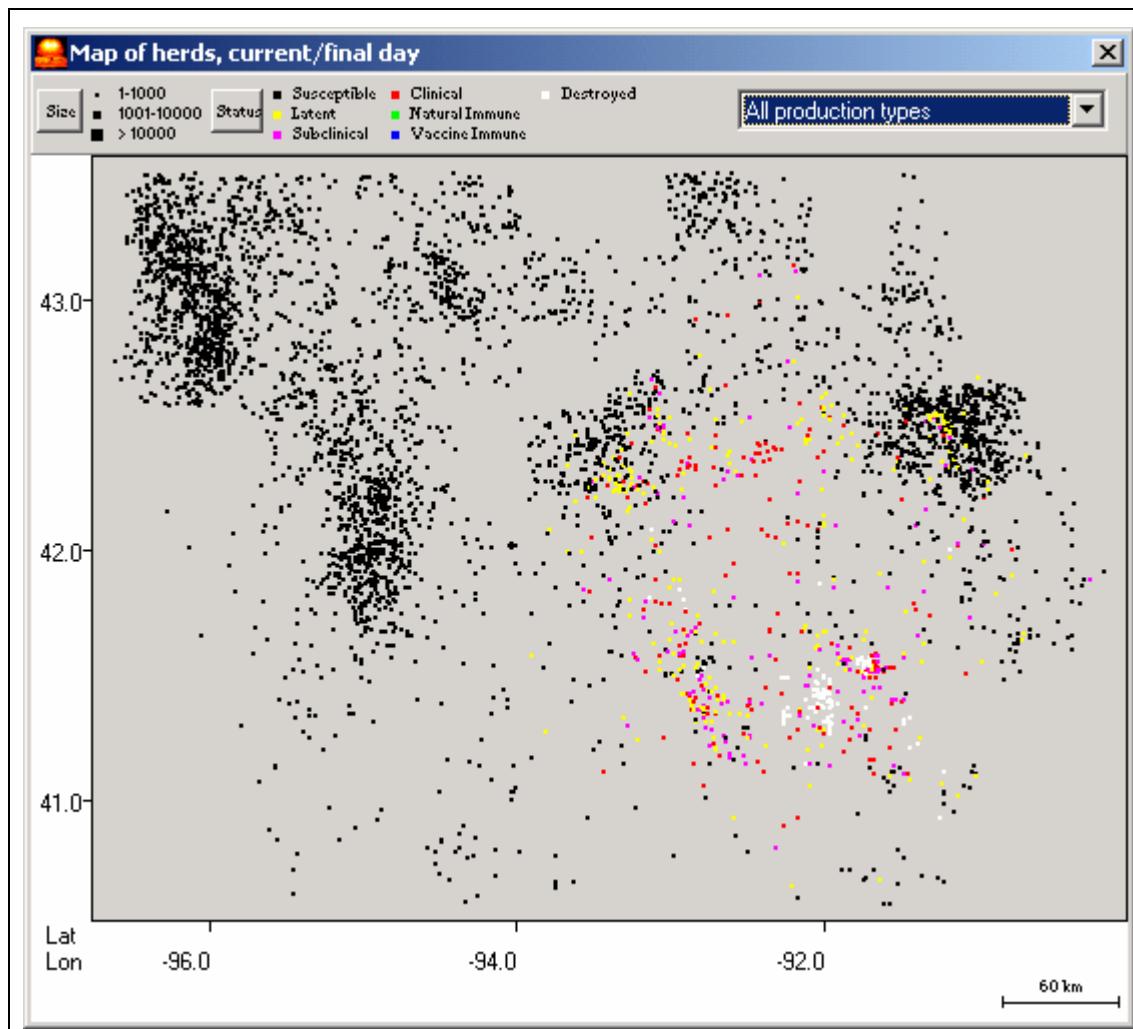


Figure 10-2. A map of herds from a simulation in progress. This view shows the actual disease status of affected herds, according to the color key shown at the top of the window. Contrast this map with the display in Figure 10-3.

10.2.1.1. Map options

Size: this button toggles between displaying all herds identically, or indicating larger herds with larger blocks.

Status: this button toggles between displaying herds based on their actual or their apparent status. Figure 10-2 shows herds based on their actual disease status (susceptible, latent, subclinical and infectious, clinical and infectious, naturally immune, vaccine immune, or destroyed). By contrast, Figure 10-3 shows the same herds on the same simulated day based on

10. Viewing model output

The **Output windows** menu

their detection status and method of exposure (unknown, detected, slaughtered/destroyed, vaccinated, exposed via direct contact, exposed via indirect contact: Figure 10-2).

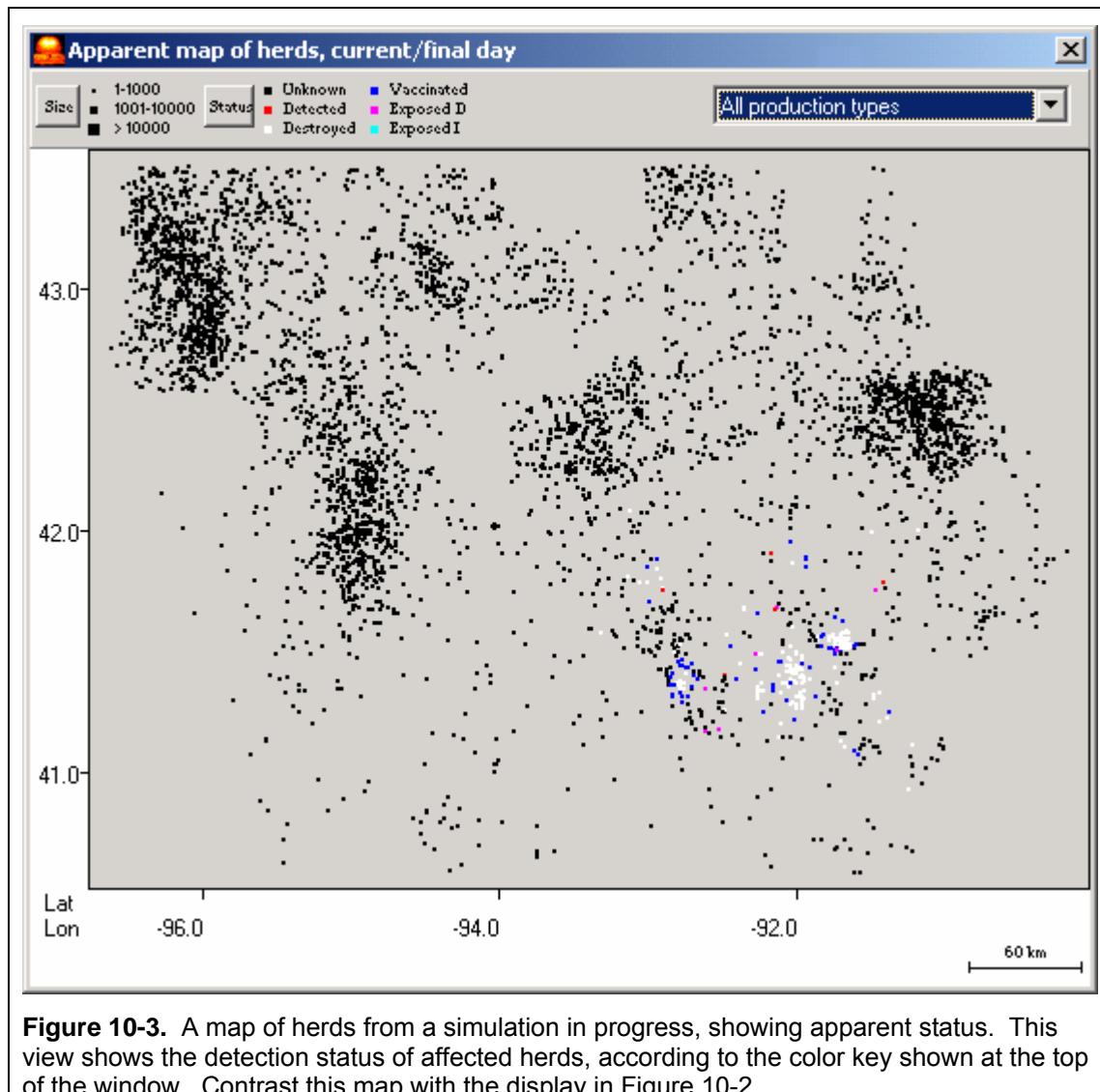


Figure 10-3. A map of herds from a simulation in progress, showing apparent status. This view shows the detection status of affected herds, according to the color key shown at the top of the window. Contrast this map with the display in Figure 10-2.

Production types selection box: use the drop-down menu to display only a specific production type on the map. By default, herds of all production types are displayed.

10.2.2. Daily unit status for 1 iteration

The **Output windows → Daily unit status for 1 iteration** command opens a chart that displays the proportion of herds in various actual disease states over the length of an outbreak (Figure 10-4).

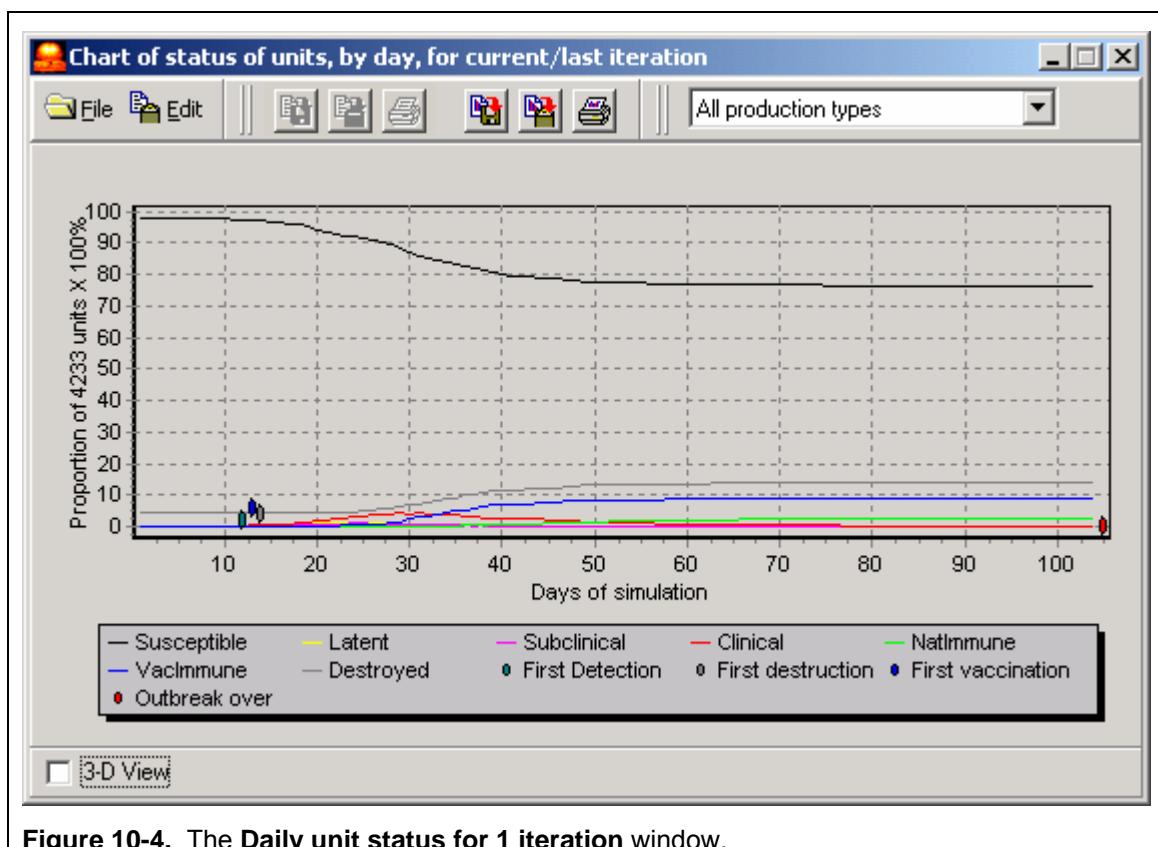


Figure 10-4. The Daily unit status for 1 iteration window.

Note in Figure 10-4 how the percentage of susceptible herds declines as the percentages of herds in the other disease stages increase over time.

This graph also displays markers to indicate when key events occurred in the iteration. A red dot (●) indicates the end of the outbreak. Other markers indicate the simulation day when the first detection of a clinically ill herd (teal: ●), the first destruction (gray: ●), and the first vaccination (blue: ●).

10.2.3. Summary of 1 iteration

The Output windows → Summary of 1 iteration command shows a comprehensive review of all of the major events that occur in a single iteration. This window has two tabs, one for epidemiological outputs and one for cost accounting outputs. The cost accounting tab will be enabled only if cost accounting parameters were assigned when the scenario was created (see Section 8.11).

10. Viewing model output

The **Output windows** menu

10.2.3.1. Epidemiological outputs

The **Epidemiology** tab summarizes the number of herds and animals detected, destroyed, and vaccinated during the outbreak from one iteration. This tab also shows epidemic curves for all herds infected during the course of the iteration (the actual epidemic curve) as well as for those herd detected during the course of the iteration (the apparent epidemic curve). Finally, this tab displays a breakdown of all infected herds and animals by method of infection.

Six checkboxes divide the **Epidemiology** tab into six main sections, each of which is discussed below. Check or uncheck one of these boxes (shown in Figure 10-5) to display or hide the outputs for that section.

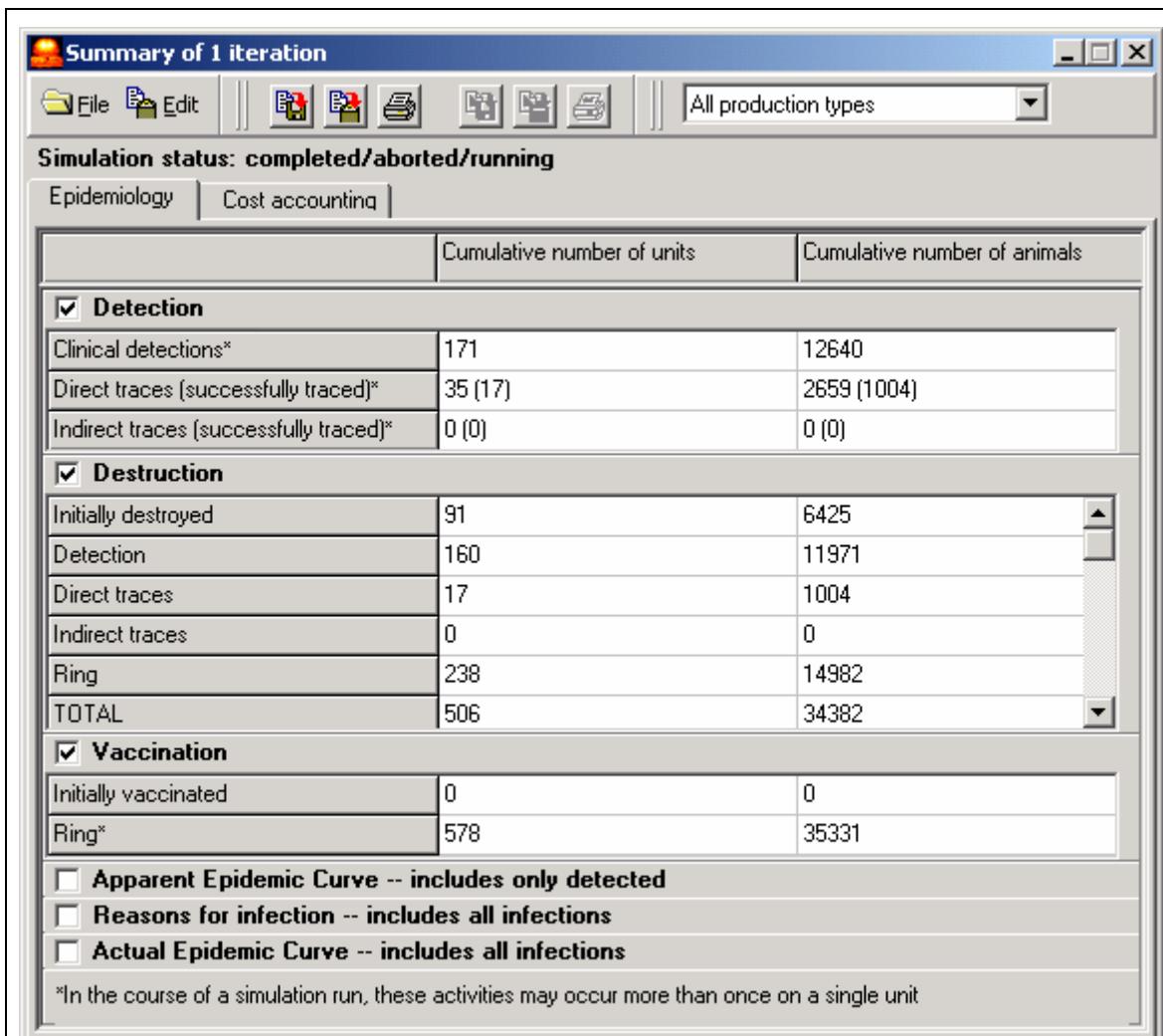


Figure 10-5. The Epidemiology tab of the **Summary of 1 iteration** window. Three of the six main sections (**Detection**, **Destruction**, and **Vaccination**) are displayed.

10.2.3.1.1. Detection

The **Detection** section of the **Epidemiology** tab (Figure 10-5) shows the number of units (herds) detected based on the appearance of clinical signs and trace-back investigations of direct or indirect contacts.

The total number of trace-back investigations attempted as well as the number of successful trace-back investigations is shown: the former number is given first, while the latter appears in parentheses as shown in Figure 10-5. Note that it is possible for a single unit to be identified multiple times by trace-back investigations during a single iteration. Each trace attempt is counted separately in the values reported in this section, even if the herd had been traced previously.

The number of animals in the detected herds is also shown.

10.2.3.1.2. Destruction

The **Destruction** section of the **Epidemiology** tab (Figure 10-5) shows the number of units that are destroyed (depopulated) over the course of an iteration. Destructions are broken down by the five reasons for destruction: units which have already been destroyed at the start of the scenario (**Initially destroyed**); units destroyed as a result of detection by clinical signs of disease (**Detection**); units preemptively destroyed because of contact with a herd that was later detected (**Direct traces** and **Indirect traces**: see Section 8.8); and units preemptively destroyed because of proximity to a detected herd (**Ring** destruction: see Section 8.9.3.).

The number of animals in the destroyed herds is also shown.

10.2.3.1.3. Vaccination

The **Vaccination** section of the **Epidemiology** tab (Figure 10-5) shows the number of units that are vaccinated over the course of an iteration. Herds may have been vaccinated at the start of the scenario (**Initially vaccinated**) or may be vaccinated as part of a **Ring** vaccination campaign (see Section 8.10). Note that it is possible for a single unit to be vaccinated multiple times during a single iteration. Each time a unit is vaccinated, it is counted in this value.

The number of animals in the vaccinated herds is also shown.

10.2.3.1.4. Reasons for infection

The **Reasons for infection** section of the **Epidemiology** tab (Figure 10-6) shows the number of units that are infected by various mechanisms over the course of an iteration. Note that infections are actual events, not apparent events: while it is possible to infer the source of an infection by surveillance, the true source and timing of infection will never be completely apparent. Note that it is possible for a single unit to be infected multiple times during a single iteration. Each infection is counted as a separate event in this section.

10. Viewing model output

The **Output windows** menu

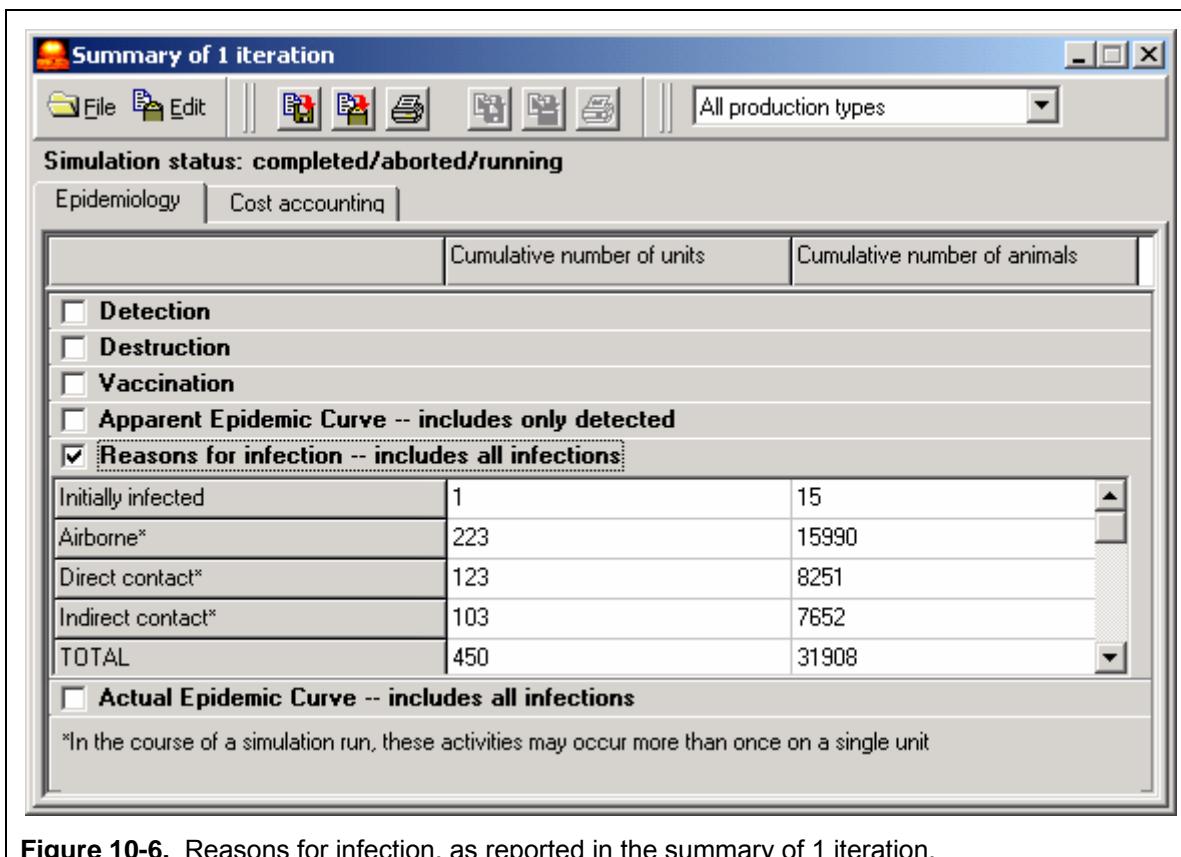


Figure 10-6. Reasons for infection, as reported in the summary of 1 iteration.

Units may be infected at the beginning of a scenario (**Initially infected**), or may become infected by **Airborne** spread, **Direct contact** with an infectious unit, or **Indirect contact** with an infectious unit.

The number of animals in herds infected by these mechanisms is also shown.

10.2.3.1.5. Actual and apparent epidemic curves

The **Apparent Epidemic Curve** and **Actual Epidemic Curve** sections of the **Epidemiology** tab (Figure 10-7) show two versions of the epidemic curve for the outbreak over the course of a single iteration. The apparent epidemic curve is based on the number of detected herds; the actual epidemic curve is based on the actual number of infected herds (see Section 10.1). For both graphs, time (in days) from the start of the simulation is on the *x* axis, and number of herds newly detected/infected on that day is shown on the *y* axis.

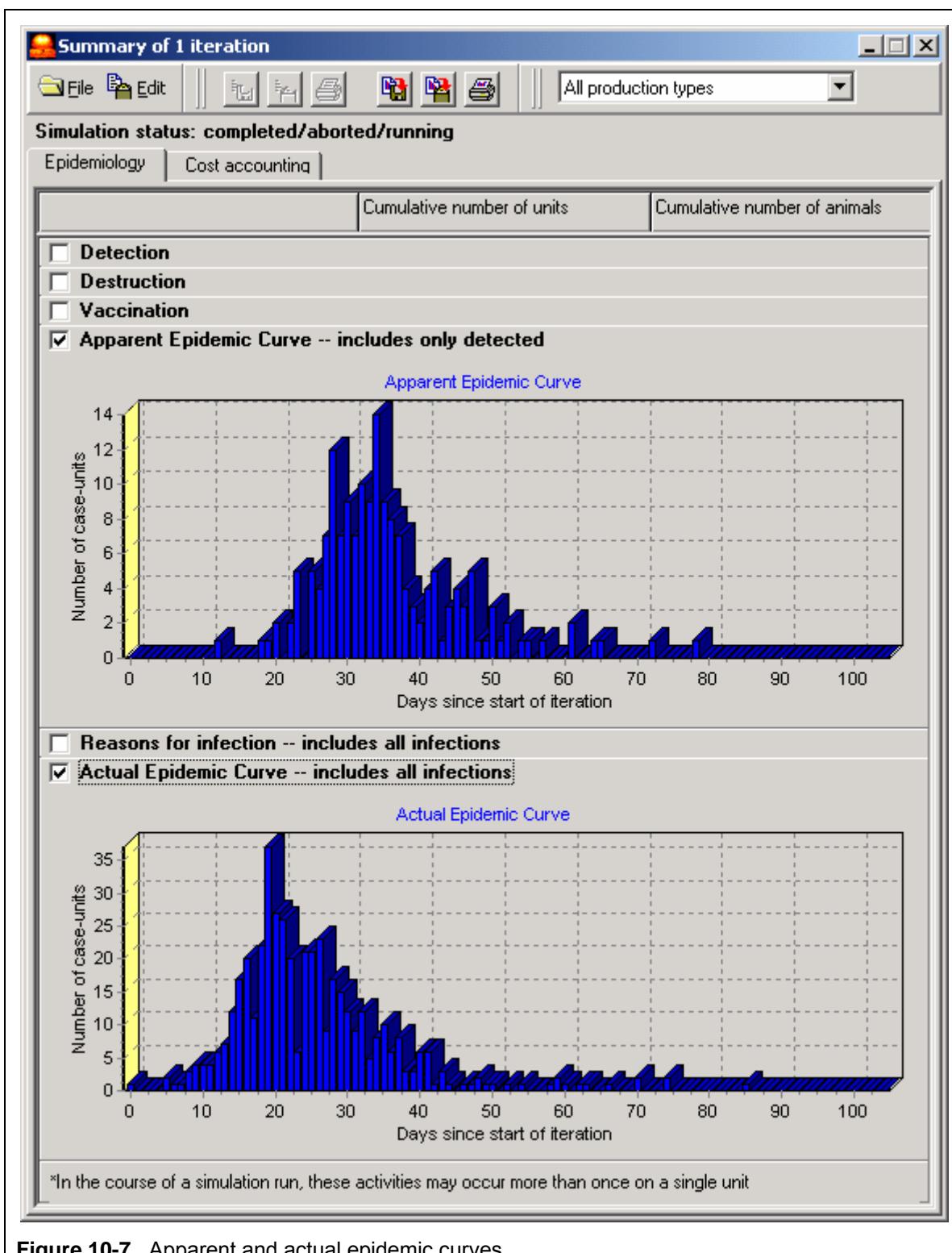


Figure 10-7. Apparent and actual epidemic curves.

10. Viewing model output
The **Output windows** menu

10.2.3.2. Cost accounting outputs

If cost accounting was enabled when the scenario was set up (Section 8.11), the **Cost accounting** tab of the **Summary of 1 iteration** window will be activated (Figure 10-8). This tab has two main sections: the **Table of daily costs**, and the **Graph of daily costs**.

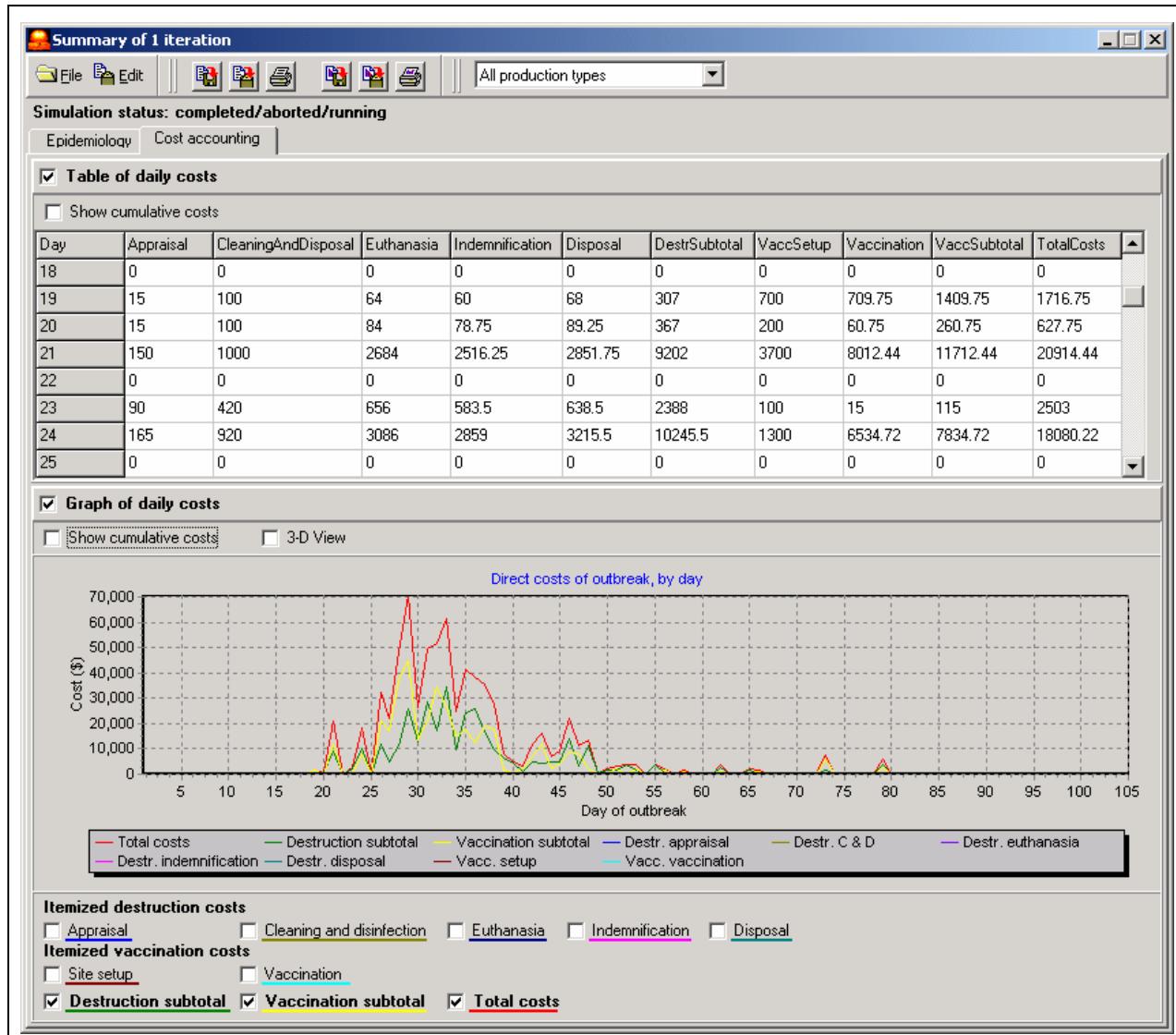


Figure 10-8. The **Cost accounting** tab of the **Summary of 1 iteration** window.

10.2.3.2.1. Table of daily costs

The **Table of daily costs** section of the **Cost accounting** tab presents a summary of the direct costs incurred on each day of the outbreak. Costs are itemized by the nine categories used for cost accounting (Section 8.11). Subtotals for costs associated with vaccination and destruction are also available. By default, the values shown in the table reflect the new costs incurred on that day. Check the **Show cumulative costs** box to display cumulative costs instead.

10.2.3.2.2. Graph of daily costs

The **Graph of daily costs** section of the **Cost accounting** tab presents a graph displaying the direct costs incurred on each day of the outbreak. Costs are itemized as above. Select as many itemized categories in the bottom portion of the window as desired to plot itemized costs separately.

By default, daily values are shown in the graph. Check the **Show cumulative costs** box to display cumulative costs instead.

10.3. Daily events and exposures

Most of the time, detailed information about each daily event or exposure is unnecessary: summary information, particularly for multiple iterations, will be far more useful for analytical purposes. In some instances, however, it might be helpful to run a single iteration and track individual events and exposures for every day of that iteration. The commands **Output windows** → **Events by day** and **Contact-exposure events by day** are used to display information for events and exposures.

NOTE: Enabling daily events and exposures

By default, daily events and exposures are not recorded, and these two menu commands are disabled. To record daily events and exposures, set the appropriate output options (Section 8.12).

Even a simple scenario may generate a lot of events and exposures. Make sure that you have plenty of hard disk space when using these output options.

Events involve only a single unit. Examples of events include disease state changes (see Section 8.5) and destruction of an infected unit. A complete list of the events recorded by NAADSM is given in Table 10-2.

10. Viewing model output

The **Output windows** menu

Table 10-2. Events recorded by NAADSM.

Event type	Description
State change	Any change in disease transition state, either by natural progression (e.g. “latent” to “subclinical” or “clinical” to “naturally immune” or by intervention (e.g. “susceptible” to “destroyed”). See Figure 8-15 for all possible state changes.
Infection	Susceptible units are infected upon effective contact with or airborne disease transmission from a contagious herd.
Detection	Herds showing clinical signs may be detected according to the criteria described in Section 8.7.
Destruction	A destruction event is recorded on the day that a unit is destroyed for reasons described in Section 8.8. The transition state of a destroyed unit will change to “Destroyed” on the day after the Destruction event is recorded.
Vaccination	A vaccination event is recorded on the day that a unit is vaccinated for reasons described in Section 8.8. The transition state of a vaccinated unit will change to “Vaccine immune” after the specified delay, if the unit was susceptible when it was vaccinated and if it does not become infected during the delay period.
Trace of direct contact	This event occurs when a unit is successfully identified by trace-back investigation of a direct contact (see Section 8.8).
Trace of indirect contact	This event occurs when a unit is successfully identified by trace-back investigation of an indirect contact (see Section 8.8).

Exposures involve two units and the possible transmission of disease. There are three types of exposures (mechanisms of disease spread) in NAADSM: these are direct contact, indirect contact, and airborne spread (see Section 8.6).

10.3.1. The Events for 1 iteration window

Output windows → Events by day is used to open the **Events for 1 iteration** window (Figure 10-9). This window displays all of the events recorded for the most recent iteration of a scenario in a tabular format. Table 10-2 describes all of the recorded events. Table 10-3 shows all of the information recorded for each event.

A lengthy list of events may be difficult to view. To simplify the process, the table of events may be sorted into a particular order, or may be filtered to display only events that match a particular criterion.

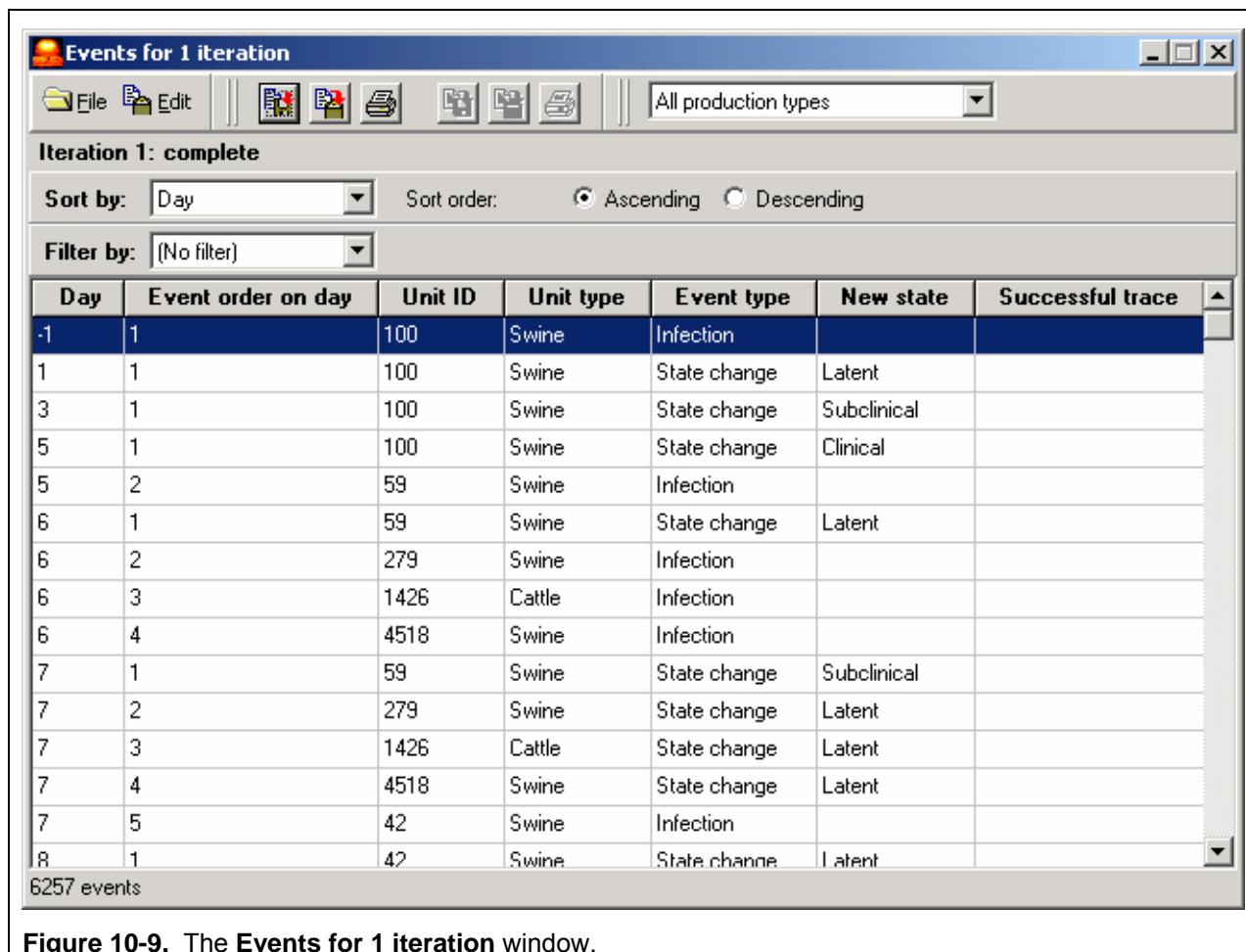


Figure 10-9. The Events for 1 iteration window.

Table 10-3. Columns of the daily events window.

Column header	Description
Day	The simulation day on which the event occurred.
Event order on day	The order in which events occurred on the specified simulation day.
Unit ID	The ID number of the herd affected by the event.
Unit type	The production type of the herd affected by the event.
Event type	The type of event. See Table 10-2.
New state	For transition state change events, the new state of the unit that underwent the state change.
Successful trace	For traces of direct and indirect contact, an indication of whether the trace was successful (TRUE) or not (FALSE).

10. Viewing model output

The **Output windows** menu

NOTE: Events on day “-1”

A close look at Figure 10-9 will show that the first event recorded occurred on day “-1” of the iteration in question. “-1” indicates that an event occurred before to the point in time when a simulation starts. For example, any herds that are infected at the beginning of a scenario had to have been infected some time before the beginning of the scenario. The model does not have or need specific information about when these events occurred: it is sufficient to say that they occurred prior to the period covered by the simulation.

Day 1 in the events and exposures tables represents the first day simulated by NAADSM for an iteration.

10.3.1.1. Sorting events

Events may be sorted in ascending or descending order by any column. Use the **Sort by** dropdown menu to select the column to use for sorting, and select a sort direction (Figure 10-10a). Alternatively, click inside a column header to sort by the selected column. Click on the column header again to reverse the sort direction. A sort direction indicator (a small triangular arrow) will appear in the column header to indicate the sorted column and the direction of the sort (Figure 10-10).

(a)

Day	Event order on day	Unit ID	Unit type	Event type	New state	Successful trace
-1	1	100	Swine	Infection		
1	1	100	Swine	State change	Latent	
3	1	100	Swine	State change	Subclinical	
5	1	100	Swine	State change	Clinical	
5	2	59	Swine	Infection		
6	1	59	Swine	State change	Latent	
6	2	279	Swine	Infection		

6257 events

(b)

Day	Event order on day	Unit ID	Unit type	Event type	New state	Successful trace
56	8	1562	Cattle	Detection		
59	7	2150	Cattle	Detection		
62	4	1811	Cattle	Detection		
67	2	3357	Cattle	Detection		
68	4	4007	Cattle	Detection		
-1	1	100	Swine	Infection		
5	2	59	Swine	Infection		

6257 events

Figure 10-10. Sorting events by event type. **(a)** Selecting event type from the Sort by dropdown menu. Note that, prior to sorting, events are sorted in ascending order by Day, as shown by the sort indicator (the small gray arrow) in the Day column. **(b)** After sorting by event type. The sort direction indicator now appears in the column Event type.

10. Viewing model output

The Output windows menu

10.3.1.2. Filtering events

A subset of the recorded events may be displayed by filtering the complete list based on **Day**, **Unit ID**, or **Event type**. Use the **Filter by** dropdown menu to select an attribute to use for a filter (Figure 10-11a). Next, enter a value to use as the filter criterion. Depending on the selected filter attribute, the filter criterion might be entered in another dropdown menu or in a text box (Figure 10-11b shows an example of the latter).

The screenshot displays three panels (a), (b), and (c) of the 'Events for 1 iteration' application window.

(a) The 'Filter by' dropdown is set to 'Day'. A context menu is open over the 'Filter by' dropdown, with 'Day' selected. The table below shows two rows of data: Day 17, Unit ID 91, Event type Swine, New state Destruction; and Day 17, Unit ID 279, Event type Swine, New state Destruction.

Day	Unit ID	Unit type	Event type	New state	Successful tra
17	91	Swine	Destruction		
17	279	Swine	Destruction		

(b) The 'Filter by' dropdown is set to 'Day'. The 'Day:' text input field contains '25'. To the right of the input field are two buttons: a green checkmark button and a red X button.

(c) The 'Filter by' dropdown is set to 'Day'. The 'Day:' text input field contains '25'. The table now shows a filtered list of events for Day 25:

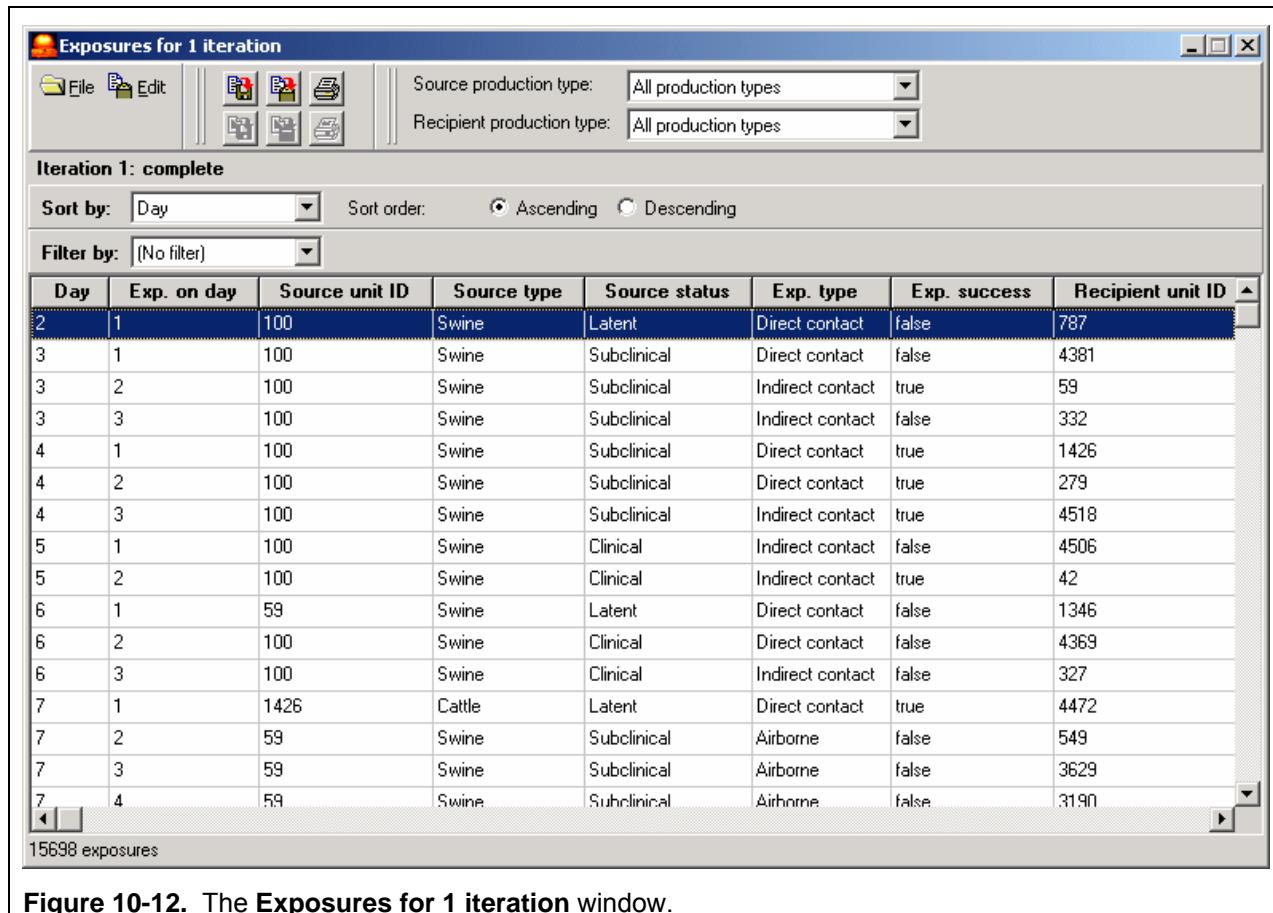
Day	Event order on day	Unit ID	Unit type	Event type	New state	Successful tra
25	200	17	Swine	Destruction		
25	201	37	Swine	Destruction		
25	202	132	Swine	Destruction		
25	204	215	Swine	Destruction		
25	213	579	Cattle	Destruction		
25	215	611	Cattle	Destruction		
25	216	612	Cattle	Destruction		

In the bottom left corner of the window, the message '344 events (filtered)' is displayed.

Figure 10-11. Filtering events by day. **(a)** Selecting **Day** from the **Filter by** dropdown menu. **(b)** Specifying a day. Click on the button or press the **Enter** key to apply the filter criterion. **(c)** The filtered list of events. Note the message in the lower left portion of the window, displaying the number of events which meet the filter criterion. Also note that sort order is unchanged.

10.3.2. The Exposures for 1 iteration window

Output windows → Contact-exposure events by day is used to open the **Exposures for 1 iteration** window (Figure 10-12). This window displays all of the exposures recorded for the most recent iteration of a scenario in a tabular format. Table 10-4 shows all of the information recorded for each exposure.



The screenshot shows a Windows application window titled "Exposures for 1 iteration". The window has a toolbar with icons for File, Edit, and various data management functions. Below the toolbar are dropdown menus for "Source production type" (set to "All production types") and "Recipient production type" (set to "All production types"). A message "Iteration 1: complete" is displayed above the main table area. The table has columns: Day, Exp. on day, Source unit ID, Source type, Source status, Exp. type, Exp. success, and Recipient unit ID. The data shows multiple rows of exposure records, with the last row indicating 15698 exposures. The table is sorted by "Day" and "Exp. on day".

Day	Exp. on day	Source unit ID	Source type	Source status	Exp. type	Exp. success	Recipient unit ID
2	1	100	Swine	Latent	Direct contact	false	787
3	1	100	Swine	Subclinical	Direct contact	false	4381
3	2	100	Swine	Subclinical	Indirect contact	true	59
3	3	100	Swine	Subclinical	Indirect contact	false	332
4	1	100	Swine	Subclinical	Direct contact	true	1426
4	2	100	Swine	Subclinical	Direct contact	true	279
4	3	100	Swine	Subclinical	Indirect contact	true	4518
5	1	100	Swine	Clinical	Indirect contact	false	4506
5	2	100	Swine	Clinical	Indirect contact	true	42
6	1	59	Swine	Latent	Direct contact	false	1346
6	2	100	Swine	Clinical	Direct contact	false	4369
6	3	100	Swine	Clinical	Indirect contact	false	327
7	1	1426	Cattle	Latent	Direct contact	true	4472
7	2	59	Swine	Subclinical	Airborne	false	549
7	3	59	Swine	Subclinical	Airborne	false	3629
7	4	59	Swine	Subclinical	Airborne	false	3190

15698 exposures

Figure 10-12. The Exposures for 1 iteration window.

10. Viewing model output

The **Output windows** menu

Table 10-4. Columns of the daily exposures table.

Column header	Description
Day	The simulated day on which the exposure occurred.
Exp. on day	The order in which exposures occurred on the specified simulation day.
Source unit ID	The ID number of the herd which was the source of the exposure.
Source type	The production type of the source herd.
Source status	The actual disease status (see Table 8-3) of the source herd when the exposure occurred.
Exp. type	The mechanism of exposure: direct contact, indirect contact, or airborne spread (see Section 8.6)
Exp. success	TRUE if the exposure would have resulted in infection of a susceptible herd. Otherwise FALSE.
Recipient unit ID	The ID number of the herd which was the recipient (target) of the exposure.
Recipient type	The production type of the recipient (target) herd.
Recipient status	The actual disease status (see Table 8-3) of the recipient (target) herd when the exposure occurred.

NOTE: Interpreting exposure success

As long as a herd has not been quarantined (see Appendix A) or destroyed, it may be the target (recipient) of an exposure. Herds that have already been infected or herds that are vaccine immune, for example, can be exposed. In cases like these, an exposure which would normally be adequate to transmit disease will have no effect. Such exposures are still recorded as "successful" by NAADSM.

Table 10-4 offers a different interpretation of the same principle: an exposure will be recorded as successful if it would have resulted in the infection of a susceptible herd.

10.3.2.1. Sorting and filtering exposures

The table in the **Exposures for 1 iteration** window may be sorted or filtered in the same way described for the **Events for 1 iteration** window (Section 10.3.1.1 and Section 10.3.1.2).

10.4. Summarizing output from multiple iterations

While model output based on individual iterations may be interesting and useful in a limited way, the real power of stochastic modeling comes from its ability to generate a

distribution of possible outcomes based on many iterations (Section 3). The windows discussed in this section allow users to explore the range of possible outcomes.

10.4.1. The **Output statistics** window

Depending on the purpose of an analysis, many different characteristics of a disease outbreak might be of interest: examples include the duration of an outbreak, the total number of herds affected, or the overall direct cost of disease control. *NAADSM* offers a fairly comprehensive set of outputs to address a wide variety of analytical purposes. The **Output windows → Output statistics** command is used to display the output statistics associated with multiple iterations of a scenario. The **Output statistics** window is probably the most important and certainly the most informative screen in the *NAADSM* application.

10.4.1.1. Epidemiological and cost accounting outputs

The **Output statistics** window (Figure 10-13) has two tabs, one for epidemiological outputs and one for cost accounting outputs. The cost accounting tab will be enabled only if cost accounting parameters were assigned when the scenario was created (see Section 8.11). These tabs have the same features and are formatted identically: the only difference is the set of calculations that they display. Each tab has three major components: an output table, a histogram, and a convergence plot. These components are described in detail in the following sections.

10. Viewing model output

The **Output windows** menu

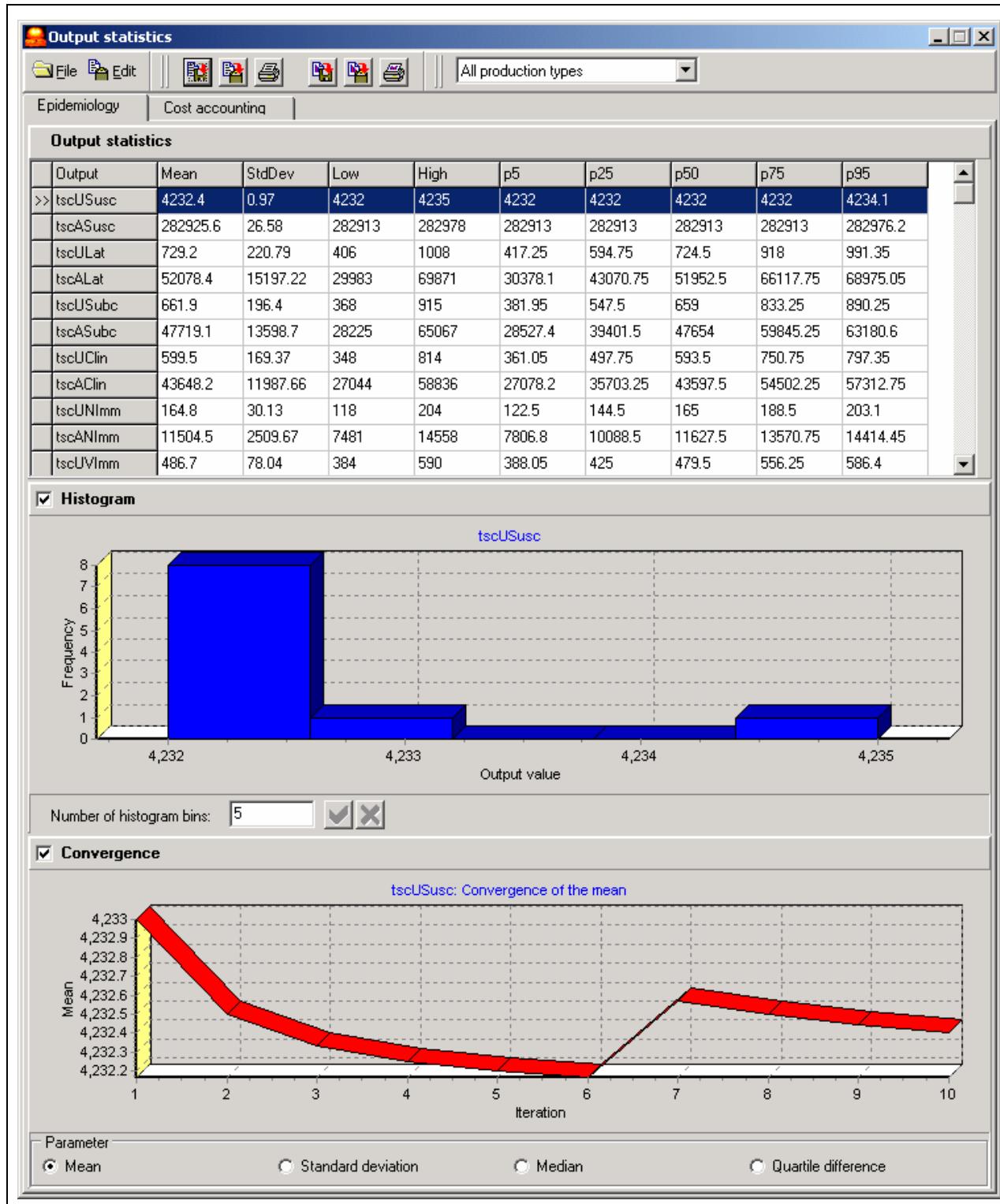


Figure 10-13. The **Output statistics** window. The variable **tscUSusc** is selected in the **Output statistics** table. Values of **tscUSusc** from each iteration were used to generate the **Histogram**. The **Convergence** plot shows how the mean value of **tscUSusc** was affected by each new iteration.

10.4.1.2. The output table

The table at the top of the **Output statistics** window shows all of the outputs available in NAADSM. In the interest of saving screen space, outputs are displayed with an abbreviated and somewhat cryptic name. A definition of each output may be viewed by clicking on the output name in the **Output** column (Figure 10-14). Select a row in the output table to view a histogram and a convergence plot for the output in that row (see Sections 10.4.1.3 and 10.4.1.4).

NOTE: Variable definitions

Complete descriptions of the outputs on the **Epidemiology** and **Cost accounting** tabs are provided in Appendix B.

Output statistics										
Output	Mean	StdDev	Low	High	p5	p25	p50	p75	p95	
>>tscUSusc	4232.4	0.97	4232	4235	4232	4232	4232	4232	4234.1	
tscASusc	282976.2	282913	282913	282913	282913	282913	282913	282913	282976.2	
tscULat	723.2	220.73	406	1000	417.23	534.75	724.5	918	991.35	
tscALat	15197.22	29993	89971	20379.1	43070.75	51952.5	61117.75	69975.05		

Figure 10-14. Displaying output definitions. Click on the name of a variable in the **Output** column to view a definition of the variable.

The output table shows the following statistics for most variables:

- **Mean** value across all iterations
- Standard deviation (**StdDev**)
- Low value (**Low**)
- High value (**High**)
- 5th percentile (**p5**)
- 25th percentile (**p25**)
- Median (**p50**)
- 75th percentile (**p75**)
- 95th percentile (**p95**)

In a few cases, these statistics are not applicable. Near the bottom of the output table on the **Epidemiology** tab are listed several qualitative variables (Figure 10-15). One of these is a simple count of the number of iterations in which at least one herd was detected by clinical signs (**detOccurred**). Others are **vaccOccurred** (the number of iterations in which vaccination occurred), **destrOccurred** (the number of iterations in which destruction occurred), and **outbreakEnded** (the number of iterations in which the outbreak ended: see Appendix B for a description of this odd-seeming variable). These values reported for these variables show the count and the proportion of iterations for which the indicated condition was met.

10. Viewing model output

The **Output windows** menu

vaccining	40231.2	10110.73	33301	03330	33030.03	42213	43340.3	33077.3	01231
detOccurred	10 of 10 iterations								
firstDet	15.6	2.07	12	18	12.45	14.25	16	17	18
vaccOccurred	10 of 10 iterations								
firstVacc	16.8	1.87	13	19	13.9	16	17	18	19
destrOccurred	10 of 10 iterations								
firstDestr	17.6	2.07	14	20	14.45	16.25	18	19	20
outbreakEnded	10 of 10 iterations								
outbreakLen	111.6	13.44	95	139	95.9	100.25	113	115	132.25

Figure 10-15. Qualitative variables on the **Epidemiology** tab: **detOccurred**, **vaccOccurred**, **destrOccurred**, and **outbreakEnded**. The statistical calculations shown for the other variables do not apply in these cases.

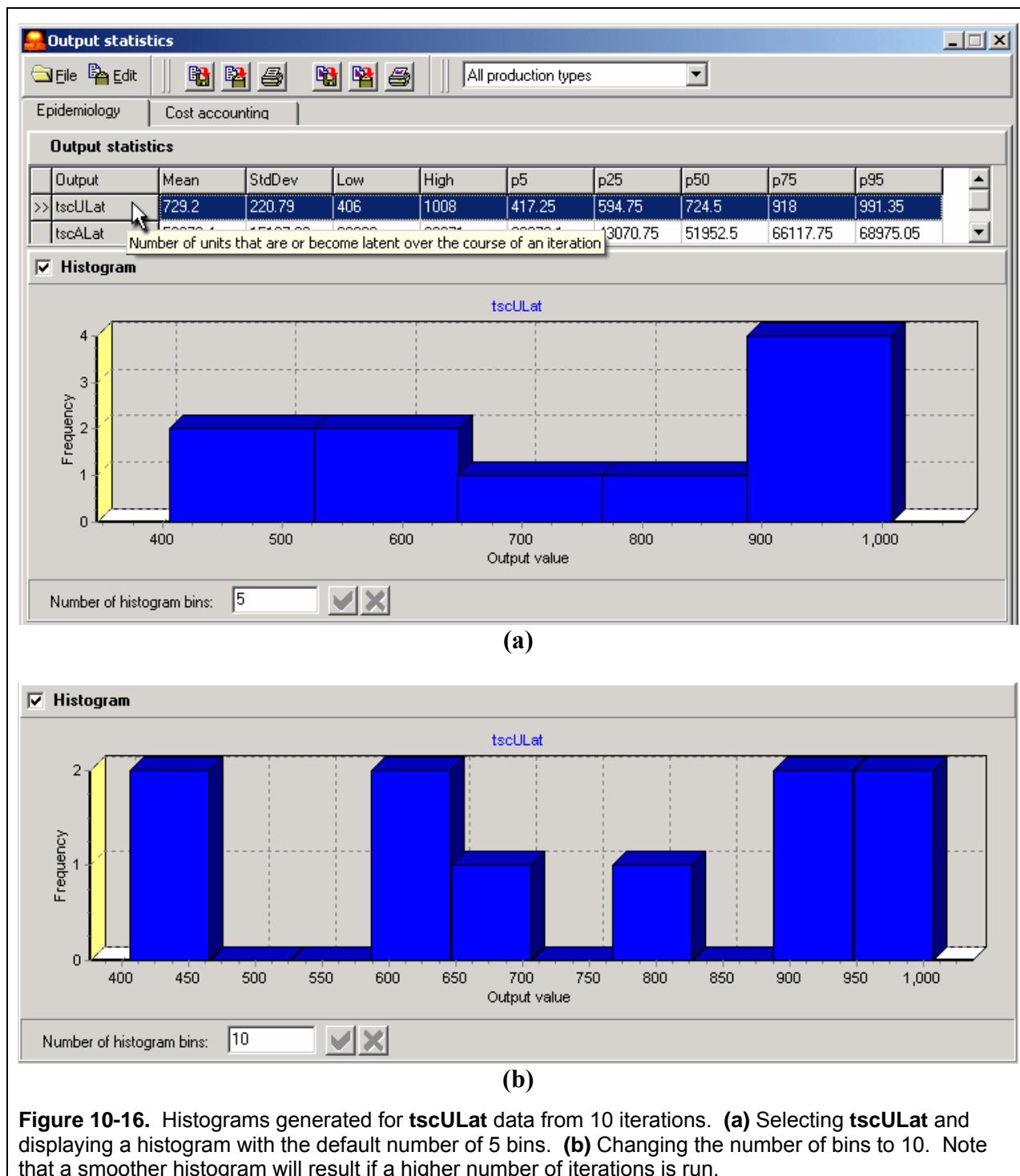
NOTE: Obtaining the data used to generate summary statistics

The raw values used to calculate the summary statistics shown in the **Output statistics** window are not accessible via NAADSM, but they are available in the Microsoft Access-compatible (*.mdb) scenario file. See Appendices B and E for more information.

The abilities to view and export raw data may be incorporated in a future version of NAADSM.

10.4.1.3. The histogram

A histogram can be used to help give a sense of the distribution of values obtained for a particular output. Check the **Histogram** box to view a histogram of the values used to calculate the statistics shown in the selected row of the output table (Figure 10-16a). Change the **Number of histogram bins** to alter the appearance of the histogram (Figure 10-16b).



10. Viewing model output

The **Output windows** menu

10.4.1.4. The convergence plot

The convergence plot shown in Figure 10-17 shows the effect that each new iteration has on the overall mean value of **tscULat**. As more iterations are run, the convergence plot flattens out, indicating that additional iterations are decreasingly likely to have a notable influence on the distribution of results. Plots like this can be used to provide a rough estimate of how many iterations are “enough” to obtain reliable results.

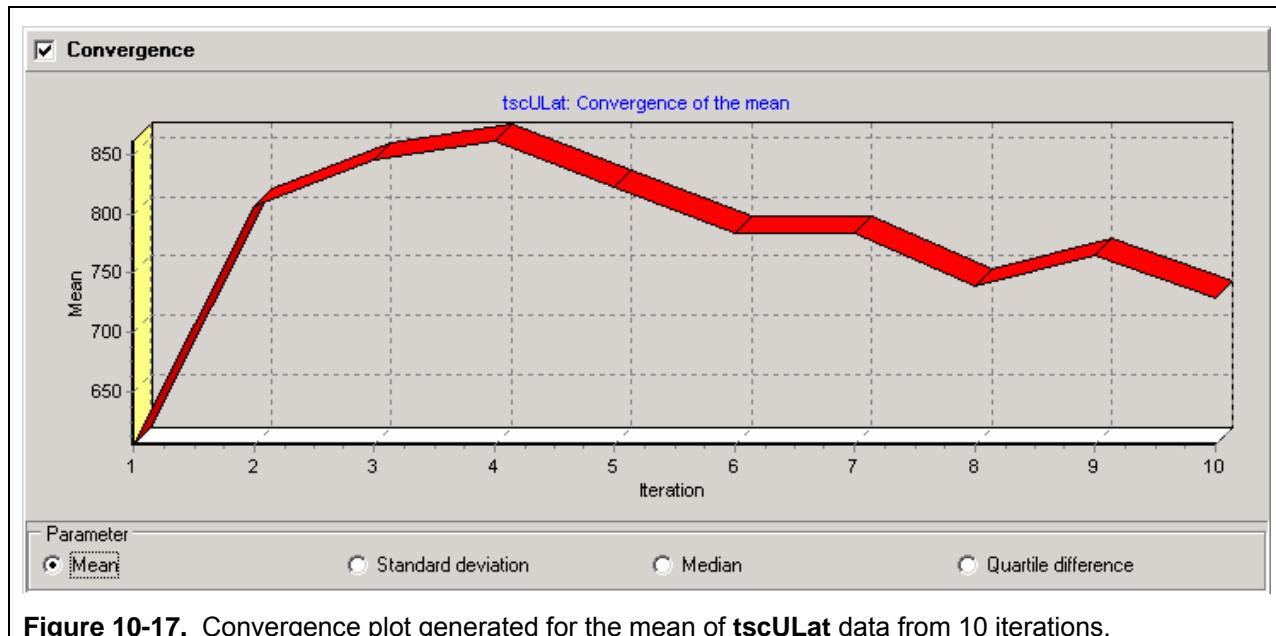


Figure 10-17. Convergence plot generated for the mean of **tscULat** data from 10 iterations.

Check or uncheck the **Convergence** box to show or hide the convergence plot. Change the selection in the **Parameter** panel at the bottom of the screen to show the convergence of the **Mean**, **Standard deviation**, **Median**, or **Quartile difference**.

10.4.2. The **Summary epidemic curves** window

An epidemic curve can be generated pretty easily for a single disease outbreak. Typical output of an *NAADSM* scenario, however, includes data from multiple simulated outbreaks. The options of the **Summary epidemic curves** window (displayed by the **Output windows → Summary epidemic curves** command) offer one possible approach for presenting something akin to an epidemic curve for the multiple iterations generated by in an *NAADSM* simulation.

Before demonstrating the operation of the **Summary epidemic curves** window, the procedure used to generate the summary curves is demonstrated by example in the following section.

10.4.2.1. Summary epidemic curves by example

Table 10-1 shows output from an *NAADSM* simulation involving a single production type. Twenty iterations of the scenario were run. The upper portion of the table shows the number of new cases that occurred on each simulated day of each iteration. Individual iterations are shown in the rows of the table. The shortest outbreak (the 15th iteration) had a duration of 13 days, while other iterations lasted up to 19 days. By day 20, outbreaks had ended in all iterations.

Epidemic curves for individual iterations could be produced as shown in Figure 10-18. Each series in this figure represents an epidemic curve for a single iteration.

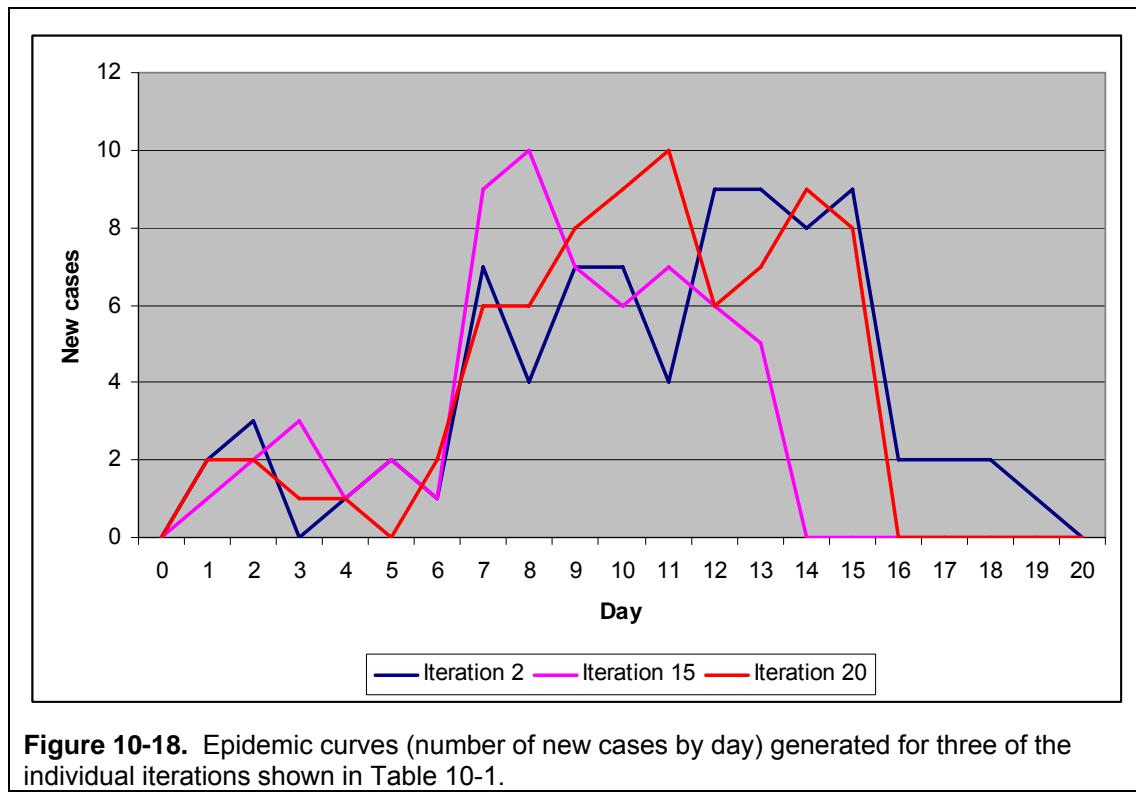


Figure 10-18. Epidemic curves (number of new cases by day) generated for three of the individual iterations shown in Table 10-1.

The lower portion of Table 10-1 shows a set of summary statistics generated for each simulation day across multiple iterations: for example, the mean number of new cases that occurred on day one (down the column for all 20 iterations) is 1.8. In addition to the mean, the 95th percentile (p95), the median (50th percentile or p50) and the 5th percentile (p5) are shown.

Looking across the rows in the lower section of Table 10-1, one might notice that the summary statistics calculated for each day are reminiscent of the epidemic curve data for each iteration in the upper portion of the table. The **Summary epidemic curves** window in Figure 10-19 shows “epidemic curves” drawn from the summary statistics of Table 10-1.

10. Viewing model output
 The **Output windows** menu

Table 10-1. Sample data for the generation of summary epidemic curves. The rows of the upper portion of the table show the number of new cases that occurred on each day for each of 20 iterations. Epidemic curves for the highlighted rows are shown in Figure 10-18. The rows of the lower part of the table show summary calculations for each simulation day (column) of the 20 iterations. These values are shown in the summary epidemic curves in Figure 10-19. **ITER.** column: iteration numbers. **DUR.** column: duration in days of the outbreak from each iteration. **p95:** 95th percentile. **p50:** 50th percentile (median). **p5:** 5th percentile.

	DAY																				
ITER.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	DUR.
1	3	1	2	2	1	1	7	10	8	7	4	9	10	7	2	3	0	0	0	0	16
2	2	3	0	1	2	1	7	4	7	7	4	9	9	8	2	2	1	0	19	0	
3	2	0	0	3	2	7	7	9	10	7	4	9	5	1	2	2	0	0	0	17	
4	1	2	0	0	0	0	5	5	4	5	5	10	5	5	0	0	0	0	0	0	14
5	3	1	3	3	0	0	8	4	4	6	8	4	9	5	1	1	2	0	0	0	17
6	3	3	1	2	1	0	9	6	4	7	10	7	6	3	10	3	1	0	0	0	17
7	2	3	0	2	2	0	6	5	9	6	6	6	7	7	5	3	1	2	4	0	19
8	1	0	3	0	1	0	10	7	6	8	10	6	8	6	10	4	2	0	0	0	17
9	2	1	2	1	3	1	8	4	10	7	10	5	10	6	2	0	0	0	0	0	15
10	1	0	0	1	2	0	4	7	7	5	7	5	4	4	5	3	4	2	1	0	19
11	1	2	1	0	3	3	5	8	5	5	9	8	5	10	2	1	2	0	0	0	17
12	2	1	3	1	0	0	9	4	4	5	6	7	0	3	2	0	0	0	0	0	15
13	1	0	1	3	2	2	10	7	4	6	7	9	7	6	7	3	0	0	0	0	16
14	2	1	2	1	0	6	6	10	10	5	5	5	7	4	0	0	0	0	0	0	15
15	1	2	3	1	2	1	9	10	7	6	7	6	5	0	0	0	0	0	0	13	
16	3	1	0	1	2	0	8	4	5	4	9	7	3	10	8	2	0	0	0	0	16
17	1	0	2	2	0	1	8	6	9	7	5	7	1	3	0	0	0	0	0	0	14
18	1	3	2	2	1	0	5	6	9	7	7	6	9	5	2	1	3	2	0	0	19
19	2	1	3	0	3	2	5	5	9	10	7	7	4	8	8	4	0	0	0	0	16
20	2	2	1	1	0	2	6	6	8	9	10	6	7	9	8	0	0	0	0	0	15
<i>p95</i>	3	3	3	3	2.05	10	10	10	9.05	10	10	9.05	10	10	4	2.1	2.05	2.1	0	0	
<i>p50</i>	2	1	1.5	1	1.5	0.5	7	6	7	7	7	7	6	6	4.5	2	0	0	0	0	
<i>Mean</i>	1.8	1.35	1.45	1.2	1.45	0.8	7.1	6.05	6.9	6.85	7.15	6.7	6	6.05	4.45	1.65	0.85	0.45	0.4	0	
<i>pP5</i>	1	0	0	0	0	0	4.95	4	4	4.95	4	4	4	0.95	2.85	0	0	0	0	0	

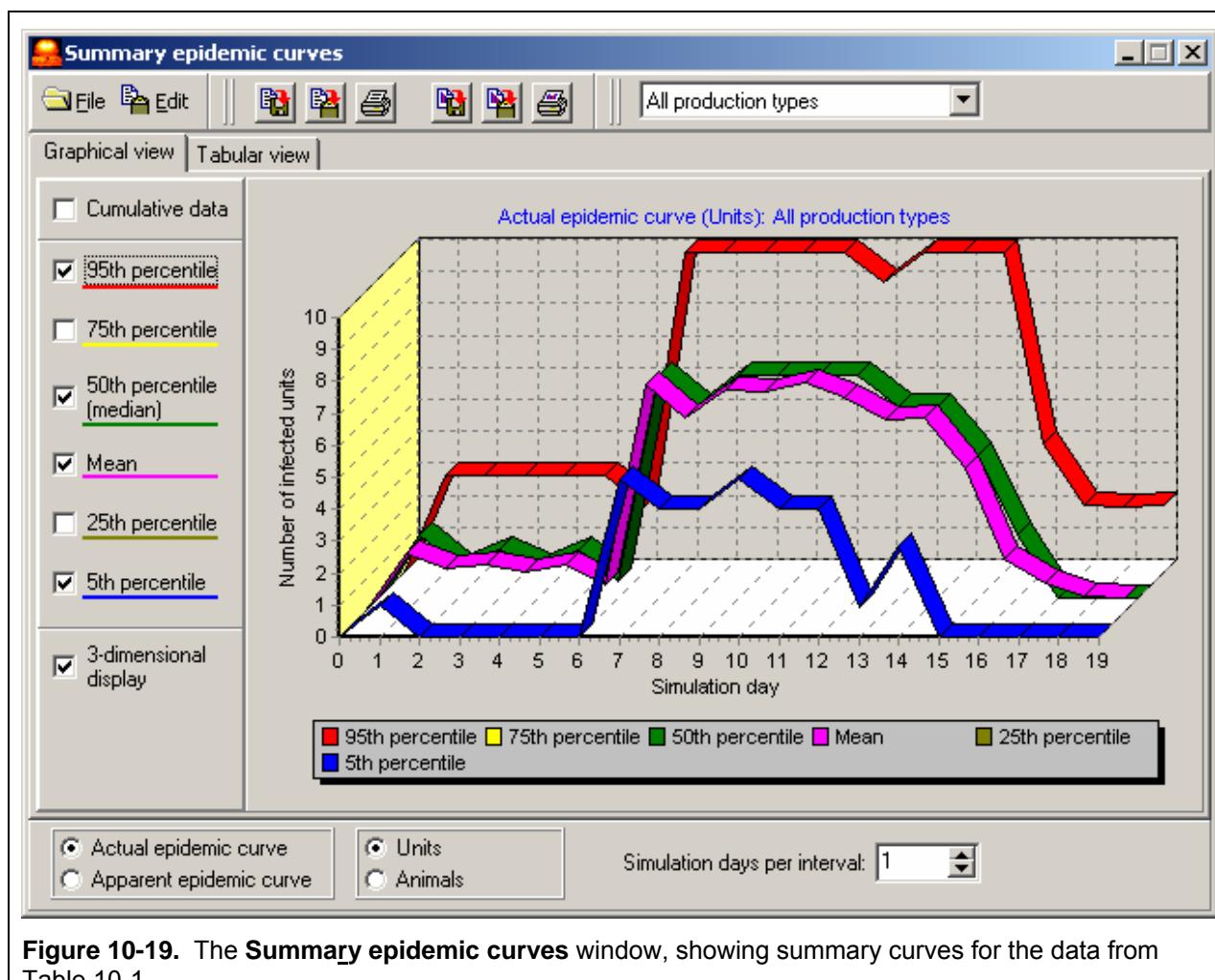


Figure 10-19. The **Summary epidemic curves** window, showing summary curves for the data from Table 10-1.

NOTE: Interpreting summary epidemic curves

Care should be used in interpreting these summary epidemic curves: while they closely resemble conventional epidemic curves, this resemblance is somewhat superficial. It pays to spend a little time thinking about how these curves are generated, and what information they actually convey.

10.4.2.2. Using the **Summary epidemic curves** graphical view

A graphical display of the summary epidemic curves is shown on the **Graphical view** tab of the **Summary epidemic curves** window (Figure 10-19). Select one or more of the check boxes on the left side of the window to choose which summary curve or curves (for the mean or various percentiles) to display. Check the **Cumulative data** box to display cumulative epidemic curves.

10. Viewing model output

The **Output windows** menu

Check the **3-dimensional display** box to toggle between a three-dimensional and a “flat” view of the graph.

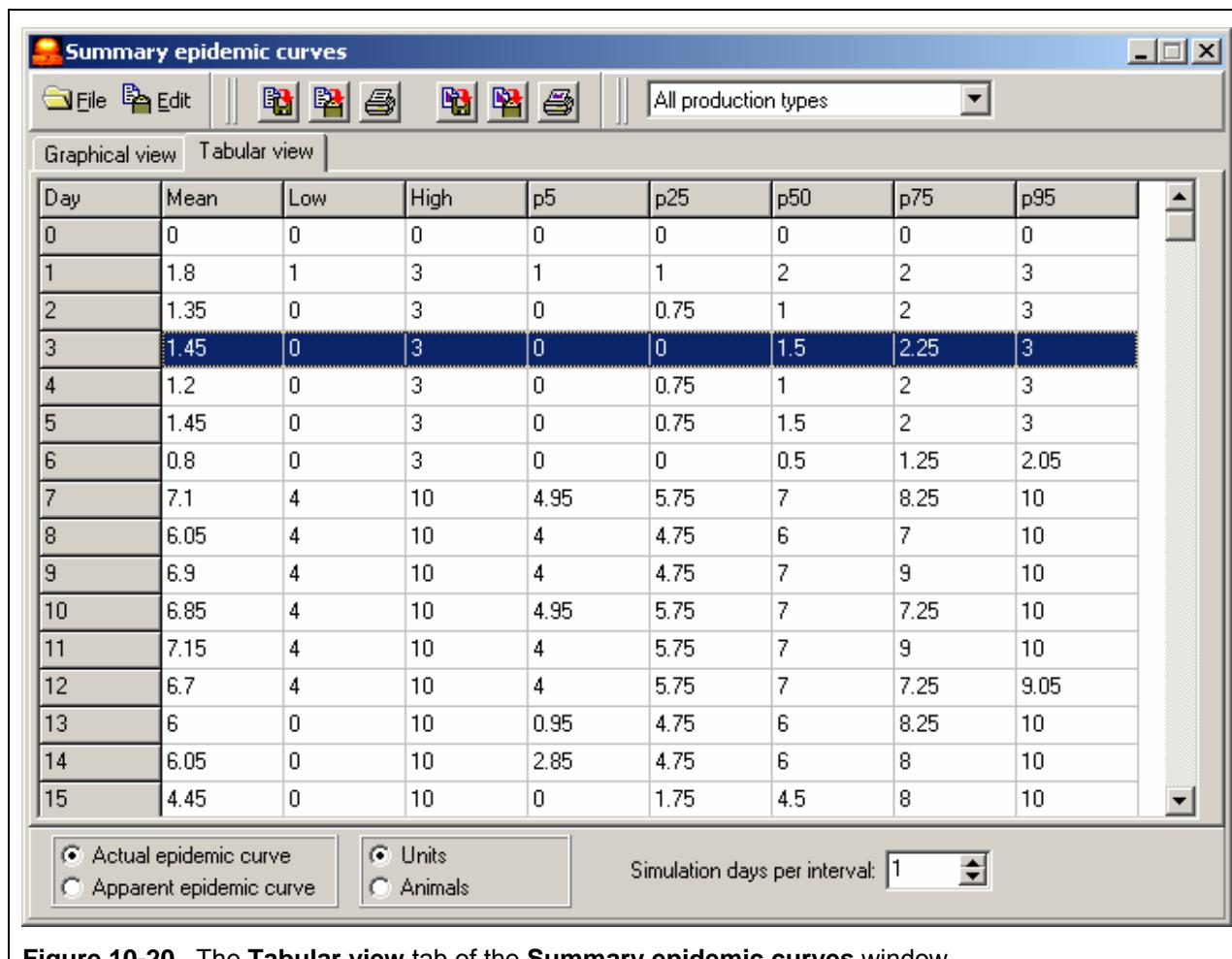
Users may choose to view summary curves for all disease cases (**Actual epidemic curve**) or for only detected cases (**Apparent epidemic curve**). Numbers of herds (units) or animals may be selected.

By default, each interval shown in the chart represents a single day. To smooth the curves, change the value in the **Simulation days per interval** box.

As with other output windows, the summary epidemic curve chart may be printed, saved to a file, or copied to the clipboard (see Section 10.5).

10.4.2.3. Using the **Summary epidemic curves** tabular view

The **Tabular view** tab of the **Summary epidemic curves** window (Figure 10-20) shows the actual numerical data used to generate the chart in the **Graphical view** tab. The options of this tab are very similar to those described above.



The screenshot shows a Windows application window titled "Summary epidemic curves". The window has a toolbar with icons for File, Edit, and various file operations like Open, Save, Print, and Copy. A dropdown menu shows "All production types". Below the toolbar is a tab bar with "Graphical view" and "Tabular view", where "Tabular view" is selected. The main area is a table with data rows from Day 0 to Day 15. The columns are labeled: Day, Mean, Low, High, p5, p25, p50, p75, and p95. The table shows values such as Day 0: Mean 0, Low 0, High 0; Day 1: Mean 1.8, Low 1, High 3; Day 3: Mean 1.45, Low 0, High 3; and so on up to Day 15: Mean 4.45, Low 0, High 10. At the bottom of the table are two radio buttons: "Actual epidemic curve" (selected) and "Apparent epidemic curve". Next are two radio buttons: "Units" (selected) and "Animals". Finally, there is a "Simulation days per interval" input field set to 1, with up and down arrows to its right.

Day	Mean	Low	High	p5	p25	p50	p75	p95
0	0	0	0	0	0	0	0	0
1	1.8	1	3	1	1	2	2	3
2	1.35	0	3	0	0.75	1	2	3
3	1.45	0	3	0	0	1.5	2.25	3
4	1.2	0	3	0	0.75	1	2	3
5	1.45	0	3	0	0.75	1.5	2	3
6	0.8	0	3	0	0	0.5	1.25	2.05
7	7.1	4	10	4.95	5.75	7	8.25	10
8	6.05	4	10	4	4.75	6	7	10
9	6.9	4	10	4	4.75	7	9	10
10	6.85	4	10	4.95	5.75	7	7.25	10
11	7.15	4	10	4	5.75	7	9	10
12	6.7	4	10	4	5.75	7	7.25	9.05
13	6	0	10	0.95	4.75	6	8.25	10
14	6.05	0	10	2.85	4.75	6	8	10
15	4.45	0	10	0	1.75	4.5	8	10

Figure 10-20. The **Tabular view** tab of the **Summary epidemic curves** window.

10.5. Common features of the *NAADSM* output windows

Most of the output windows discussed in the preceding sections have capabilities in common: for example, with the exception of the **Map of units for 1 iteration** window, all output screens have the ability to save, copy, or print data and graphics. The following sections describe these features, and point out the few exceptions to the general procedures they describe.

The typical output window has a toolbar similar the one shown in Figure 10-21. The production type dropdown menu is used to change the production type for which data is displayed (Section 10.5.1). The **File** menu, **Edit** menu, and toolbar buttons are used for saving, copying, and printing data or graphics (Sections 10.5.2 and 10.5.3).

10. Viewing model output

The **Output windows** menu

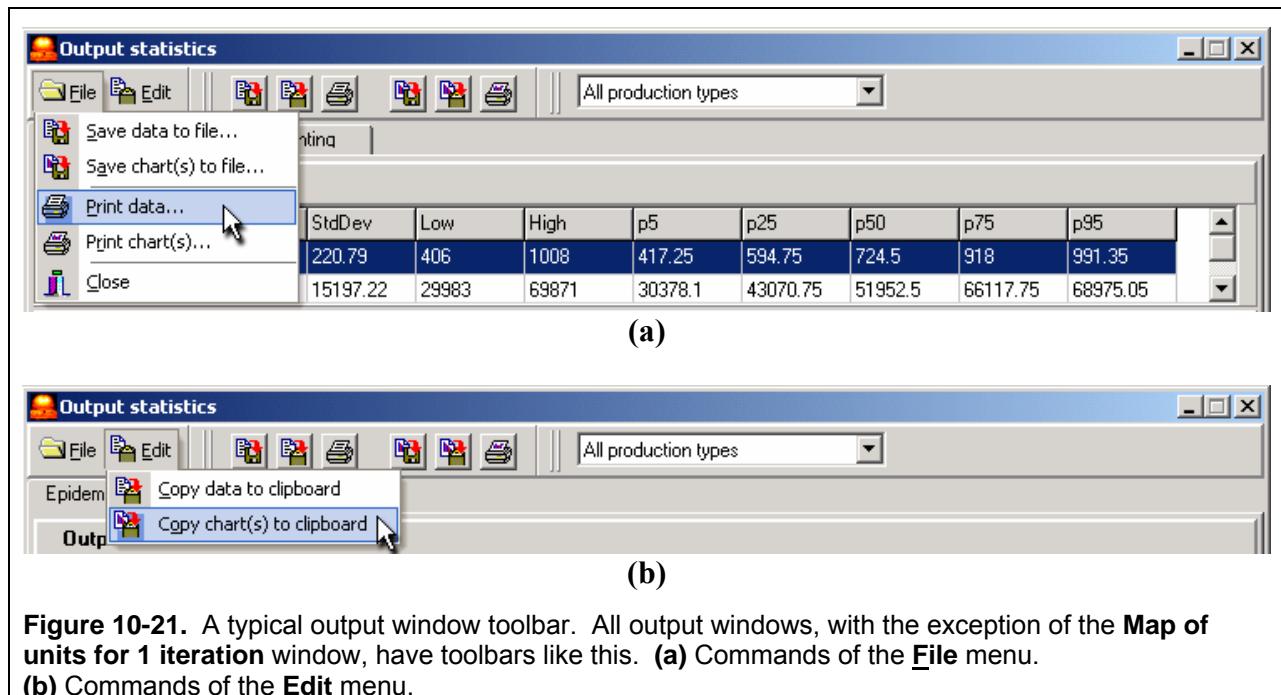


Figure 10-21. A typical output window toolbar. All output windows, with the exception of the **Map of units for 1 iteration** window, have toolbars like this. (a) Commands of the **File** menu.
(b) Commands of the **Edit** menu.

10.5.1. Changing the production type

By default, all output windows display data for all production types. To show only the data for a specific production type, simply change the selection in the production type dropdown menu.

The **Exposures for 1 iteration** window (Section 10.3.2) has two production type dropdown menus: the upper menu allows you to select a source production type for the displayed exposures, while the lower menu allows you to select a target (recipient) production type (Figure 10-12).

10.5.2. Saving, copying, or printing data

Output windows which display textual data or data in a tabular format have options for saving the displayed data to a file, copying it to the clipboard for use in another application, or printing the data.

To save data to a file, use **File → Save data to file...** or use the button and provide an appropriate file name when prompted. *NAADSM* always writes data files as comma-delimited plain text (*.csv format).

To copy data to the clipboard, use **Edit → Copy data to clipboard** or use the button.

To print data, use the **File → Print data...** command to select a printer, or click on the  button to print to your default printer.

For windows with multiple data tables (*e.g.*, the **Summary of 1 iteration window** described in Section 10.2.3), data from all selected tables will be saved, copied or printed. Change this selection by checking or unchecking the appropriate section boxes.

For windows with separate tabs for **Epidemiology** and **Cost accounting** outputs, text data from the visible tab will be saved, copied, or printed.

10.5.3. Saving, copying, or printing charts and graphs

Output windows which display graphical data have options for saving the displayed charts or graphs to a file, copying them to the clipboard for use in another application, or printing them.

To save graphics to a file, use the **File → Save chart(s) to file...** or click on the  button. Then provide an appropriate file name when prompted. *NAADSM* always writes graphical files in the *Windows* metafile (*.wmf) file format.

To copy graphics to the clipboard, use **Edit → Copy chart(s) to clipboard** or use the  button.

To print graphics, use **File → Print charts...** and select a printer, or click on  to use your default printer.

For windows with multiple charts or graphs (*e.g.*, the **Output statistics** window described in Section 10.4.1), all selected images will be saved, copied, or printed. Change this selection by checking or unchecking the appropriate section boxes.

For windows with separate tabs for **Epidemiology** and **Cost accounting** outputs, graphics from the visible tab will be saved, copied, or printed.

10.6. Arranging the output windows

The three commands at the bottom of the **Output windows** menu (Figure 10-1) provide convenient ways to organize multiple output windows on the screen. These commands are available only if two or more output windows are open at a time.

10.6.1. The **Cascade** command

Output windows → Cascade resizes each output window to use all available space, and arranges them so that their title bars are staggered from the upper left corner of the *NAADSM* window down (Figure 10-22a).

10. Viewing model output

The **Output windows** menu

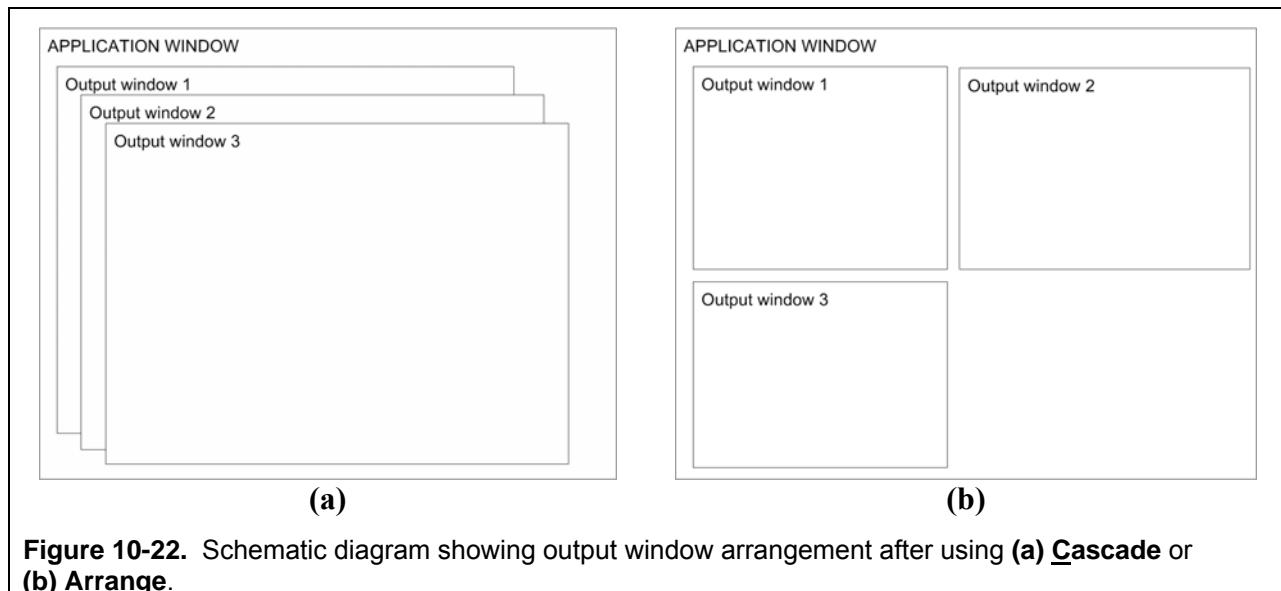


Figure 10-22. Schematic diagram showing output window arrangement after using (a) Cascade or (b) Arrange.

10.6.2. The Arrange command

Output windows → Arrange resizes each output window to use part of the available space in the *NAADSM* window so that all output windows are visible at once (Figure 10-22b).

10.6.3. The Close all windows command

Output windows → Close all windows (not surprisingly) closes all open output windows.

11. The **Help** menu

Two commands are given on the **Help** menu.

Help→About NAADSM... displays a screen showing version information for the *NAADSM* application and contact information for members of the development team (Figure 11-1).



Figure 11-1. The **About NAADSM** screen.

11. The **Help** menu

NOTE: NAADSM version numbers

The *NAADSM* version and build identifiers, as shown in Figure 11-1, convey a great deal of information. Each major version number (for example, “3.0”) is associated with one particular form of the conceptual model behind the application (see Appendix A). As the development team devises new approaches to disease modeling, the conceptual model changes, and new versions of the application are written. The minor version number (for example, the “.78” in “3.0.78”) is generally incremented when bugs are fixed or when new user interface features are introduced.

It is important to identify the version of *NAADSM* used to generate results, particularly if these results are to be used in publication: subtle (or in some cases, not-so-subtle) differences exist between versions.

Build numbers are less consequential: new build numbers are generally assigned when very minor tweaks are introduced to the graphical interface. If you wish to report a problem or an unusual behavior of the user interface (see Appendix G), it would be very helpful to include the build number as well as the version number in your report.

Notices of bug fixes, new features, and other changes made to *NAADSM* will be appropriately publicized, so that researchers and analysts who use the program are made aware of changes that may influence or affect their work.

Help → Go to the NAADSM website will open your default Internet browser and take you automatically to the *NAADSM* website, <<http://www.naadsm.org>>. The *NAADSM* website is the source for installation packages, source code, user documentation (including the latest version of this manual and any errata), and other useful materials for users of the *North American Animal Disease Spread Model*.

NOTE: Online help

Online help (the built-in help system found in many *Microsoft Windows* programs) is not available in *NAADSM*. It may be incorporated in future versions, depending on demand and availability of resources. Fortunately, we all have this delightful and entertaining manual to rely upon in the mean time!

Appendix A. *NAADSM 3.0* model description

Version 1.0.6

April 6, 2006

Written by Neil Harvey, Francisco J. Zagmutt-Vergara, Aaron Reeves, Mark Schoenbaum, and the *NAADSM* development team.

Contents of appendix A

Contents of appendix A	129
1. Introduction.....	130
Keywords	130
2. Basics.....	130
3. Disease.....	131
Disease parameters.....	132
4. Spread	132
4.1. Direct contact spread.....	132
Parameters for direct contact spread	135
Examples of direct contact spread	136
4.2. Indirect contact spread	137
Parameters for indirect contact spread	137
4.3. Airborne spread.....	138
Parameters for airborne spread	139
5. Detection	139
Detection parameters	140
6. Surveillance	140
6.1 Trace surveillance	140
Trace surveillance parameters.....	141
7. Control measures	142
7.2. Destruction.....	142
Destruction parameters	142
7.2.1 Destruction capacity.....	143
7.2.2 Destruction priorities	143
Examples of destruction priorities	144
7.3. Vaccination	145
Vaccination program parameters	145
7.3.1 Initiation of a vaccination program.....	146
7.3.2 Vaccination capacity	146
7.3.3 Vaccination priorities.....	146
7.3.4 Minimum time between vaccinations	147
8. Biological effects of vaccination.....	148
Parameters for effects of vaccination.....	148
9. Priorities of action.....	148
10. Costs	149
10.1 Costs associated with destruction	149
Parameters for destruction costs	149
10.2 Costs associated with vaccination.....	150
Parameters for vaccination costs.....	150

A1. Introduction

This document is intended to be a plain-language description of the simulation model implemented in the North American Animal Disease Spread Model v3.0. Its purpose is to facilitate agreement among current team members on details of the model, to provide a basis for functional testing, and to provide the validation committee and future team members with a complete but accessible description of the model.

The description is based on the paper *Modeling alternative mitigation strategies for a hypothetical outbreak of foot-and-mouth disease in the United States* by Mark A. Schoenbaum and W. Terry Disney, the document *SpreadModel Version 3.0 Diagrams and pseudocoding* by Mark A. Schoenbaum and Francisco Zagmutt-Vergara, observations of *SpreadModel* v2.14 and beta versions of 3.0, and discussions with the project team.

Keywords

herd-level stochastic spatial state-transition simulation

A2. Basics

A collection of animals, called a “unit,” is the basis of the simulation. A unit has a production type, size, location, and disease state. The production type may be a single kind of livestock (e.g., “dairy cattle”) or a mixed type (e.g., “sheep and poultry”). Figure A2-1 shows the states a unit may be in and possible transitions among them.

The simulation proceeds in time steps of one day. Each day units may be affected by biological processes happening in the animals (e.g. the natural progress of the disease), processes happening in the environment (e.g. airborne spread), and/or human actions (e.g. detection, vaccination, and destruction). The “model” is the sum of these processes and actions.

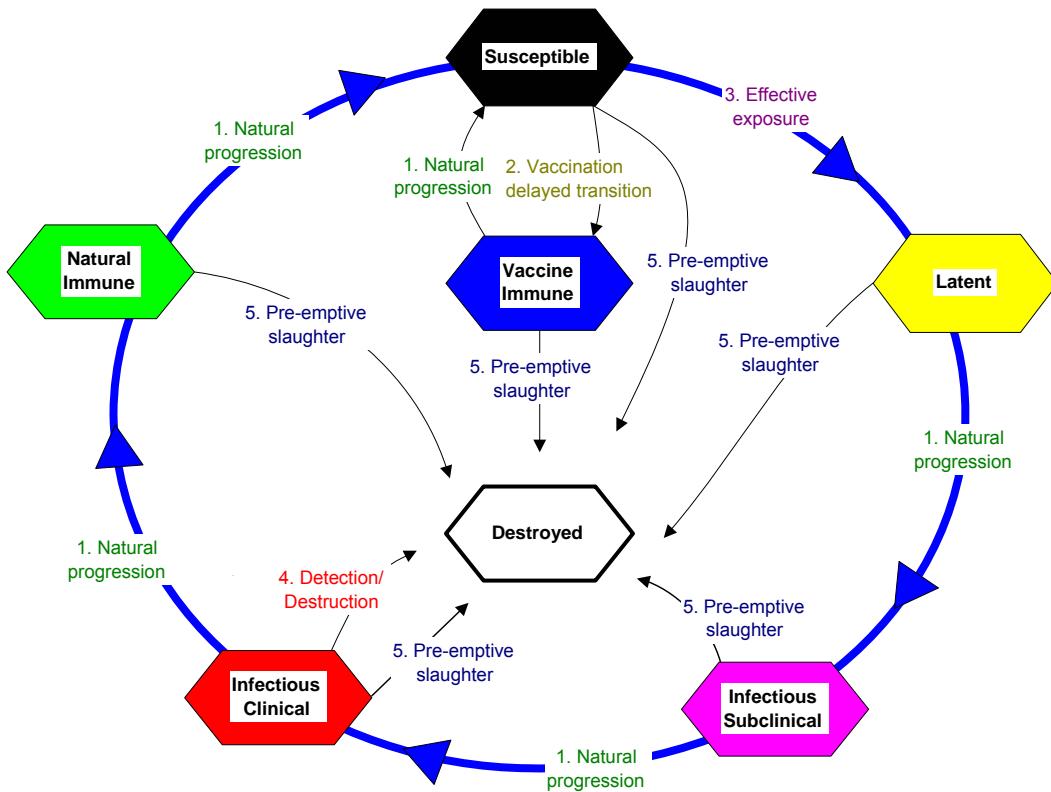


Figure A2-1. States and transitions

A3. Disease

When a Susceptible unit is infected, it becomes Latent. The infection progresses in the unit from Latent to Infectious Subclinical (shedding agent without visible signs of disease), to Infectious Clinical (shedding agent with visible signs of disease), to Natural Immune, and back to the Susceptible state. Probability functions characterize the length of the periods and this length is determined stochastically for each new infection. The disease is never fatal: that is, all infected units will eventually return to Susceptible unless destroyed. If time-frames for simulations are long, a particular unit may progress through the infected states more than once.

A unit can spend zero time in a state. For example, the parameter for time spent as Infectious Subclinical can be zero. In that case, units will change directly from Latent to Infectious Clinical. A unit undergoes its first transition state change on the day immediately following its infection.

Attempting to infect a unit that is not Susceptible has no effect.

If two units are at the same location, infecting one does not automatically infect the other.

Disease parameters

Parameters specified for each production type:

- latent period (days)  ³
- infectious subclinical period (days) 
- infectious clinical period (days) 
- natural immune period (days) 

The parameters are given separately for each production type. That is, the duration of the disease stages can be different for cattle, pigs, etc.

A4. Spread

A4.1. Direct contact spread

The simulation of direct contact – movement of animals among units – works as follows:

On each day,

1. Look up a multiplier to adjust the rate of movement of animals based on the number of days since the first detection of the disease. Use this multiplier to scale the movement rate. This approximates applying movement-controls over the course of an infection spreading through the population of units.
2. For each unit A ,
 - (a) Check whether A can be the source of an infection. That is, is it Latent, Infectious Subclinical, or Infectious Clinical, and not quarantined?⁴ (Infectious Clinical is always a source. Latent and Infectious Subclinical are optionally a source.)
 - (b) If A cannot be a source, go on to the next unit.
 - (c) Sample a number N from a Poisson distribution whose mean is the movement rate (adjusted by 1 above).
 - (d) Create N shipments from A .
3. For each shipment,
 - (a) Sample a number, $distance$, from the movement distance distribution.
 - (b) From all units that can be the target of disease exposure (that is, those that are not Destroyed or quarantined or are the source), choose the unit B whose distance from the source is closest to $distance$. If several possible targets are the same distance from the source, choose one randomly, giving preference to larger units (a unit with twice as many animals is twice as likely to be chosen).
 - (c) If no target B could be found, this shipment is dropped. Do not record an exposure; go on to the next shipment. This can happen if there are no units of the

³  indicates a parameter that is given as a probability density function.

⁴ See section A

A7. Control measures for a description of quarantine.

desired target production type or if all units of the desired target production type are Destroyed or quarantined.

- (d) If B is not Susceptible, the shipment has no effect on the disease state but is recorded as an exposure; go on to the next shipment.
- (e) Generate a random number r in $[0,1)$, that is, from 0 up to but not including 1.
- (f) If $r < P$, the probability of infection given exposure, turn B Latent after a shipping delay.

These steps are also illustrated as a flowchart in Figure A4-1, and examples are shown below.

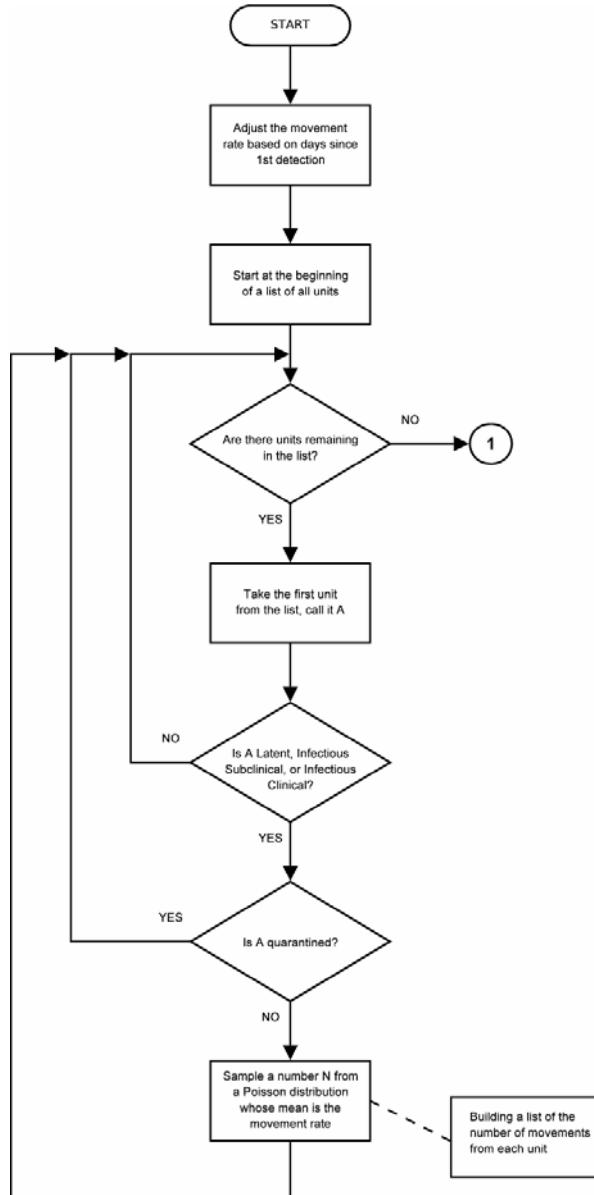


Figure A4-1: Generating direct contacts. The number in the circle links to the continued flowchart on the next page.

Appendix A. NAADSM model description

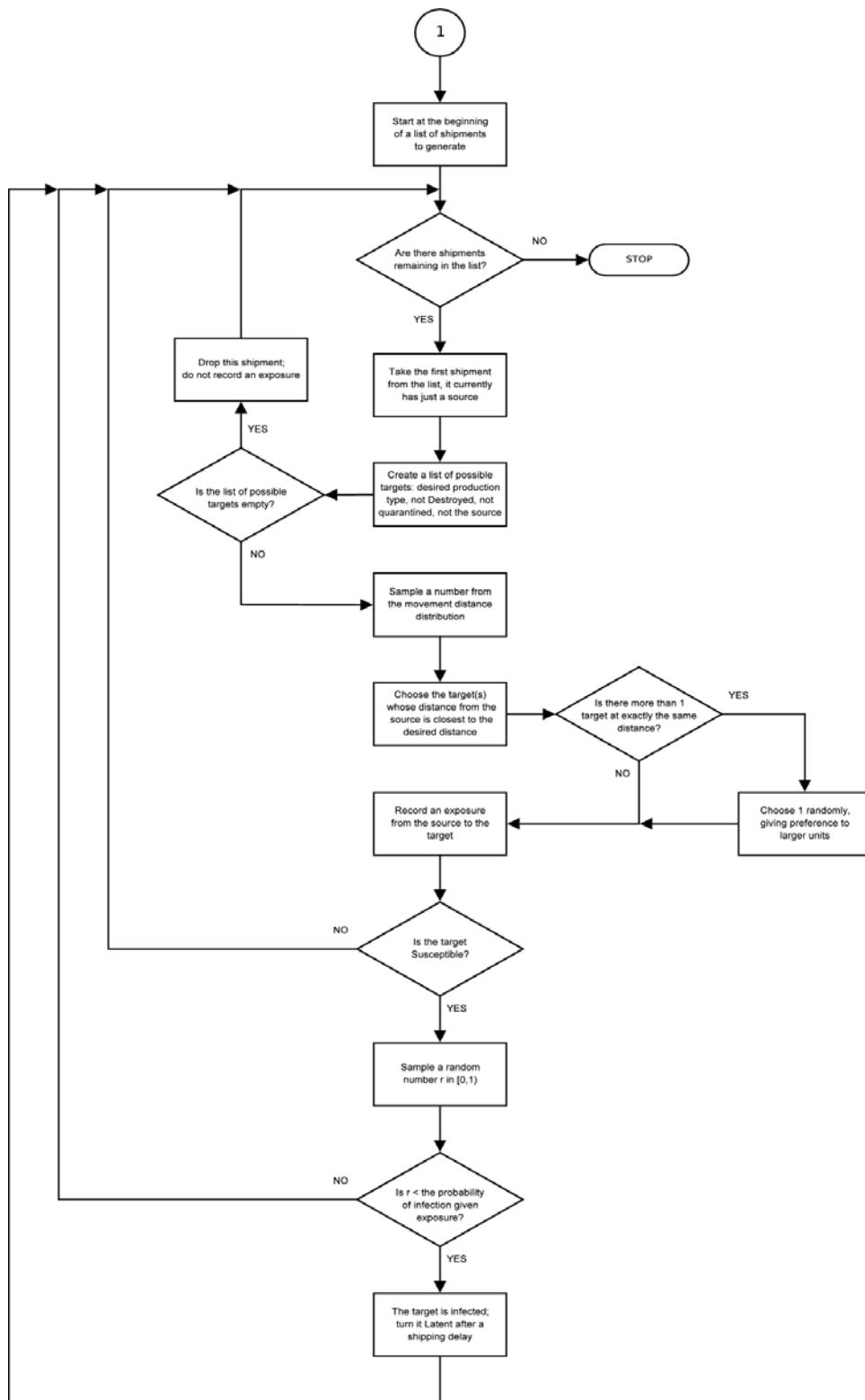


Figure A4-1 (cont'd): Generating direct contacts. The number in the circle continues the flowchart on the previous page.

The progress of the disease in the receiving unit starts at Latent, with the duration of each stage of the disease chosen stochastically, regardless of whether the shipping unit was Latent, Infectious Subclinical, or Infectious Clinical. A unit that receives Infectious Clinical animals could technically be regarded as immediately Infectious Clinical (able to infect other units by airborne spread and indirect contact, and detectable by a farmer or attending veterinarian) but starting the receiving herd at Latent reflects the fact that *most* of the animals in the receiving unit have to progress through the earlier disease-stages. The disease-state is an attribute of the unit as a whole rather than a direct reflection of the state of a particular animal in the unit.

Direct contacts (even ones that do not result in a new infection) are recorded and can be discovered later during trace-investigations.

The size of a shipment is not considered, and the number of animals in each unit does not change during the simulation.

The distance between lat_1, lon_1 and lat_2, lon_2 is approximated as:

$$y = lat_2 - lat_1$$

$$x = (lon_2 - lon_1) \cdot \cos(lat_1)$$

$$d = \frac{c}{360} \cdot \sqrt{x^2 + y^2}$$

where c = the circumference of the earth.

Parameters for direct contact spread

Parameters for each pair of production types:

- Mean rate of movement (recipient-units for shipments per source-unit per day)
- movement distance (km) 
- shipping delay (days) 
- probability of infection given exposure
- movement rate multiplier vs. days since the first detection  ⁵

When more than one production-type is considered, the above parameters are specified for each pairing of one production-type with another. Consider the production-types “Beef” and “Dairy”, referring to herds of beef cattle and dairy cattle. You specify the most important pairings of these production-types with regard to direct-contact parameters. For example: “Beef” to “Beef”, “Beef” to “Dairy”, “Dairy” to “Beef”, and “Dairy” to “Dairy” are the possible pairings of these production-types. Separate parameters are specified for each of these pairs since the direct-contact among different production-type may vary.

Note that parameters are separate for movement in each direction between each pair of production-types. That is, the parameters for movement from “Beef” to “Dairy” can be different from the parameters for movement from “Dairy” to “Beef”, and the parameters for movement from “Beef” to say “Pigs” can be different again.

⁵ indicates a parameter that is given as a relational chart.

Appendix A. NAADSM model description

If parameters are given for movements from “Beef” to “Beef” and from “Beef” to “Dairy”, the number of shipments a “Beef” cattle herd A sends to “Beef” cattle herds on a particular day and the number of shipments A sends to “Dairy” cattle herds on the same day are independent.

Shipping animals from a Latent, Infectious Subclinical, or Infectious Clinical unit to a Natural Immune or Vaccine Immune unit has no effect on the disease-stage of the recipient unit. Shipping animals from a unit in a more advanced disease stage to a unit in a less advanced disease stage (e.g., from an Infectious Clinical unit to a Latent unit) also has no effect.

Examples of direct contact spread

Example 1

As an example, suppose that shipments are being generated from “Beef” to “Dairy” units. In step 3 above, shipments are being generated from a particular source unit of production type “Beef”. Other units in the population are:

Unit 1. Swine unit, Susceptible, not quarantined, 25 km away
Unit 2. Dairy unit, Susceptible, not quarantined, 40 km away
Unit 3. Dairy unit, Susceptible, not quarantined, 300 km away

Suppose that in step 3a, the value “30 km” is sampled from the movement distance distribution. In step 3b, the possible target units are units 2 and 3. (Unit 1 is excluded because shipments from Beef units to Swine units are separate from and independent of shipments from Beef units to Dairy units. See notes on multiple production types below.) Unit 2 will be chosen because 40 km is closer than 300 km to the desired movement distance of 30 km.

Example 2

Suppose that the other units were instead:

Unit 1. Swine unit, Susceptible, not quarantined, 25 km away
Unit 2. Dairy unit, Infectious Clinical, not quarantined, 40 km away
Unit 3. Dairy unit, Susceptible, not quarantined, 300 km away

In this case, unit 2 will still be chosen. The fact that unit 2 is already diseased does not affect the decision.

Example 3

An example with a Destroyed unit:

Unit 1. Swine unit, Susceptible, not quarantined, 25 km away

Unit 2. Dairy unit, Destroyed, 40 km away

Unit 3. Dairy unit, Susceptible, not quarantined, 300 km away

In this case, the only possible target is unit 3, so it will be chosen. It does not matter that 300 km is much farther than the desired movement distance of 30 km, only that *of all possible targets* (in this case there is only one), 300 km is the best match.

Example 4

A final example:

Unit 1. Swine unit, Susceptible, not quarantined, 25 km away

Unit 2. Dairy unit, Destroyed, 40 km away

Unit 3. Dairy unit, Susceptible, quarantined, 300 km away

In this case, the shipment will be “dropped” because there is no possible target unit in the population.

A4.2. Indirect contact spread

Indirect contact – movement of people, materials, vehicles, equipment, animal products, etc among units – is simulated in the same manner as direct contact, except that only Infectious Subclinical and Infectious Clinical units, not Latent units, can be the source of infection. The parameters for indirect contact are similar to but independent of those for direct contact.

Indirect contacts can be discovered later during trace-investigations.

Parameters for indirect contact spread

Parameters for each pair of production types:

- Mean rate of movement (recipient-units for shipments per source-unit per day)
- movement distance (km) 
- shipping delay (days) 
- probability of infection given exposure
- movement rate multiplier vs. days since the first detection 

A4.3. Airborne spread

The simulation of airborne spread works as follows:

On each day,

1. For each unit A ,
 - (a) Check whether A can be the source of an infection. That is, is it Infectious Subclinical or Infectious Clinical?
 - (b) If A cannot be a source, go on to the next unit.
 - (c) For each other unit B ,
 - i. Check whether B can be the target of an infection. That is, is it Susceptible, is the distance from A to B < the maximum distance of spread, and is the direction from A to B inside the wind direction range?
 - ii. If B cannot be a target, go on to the next unit.
 - iii. Compute the probability of infection $P = \text{probability of infection at } 1 \text{ km} \times \text{DistanceFactor}(A,B) \times \text{HerdSizeFactor}(A) \times \text{HerdSizeFactor}(B)$.
 - iv. Generate a random number r in $[0,1)$.
 - v. If $r < P$, turn B Latent after a delay.

Where

$$\text{DistanceFactor}(A,B) = (\text{maximum distance of spread} - \text{distance from } A \text{ to } B) / (\text{maximum distance of spread} - 1)$$

$$\text{HerdSizeFactor}(A) = (\text{area under histogram of unit sizes from } 0 \text{ to size of } A) \times 2$$

The distance between lat_1, lon_1 and lat_2, lon_2 is approximated as before. The direction from lat_1, lon_1 to lat_2, lon_2 is approximated with the inverse tangent using the same x and y .

Airborne spread can occur from and to quarantined units.⁶

Airborne exposure is not recorded since it would not be used directly by mitigation processes such as movement-controls, vaccination, and destruction.

⁶See section A

A7. Control measures for a description of quarantine.

Parameters for airborne spread

Parameters for each pair of production types:

- probability of infection at 1 km from source (Infectious Subclinical or Infectious Clinical unit)
- wind direction, given as a range (*start* and *end*) in degrees
- maximum distance of spread (km)
- airborne transport delay (days) 

The parameters are given separately for spread in each direction between each pair of production types. That is, the parameters for spread from pig herds to cattle herds can be different from the parameters for spread from cattle herds to pig herds, to account for potential differences in amount of virus produced and/or different minimum-infective-doses for animals in different production-types.

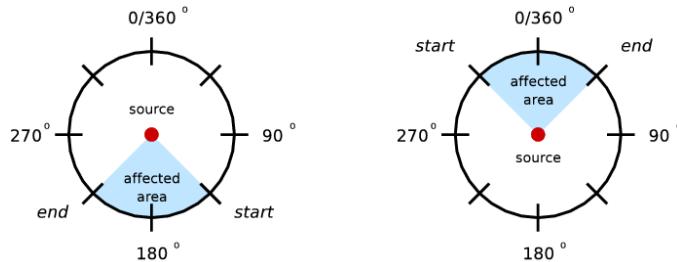


Figure A4-2. Example parameters for north winds (left) and south winds (right).

A5. Detection

The simulation of detection works as follows:

On each day,

1. Look up the probability that a farmer or attending veterinarian, for example, will report signs of disease to authorities based on the number of days since the first detection in the population. A nonzero static probability represents the baseline before the first detection.
2. For each Infectious Clinical unit,
 - (a) Look up the probability of detecting signs of disease based on the number of days the unit has been Infectious Clinical.
 - (b) Compute the probability of detection and reporting (Equation 1 X 2a).
 - (c) Generate a random number r in $[0,1]$.
 - (d) If $r < P$, the disease is detected and reported.

There are no false-positive detections.

A report is immediately known to the authorities.

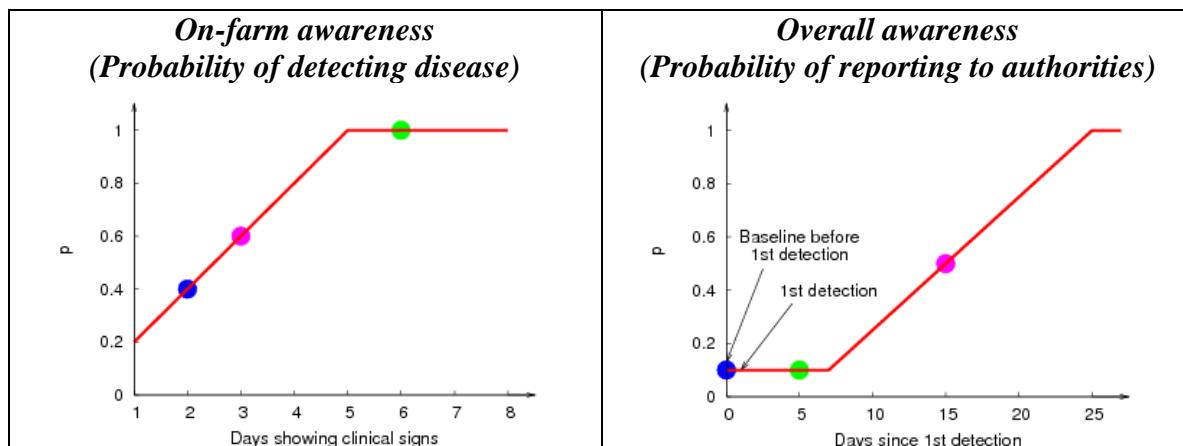


Figure A5-1. Probability of detection and reporting is found by two charts. In this example,

- before 1st detection, 2nd day of clinical signs, $P = 0.4 \times 0.1 = 0.04$
- 5 days since 1st detection, 6th day of clinical signs, $P = 1 \times 0.1 = 0.1$
- 15 days since 1st detection, 3rd day of clinical signs, $P = 0.6 \times 0.5 = 0.3$

Detection parameters

Parameters for each production type:

- probability of reporting vs. days since the first detection
- probability of detection vs. days the unit has been Infectious Clinical

The parameters are given separately for each production type, to account for the possibility that signs of disease may be more obvious in animals of certain production-types, e.g., signs may be reported more rapidly in intensive swine production systems versus cow-calf operations on pastures.

A6. Surveillance

Surveillance in the model refers to the process of identifying units at high risk for disease based upon exposure or (potentially) proximity to infected, detected units. Units identified by surveillance will be quarantined and thus can no longer spread disease by direct contact (see section A7.1).

Surveillance does not affect disease detection: that is, units subject to surveillance which become infected are no more likely to be detected than other units of the same production type.

A6.1 Trace surveillance

Units that have had contact with diseased units within a given number of days prior to detection of the diseased unit may be identified by trace investigations. Units subjected to surveillance will be quarantined. Optionally, units identified by trace surveillance may be preemptively destroyed (see section A7.2).

Trace-investigations are immediate. Tracing goes one level forward, that is, it identifies units that were recipients of direct or indirect contact from infected, detected units. Tracing does not identify contacts that led to the infection of infected, detected units (Figure A6-1).

Trace surveillance parameters

Parameters specified separately for every production type:

- probability of a trace-out investigation succeeding when direct contact has occurred
- period of interest for trace-out investigations of direct contacts
- probability of a trace-out investigation succeeding when indirect contact has occurred
- period of interest for trace-out investigations of indirect contacts

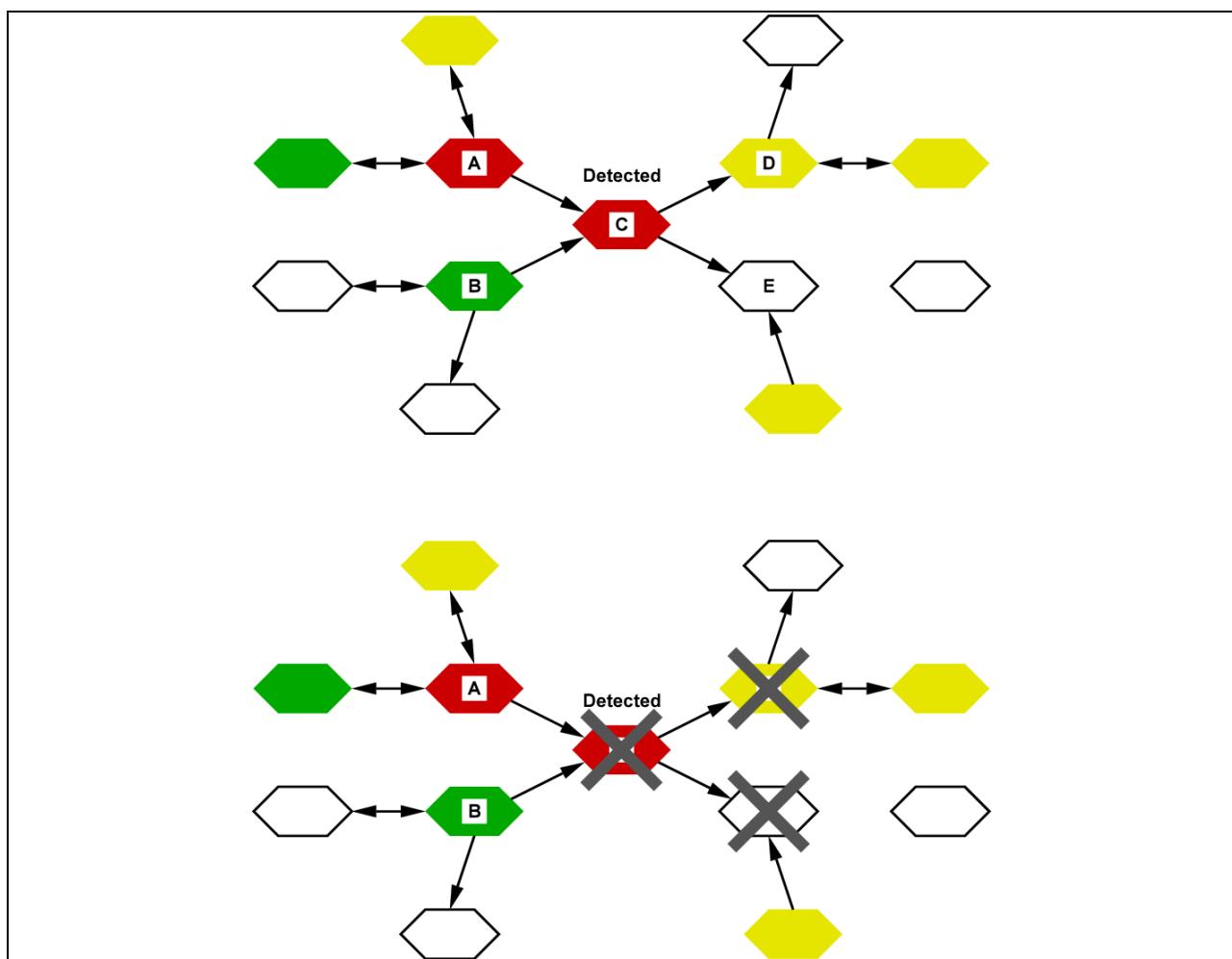


Figure A6-1. Trace out. When unit C is detected, units to which C has shipped animals or sent people or equipment are quarantined and may be marked for destruction. The trace does not extend further, e.g., to units that shipped animals to C (A or B), or units that received animals from D.

A7. Control measures

A7.1. Quarantine

A diseased unit is quarantined on the day immediately following its detection. Units are also quarantined when they are placed on the prioritized waiting list for destruction (see section A7.2.2). Quarantined units cannot be involved in direct contact, but indirect contact and airborne spread may still occur to or from a quarantined unit.

A7.2. Destruction

When the first detection happens in the study population, the authorities may initiate a destruction program. It can take several days before the authorities are ready to begin destroying.

All detected units are marked for destruction. Units that have had contact with diseased units within a given number of days prior to detection of the diseased unit (found through trace-investigations: see section A6.1) and units within a given distance of diseased units may also be marked for destruction. The destruction of these units associated by trace or distance has been called pre-emptive or dangerous-contact slaughter.

A production-type-specific parameter determines whether detection of an infected unit of a particular production type will trigger the formation of a destruction ring or not: for example, detection of an infected swine unit might lead to the destruction of surrounding units of various production types, while detection of an infected sheep unit might not trigger destruction of surrounding units.

A production-type-specific parameter also governs whether units of a particular production type are included in a destruction ring. For example, dairy cattle units might be destroyed in response to the detection of a diseased unit nearby, while sheep units might not be destroyed.

Destruction parameters

Global parameters (applied to all production types):

- delay to begin a destruction program (days)
- destruction capacity vs. days since the first detection (units per day)  (see section A7.2.1)
- destruction priorities (see section A7.2.2)

Parameters specified separately for every production type:

- indication of whether detection of units of the production type will trigger a

- destruction ring (yes/no)
- radius of destruction ring (km) , if units of the production type will trigger a destruction ring
- indication of whether units of the production type will be destroyed in response to detection of nearby units (yes/no)
- indication of whether units of this production type identified by trace surveillance after direct contact will be destroyed (yes/no)
- indication of whether units of this production type identified by trace surveillance after indirect contact will be destroyed (yes/no)

A7.2.1 Destruction capacity

There is a limit (called destruction capacity) on how many units can be destroyed per day. Destruction capacity does not consider unit size (*i.e.* the number of animals in each unit). Destruction capacity is specified as a relational chart of the number of units which can be destroyed per day versus the number of days since the first detection of disease. A single destruction capacity applies to units of all production types: for example, if the destruction capacity on a given day is 10 units, then 10 beef units may be destroyed on that day, or 10 swine units, or six units of one and four of the other, depending on the assigned destruction priorities (see section A7.2.2).

A7.2.2 Destruction priorities

If a unit is marked for destruction but cannot be destroyed immediately, it is quarantined and goes onto a prioritized waiting list.

There are three criteria which may be used to set destruction priorities: the production type of the unit, the reason for destruction of the unit, and the number of days a unit has been waiting in the destruction queue. Within the production type criterion, the production types present in a scenario are further prioritized (*e.g.* cattle may have a higher destruction priority than swine, or *vice versa*). Similarly, within the action reason criterion, the reasons for destruction are further prioritized: these reasons are detection of disease, exposure by direct contact, exposure by indirect contact, and presence within a specified destruction ring. For example, cattle herds that are marked for destruction because they were detected as diseased may have a higher priority than cattle herds that are marked for destruction because they are near a diseased unit.

The order in which the three criteria are applied must be specified in each scenario. For example, the number of days a unit has been in the destruction queue may be the overriding priority, so that units of any production type, holding for any reason, that have been holding for the longest period of time are destroyed before any others. Criteria with the highest priority are applied first. In the event that two units are encountered that have the same priority based on the top criterion, subsequent criteria are applied (see examples below).

Appendix A. NAADSM model description

No two production-type/reason for destruction combinations can have the same priority. That is, cattle herds that were detected as diseased *must* have a strictly higher priority than pig herds that were detected as diseased (or *vice versa*), and cattle herds that were detected as diseased *must* have a strictly higher priority than cattle herds that were simply near to a detected unit (or *vice versa*).

No distinction in destruction priority is made based on source of exposure: cattle herds that have been exposed by a swine herd, for example, are treated no differently than cattle herds exposed to infection by other cattle herds.

On each day, the authorities destroy as many units as possible (up to the destruction capacity for that day) from the destruction queue, beginning with the highest priority.

Examples of destruction priorities

Consider the following examples, using these four units which have been designated for destruction:

Unit A. Cattle herd, detected infection, holding for 3 days

Unit B. Cattle herd, indirect contact, holding for 5 days

Unit C. Swine herd, direct contact, holding for 1 day

Unit D. Swine herd, within circle/ring, holding for 5 days

Example 1:

With the following destruction priorities:

Days holding > production type (swine > cattle) > destruction reason (detected > direct > indirect > circle/ring)

The four herds are destroyed in the following order:

D, B, A, C

Example 2:

Priorities: production type (cattle > swine) > destruction reason (detected > direct > indirect > circle/ring) > days holding:

Destruction order: A, B, C, D

Example 3:

Priorities: production type (cattle > swine) > days holding > destruction reason (detected > direct > indirect > circle/ring):

Destruction order: B, A, D, C

Example 4:

Priorities: destruction reason (detected > circle/ring > direct > indirect) > production type (cattle > swine) > days holding:

Destruction order: A, D, C, B

A7.3. Vaccination

When the disease is detected, authorities may also initiate a vaccination campaign. This consists of vaccinating units within a specified distance of the detected units – in circles or rings around detected units. A production-type-specific parameter determines whether detection of an infected unit of a particular production type will trigger the formation of a vaccination ring or not: for example, detection of an infected swine unit might lead to the vaccination of surrounding units of various production types, while detection of an infected sheep unit might not trigger vaccination of surrounding units.

A production-type-specific parameter also governs whether units of a particular production type are included in a vaccination program. For example, dairy cattle units might be vaccinated in response to the detection of a diseased unit nearby, while sheep units might not be vaccinated.

The initiation of a vaccination program may be delayed until a certain trigger point is reached in terms of numbers of detected units (see section A7.3.1)

If a unit is marked for vaccination but cannot be vaccinated immediately, it goes onto a prioritized waiting list (see sections A7.3.2 and A7.3.3).

For a unit to receive multiple vaccinations, vaccination of that individual unit must be triggered multiple times (see section A7.3.4.). It is not currently possible to schedule revaccination of units without an additional trigger.

Vaccination program parameters

Global parameters (applied to all production types):

- number of detected units before vaccination begins (see section A7.3.1)
- vaccination capacity vs. days since the first detection (units per day)  (see section A7.3.2)
- vaccination priorities (see section A7.3.3)

Parameters set individually for each production type:

- indication of whether detection of units of the production type will trigger a vaccination ring (yes/no)
- radius of vaccination ring (km) , if units of the production type will trigger a vaccination ring
- indication of whether units of the production type will be vaccinated in response to

Appendix A. NAADSM model description

- detection of nearby units (yes/no)
- minimum time between vaccinations (days) , if units of the production type will be vaccinated (see section A7.3.4)

A7.3.1 Initiation of a vaccination program

A vaccination program is initiated when the user-specified number of infected units has been detected. Until or unless this number is reached, units are not marked for vaccination. Once this critical number has been reached, units within the specified vaccination ring surrounding the most recently detected unit are marked for vaccination. Vaccination rings also will be created around any unit that is detected on the same simulation day that the critical number is reached. Similarly, vaccination rings will be created around infected units detected on subsequent simulation days. Units marked for vaccination are then treated according to the steps described below.

A7.3.2 Vaccination capacity

Vaccination capacity (the number of units which can be vaccinated per day) is handled in the same way as destruction capacity (see section A7.2.1), and is specified as a relational chart of the number of units which can be vaccinated per day versus the number of days since the first detection of disease. Vaccination capacity does not consider unit size (*i.e.* the number of animals in each unit). A single vaccination capacity applies to units of all production types.

Personnel for destruction cannot be temporarily loaned to vaccination teams, or *vice versa*, during a simulation run. In other words, the daily limits for destruction and vaccination operate independently of one another.

A7.3.3 Vaccination priorities

If a unit is marked for vaccination but cannot be vaccinated immediately, it goes onto a prioritized waiting list.

Vaccination priorities are set in fashion similar to destruction priorities (see section A7.2.2). There are two criteria which may be used to set vaccination priorities for units which fall within a vaccination circle. These criteria are production type of the unit and the number of days that a unit has been in the vaccination queue. Within the production type criterion, the production types present in a scenario are further prioritized.

The order in which these two criteria are applied must be specified in each scenario. For example, the number of days a unit has been in the vaccination queue may be the overriding priority, so that units of any production type that have been holding for the longest period of time are vaccinated before any others. The criterion with the highest priority is applied first. In the

event that two units are encountered that have the same priority based on the top criterion, the next criterion is applied.

A7.3.4 Minimum time between vaccinations

The minimum time between vaccinations is the number of days which must pass before a unit may be revaccinated. Once the specified number of days has passed, a unit may be revaccinated if vaccination of that unit is triggered again.

Consider the simple situation involving units *A*, *B*, and *C* as shown Figure A7-1. Disease is detected in unit *A* ten days before disease is detected in unit *C*. Both detections trigger vaccination circles as shown.

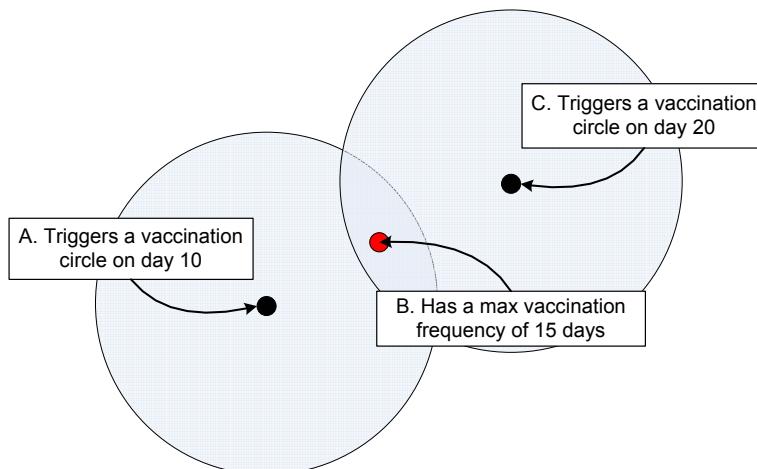


Figure A7-1. Overlapping vaccination rings. See text for description of timing of vaccinations.

Unit *B* is within vaccination circles triggered by detection of units *A* and *C*, and will be added twice to the queue of units to be vaccinated. If there is no waiting period for vaccination (*i.e.* vaccination capacity is not reached), unit *B* will receive only one vaccination: its minimum time between vaccinations will not have been reached before it comes to the head of the queue the second time.

If vaccination capacity has been reached, unit *B* will receive two vaccinations only if the elapsed time between the first and second scheduled vaccinations exceeds the unit's minimum time between vaccinations. This subsequent vaccination resets the vaccine-immune period for unit *B*. If the elapsed time is less than the unit's maximum vaccination frequency, unit *B* will not be revaccinated.

A8. Biological effects of vaccination

When a unit is vaccinated, it remains Susceptible for a time while immunity develops, then becomes Vaccine Immune. The length of the immune period is determined stochastically for each new vaccination. After the immune period, the unit reverts to a Susceptible state.

If a unit is infected after being vaccinated but before turning Vaccine Immune, the effects of the vaccination are cancelled.

Vaccinating a unit that is not Susceptible has no effect on its disease state.

Parameters for effects of vaccination

Parameters specified for each production type:

- delay to produce immunity (days)
- immunity period (days) 

The parameters are given separately for each production type.

A9. Priorities of action

Because the events in one simulation day should be considered to happen simultaneously, and because different processes may try to make conflicting changes to a unit, there is a need to order or prioritize the processes.

The ordering is:

1. Infection, destruction, or vaccination
2. Biological processes happening within units

(Note that these correspond to the transition labels in Figure A2-1.)

If a unit is both infected or vaccinated, or infected and destroyed, the order in which these happen is chosen randomly. If a unit is to be vaccinated and destroyed on the same day, destruction will always have precedence. If all three are scheduled to occur on the same day, a unit may or may not be infected before it is destroyed, but it will never be vaccinated.

If two or more processes infect the same unit on the same day, one process is chosen randomly as the cause of the infection, for the purpose of reporting in simulation statistics. Similarly, if there are two or more reasons for vaccinating or destroying a unit, one reason is chosen randomly for the purpose of reporting.

Some examples illustrating the effects of this ordering:

- If unit A is due to change from Susceptible to Vaccine Immune on day D , and a shipment of infectious animals arrives on day D , A is infected. (The exposure happens

- “before” the natural progression to Vaccine Immune.)
- If unit A is due to change from Vaccine Immune to Susceptible on day D , and the wind carries virus from an infectious unit on day D , A is not infected. (The exposure happens “before” the natural progression to Susceptible.)
- If unit A is destroyed on day D , and the wind carries virus from an infectious unit on day D , A may be reported in simulation statistics as having been an infected unit or a healthy unit. (Whether the infection happens “before” the destruction is determined randomly.)

A10. Costs

Direct costs associated with destruction and vaccination during an outbreak may be calculated.

A10.1 Costs associated with destruction

There is a fixed cost associated with appraisal of each destroyed unit, regardless of the number of animals in the unit. The cost associated with cleaning and disinfection each unit is also fixed regardless of the number of animals in each unit.

Beyond these fixed per-unit costs, the per-animal costs for euthanasia, carcass disposal, and indemnification apply.

The total cost of destruction *for each unit* of a particular production type is calculated as follows:
 (Appraisal cost + Cleaning and disinfection cost)
 $+ [(\text{Number of animals in the unit}) \times (\text{Cost of euthanasia} + \text{Cost of indemnification} + \text{Cost of disposal})]$

The total cost of destruction *for each production type* is calculated as:

$$\begin{aligned} & (\text{Number of units destroyed}) \times (\text{Appraisal cost} + \text{Cleaning and disinfection cost}) \\ & + [(\text{Total number of animals destroyed}) \times (\text{Cost of euthanasia} + \text{Cost of indemnification} \\ & \quad + \text{Cost of disposal})] \end{aligned}$$

Parameters for destruction costs

Parameters specified for each production type:

- Appraisal cost **per unit**
- Cost of cleaning and disinfection **per unit**
- Cost of euthanasia **per animal**
- Cost of indemnification **per animal**
- Cost of carcass disposal **per animal**

A10.2 Costs associated with vaccination

There is a fixed cost associated with vaccination set up for each vaccinated unit, regardless of the number of animals in the unit. The cost of vaccination of each animal in the unit is added to this fixed unit cost.

The cost of vaccination of each animal will depend on the total number of animals vaccinated. For each animal up to a specified threshold, only a baseline vaccination cost applies. For each animal over this threshold, an additional cost applies.

The total cost of vaccination *for each production type* is calculated as follows:

If the threshold is not reached:

$$[(\text{Number of units vaccinated}) \times (\text{Cost of site setup})] \\ + [(\text{Total number of animals vaccinated}) \times (\text{Baseline cost per animal})]$$

If the threshold is reached:

$$[(\text{Number of units vaccinated}) \times (\text{Cost of site setup})] \\ + [(\text{Threshold level}) \times (\text{Baseline cost per animal})] \\ + [(\text{Total number of animals vaccinated} - \text{Threshold level}) \times (\text{Baseline cost per animal} \\ + \text{Additional cost per animal})]$$

Parameters for vaccination costs

Parameters specified for each production type:

- Number of animals of this production type that can be vaccinated before the cost of vaccination increases
- Baseline cost of vaccination **per animal** (this cost applies until the specified threshold has been met)
- Additional cost of vaccination **per animal** for each animal beyond the specified threshold
- Cost of vaccination site setup per **unit**

Appendix B. NAADSM outputs

B1. Summary outputs for each iteration

At the end of each iteration, a set of values summarizing the events of that iteration are available: these include results such as the total number of units or animals that were infected over the course of that iteration, the total cost associated with destruction for that iteration, *etc.* Summary outputs for each iteration include epidemiological outputs (Table B-1) and cost accounting outputs (Table B-2).

The **Output statistics** window (see Section 10.4) displays various statistics calculated for all of these values across all iterations. The raw data used to generate those statistics for epidemiological outputs is stored in the scenario database table **outIterationByProductionType**. The raw data for cost accounting outputs is stored in the database table **outIterationCosts** (see Appendix E).

Table B-1. Summary epidemiological outputs for each iteration. Outputs are listed in the order in which they are stored in the database and presented in the **Output statistics** window.

Output name	Description
tscUSusc ¹	(transition state cumulative Units Susceptible) Total number of units that were initially susceptible and those that became susceptible over the course of the iteration.
tscASusc ¹	(transition state cumulative Animals Susceptible) Total number of animals in units as described immediately above.
tscULat ¹	(transition state cumulative Units Latent) Total number of units that were initially latent and those that became latent over the course of the iteration.
tscALat ¹	(transition state cumulative Animals Latent) Total number of animals in units as described immediately above.
tscUSubc ¹	(transition state cumulative Units Subclinical) Total number of units that were initially subclinical and those that became subclinical over the course of the iteration.
tscASubc ¹	(transition state cumulative Animals Subclinical) Total number of animals in units as described immediately above.
tscUClin ¹	(transition state cumulative Units Clinically infectious) Total number of units that were initially clinical and those that became clinical over the course of the iteration.
tscAClin ¹	(transition state cumulative Animals Clinically infectious) Total number of animals in units as described immediately above.

Appendix B. NAADSM outputs

Table B-1. Summary epidemiological outputs for each iteration. (*continued*)

Output name	Description
tscUNImm ¹	(transition state cumulative Units Naturally Immune) Total number of units that were initially naturally immune and those that became naturally immune over the course of the iteration.
tscANImm ¹	(transition state cumulative Animals Naturally Immune) Total number of animals in units as described immediately above.
tscUVImm ¹	(transition state cumulative Units Vaccine Immune) Total number of units that were initially vaccine immune and those that became vaccine immune over the course of the iteration.
tscAVImm ¹	(transition state cumulative Animals Vaccine Immune) Total number of animals in units as described immediately above.
tscUDest ²	(transition state cumulative Units Destroyed) Total number of units that were initially in the “destroyed” state and those that were destroyed during the course of the iteration.
tscADest ²	(transition state cumulative Animals Destroyed) Total number of animals in units as described immediately above.
infcUIni	(infections cumulative Units Initially infected) Number of units that were initially infected at the beginning of the iteration.
infcAIni	(infections cumulative Animals Initially infected) Number of animals in initially infected units at the beginning of the iteration.
infcUAir	(infections cumulative Units Airborne spread) Total of the number of units that became infected by airborne spread over the course of the iteration.
infcAAir	(infections cumulative Animals Airborne spread) Total number of animals in units that became infected by airborne spread over the course of the iteration.
infcUDir	(infections cumulative Units Direct contact) Total number of units that became infected by direct contact over the course of the iteration.
infcADir	(infections cumulative Animals Direct contact) Total number of animals in units that became infected by direct contact over the course of the iteration.
infcUInd	(infections cumulative Units Indirect contact) Total number of units that became infected by indirect contact over the course of the iteration.

Table B-1. Summary epidemiological outputs for each iteration. (*continued*)

Output name	Description
infcAInd	(<u>infections cumulative</u> <u>Animals</u> <u>Indirect contact</u>) Total number of animals in units that became infected by indirect contact over the course of the iteration.
expcUDir	(<u>exposures cumulative</u> <u>Units</u> <u>Direct contact</u>) Total number of units directly exposed to any infected unit over the course of the iteration.
expcADir	(<u>exposures cumulative</u> <u>Animals</u> <u>Direct contact</u>) Total number of animals in units as described immediately above.
expcUInd	(<u>exposures cumulative</u> <u>Units</u> <u>Indirect contact</u>) Total number of units indirectly exposed to any infected unit over the course of the iteration.
expcAInd	(<u>exposures cumulative</u> <u>Units</u> <u>Indirect contact</u>) Total number of animals in units as described immediately above.
trcUDir	(<u>trace cumulative</u> <u>Units</u> <u>Direct contact</u>) Total number of units directly exposed and successfully traced over the course of the iteration.
trcADir	(<u>trace cumulative</u> <u>Animals</u> <u>Direct contact</u>) Total number of animals in units as described immediately above.
trcUInd	(<u>trace cumulative</u> <u>Units</u> <u>Indirect contact</u>) Total number of units indirectly exposed and successfully traced over the course of the iteration.
trcAInd	(<u>trace cumulative</u> <u>Animals</u> <u>Indirect contact</u>) Total number of animals in units as described immediately above.
trcUDirp	(<u>trace cumulative</u> <u>Units</u> <u>Direct contact possible</u>) Total number of units directly exposed that could possibly have been traced over the course of the iteration.
trcADirp	(<u>trace cumulative</u> <u>Animals</u> <u>Direct contact possible</u>) Total number of animals in units as described immediately above.
trcUIndp	(<u>trace cumulative</u> <u>Units</u> <u>Indirect contact possible</u>) Total number of units indirectly exposed that could possibly have been traced over the course of the iteration.
trcAIndp	(<u>trace cumulative</u> <u>Animals</u> <u>Indirect contact possible</u>) Total number of animals in units as described immediately above.

Appendix B. NAADSM outputs

Table B-1. Summary epidemiological outputs for each iteration. (*continued*)

Output name	Description
detcUClin	(detection cumulative Units Clinical signs) Total number of clinical units detected over the course of the iteration.
detcAClin	(detection cumulative Animals Clinical signs) Total number of animals in detected clinical units over the course of the iteration.
descUIni	(destructions cumulative Units Initially destroyed) Total number of units in the “destroyed” state at the outset of the iteration (i.e., units destroyed prior to the point in time when the simulation began).
descAIni	(destructions cumulative Animals Initially destroyed) Total number of animals in units that were in the “destroyed” state at the outset of the iteration.
descUDet	(destructions cumulative Units Detected by clinical signs) Total number of infected units destroyed because they were detected based on clinical signs over the course of the iteration.
descADet	(destructions cumulative Animals Detected by clinical signs) Total number of animals in units destroyed for the reason listed immediately above.
descUDir	(destructions cumulative Units Direct contact) Total number of units destroyed because of direct contact with an infected unit over the course of the iteration.
descADir	(destructions cumulative Animals Direct contact) Total number of animals in units destroyed for the reason listed immediately above.
descUInd	(destructions cumulative Units Indirect contact) Total number of units destroyed because of indirect contact with an infected unit over the course of the iteration.
descAInd	(destructions cumulative Animals Indirect contact) Total number of animals in units destroyed for the reason listed immediately above.
descURing	(destructions cumulative Units Ring destruction) Total number of units destroyed because they were within a destruction ring over the course of the iteration.
descARing	(destructions cumulative Animals Ring destruction) Total number of animals in units destroyed for the reason listed immediately above.

Table B-1. Summary epidemiological outputs for each iteration. (*continued*)

Output name	Description
vaccUIni	(<u>vaccination</u> <u>cumulative</u> <u>Units</u> <u>Initially vaccinated</u>) Total number of units in the “vaccine immune” state at the outset of the iteration (<i>i.e.</i> , units vaccinated prior to the point in time when the simulation began).
vaccAIni	(<u>vaccination</u> <u>cumulative</u> <u>Animals</u> <u>Initially vaccinated</u>) Total number of animals in units that were in the “vaccine immune” state at the outset of the iteration.
vaccURing	(<u>vaccination</u> <u>cumulative</u> <u>Units</u> <u>Ring</u>) Total number of units vaccinated because they were within a vaccination ring over the course of the iteration.
vaccARing	(<u>vaccination</u> <u>cumulative</u> <u>Animals</u> <u>Ring</u>) Total number of animals in units vaccinated for the reason listed immediately above.
<u>firstDetection</u>	Day of first detection of an infected unit of the specified production type in the iteration.
<u>firstDestruction</u>	Day of first destruction of a unit of the specified production type in the iteration.
<u>firstVaccination</u>	Day of first vaccination of a unit of the specified production type in the iteration.

¹ Since it is possible for a unit to repeat the disease cycle within a single iteration, these values may exceed the number of units/animals in the population.

² It is not possible for a unit to be destroyed more than once in a single iteration. Consequently, these values can never exceed the number of units/animals in the population.

Appendix B. NAADSM outputs

Table B-2. Summary cost accounting outputs for each iteration. Outputs are listed in the order in which they are stored in the database and presented in the **Output statistics** window.

Output name	Description
destrAppraisal	Total cost associated with unit appraisal over the course of the iteration.
destrCleaning	Total cost of cleaning and disinfection over the course of the iteration.
destrEuthanasia	Total cost of euthanasia over the course of the iteration.
destrIndemnification	Total cost of indemnification over the course of the iteration.
destrDisposal	Total cost of carcass disposal over the course of the iteration.
vaccSetup	Total cost associated with vaccination site setup over the course of the iteration.
vaccVaccination	Total cost of vaccinating animals over the course of the iteration.

B2. Additional characteristics of each iteration

The length in days of each iteration is recorded. *NAADSM* also records, for each iteration, whether the simulated outbreak ended. [It is possible for iterations to end before the simulated outbreak ends: for example, if the user chooses to run each iteration for 50 days (see Section 9), the iteration may end before the outbreak is over.] These values are recorded in the scenario database table **outIteration** (see Appendix E).

B3. Daily outputs

Outputs generated on each day of the last three iterations are recorded in the scenario file database. Users have the option of storing daily outputs for more than three iterations (see Section 8.12).

Many of the daily outputs have the same name and share a similar purpose with those listed among the summary outputs. For example, the daily output **infcUDir** records the running total number of units infected by direct contact. The summary output **infcUDir** is the total number of units infected by direct contact over the course of an iteration. The summary output will always be equal to its homologous daily output for the last day of the iteration: in other words, on the last day of an iteration, the running total number of infected units *is* the total number of units infected over the course of that entire iteration.

Daily epidemiological outputs for each production type (Table B-3) are recorded in the database table **outDailyByProductionType** (see Appendix E). Daily cost accounting outputs are not recorded, but are calculated as needed with formulas described in Section 8.11.

Table B-3. Daily epidemiological outputs. Outputs are listed in the order in which they are stored in the database.

Output name	Description
tsdUSusc ³	(transition state daily Units Susceptible) Number of units that are susceptible on the indicated day.
tsdASusc ³	(transition state daily Animals Susceptible) Total number of animals in susceptible units on the indicated day.
tsdULat ³	(transition state daily Units Latent) Number of units that are latent on the indicated day.
tsdALat ³	(transition state daily Animals Latent) Total number of animals in latent units on the indicated day.
tsdUSubc ³	(transition state daily Units Subclinical) Number of units that are subclinically infectious on the indicated day.
tsdASubc ³	(transition state daily Animals Subclinical) Total number of animals in subclinically infectious units on the indicated day.
tsdUClin ³	(transition state daily Units Clinically infectious) Number of units that have clinical signs on the indicated day.
tsdAClin ³	(transition state daily Animals Clinically infectious) Total number of animals in clinically infectious units on the indicated day.
tsdUNImm ³	(transition state daily Units Naturally Immune) Number of units that are naturally immune on the indicated day.
tsdANImm ³	(transition state daily Animals Naturally Immune) Total number of animals in naturally immune units on the indicated day.
tsdUVIImm ³	(transition state daily Units Vaccine Immune) Number of units that are vaccine immune on the indicated day.
tsdAVIImm ³	(transition state daily Animals Vaccine Immune) Total number of animals in vaccine immune units on the indicated day.
tsdUDest ³	(transition state daily Units Destroyed) Number of units in the “destroyed” state on the indicated day (see note 3 below).
tsdADest ³	(transition state daily Animals Destroyed) Total number of animals in destroyed units on the indicated day (see note 3 below).
tscUSusc ⁴	(transition state cumulative Units Susceptible) Running total number of units that are initially susceptible and those that have become susceptible.
tscASusc ⁴	(transition state cumulative Animals Susceptible) Running total number of animals in units as described immediately above.

Appendix B. NAADSM outputs

Table B-3. Daily epidemiological outputs. (*continued*)

Output name	Description
tscULat ⁴	(transition state cumulative Units Latent) Running total number of units that are initially latent and those that have become latent.
tscALat ⁴	(transition state cumulative Animals Latent) Running total number of animals in units as described immediately above.
tscUSubc ⁴	(transition state cumulative Units Subclinical) Running total number of units that are initially subclinical and those that have become subclinical.
tscASubc ⁴	(transition state cumulative Animals Subclinical) Running total number of animals in units as described immediately above.
tscUClin ⁴	(transition state cumulative Units Clinically infectious) Running total number of units that are initially clinical and those that have become clinical.
tscAClin ⁴	(transition state cumulative Animals Clinically infectious) Running total number of animals in units as described immediately above.
tscUNImm ⁴	(transition state cumulative Units Naturally Immune) Running total number of units that are initially naturally immune and those that have become naturally immune.
tscANImm ⁴	(transition state cumulative Animals Naturally Immune) Running total number of animals in units as described immediately above.
tscUVIImm ⁴	(transition state cumulative Units Vaccine Immune) Running total number of units that are initially vaccine immune and those that have become vaccine immune.
tscAVIImm ⁴	(transition state cumulative Animals Vaccine Immune) Running total number of animals in units as described immediately above.
tscUDest ⁵	(transition state cumulative Units Destroyed) Running total number of units that are initially in the “destroyed” state and those that have been destroyed.
tscADest ⁵	(transition state cumulative Animals Destroyed) Running total number of animals in units as described immediately above.
infnUAir	(infections new Units Airborne spread) Number of units that became infected by airborne spread on the given day.
infnAAir	(infections new Animals Airborne spread) Total number of animals in units newly infected by airborne spread on the given day.

Table B-3. Daily epidemiological outputs. (continued)

Output name	Description
infnUDir	(infections new Units Direct contact) Number of units that became infected by direct contact on the given day.
infnADir	(infections new Animals Direct contact) Total number of animals in units newly infected by direct contact on the given day.
infnUInd	(infections new Units Indirect contact) Number of units that became infected by indirect contact on the given day.
infnAInd	(infections new Animals Indirect contact) Total number of animals in units newly infected by indirect contact on the given day.
infcUIni ⁶	(infections cumulative Units Initially infected) Number of units that are initially infected at the beginning of an iteration.
infcAIni ⁶	(infections cumulative Animals Initially infected) Number of animals in initially infected units at the beginning of an iteration.
infcUAir	(infections cumulative Units Airborne spread) Running total of the number of units that become infected by airborne spread over the course of an iteration.
infcAAir	(infections cumulative Animals Airborne spread) Running total of the number of animals in units that become infected by airborne spread over the course of an iteration.
infcUDir	(infections cumulative Units Direct contact) Running total of the number of units that become infected by direct contact over the course of an iteration.
infcADir	(infections cumulative Animals Direct contact) Running total of the number of animals in units that become infected by direct contact over the course of an iteration
infcUInd	(infections cumulative Units Indirect contact) Running total of the number of units that become infected by indirect contact over the course of an iteration.
infcAInd	(infections cumulative Animals Indirect contact) Running total of the number of animals in units that become infected by indirect contact over the course of an iteration.

Appendix B. NAADSM outputs

Table B-3. Daily epidemiological outputs. (*continued*)

Output name	Description
expcUDir	(exposures cumulative Units Direct contact) Running total number of units directly exposed to any infected unit over the course of an iteration.
expcADir	(exposures cumulative Animals Direct contact) Running total number of animals in directly exposed units over the course of an iteration.
expcUInd	(exposures cumulative Units Indirect contact) Running total number of units indirectly exposed to any infected unit over the course of an iteration.
expcAInd	(exposures cumulative Units Indirect contact) Running total number of animals in indirectly exposed units over the course of an iteration.
trcUDir	(trace cumulative Units Direct contact) Running total number of units directly exposed and successfully traced over the course of an iteration.
trcADir	(trace cumulative Animals Direct contact) Running total number of animals in directly exposed and successfully traced units over the course of an iteration.
trcUInd	(trace cumulative Units Indirect contact) Running total number of units indirectly exposed and successfully traced over the course of an iteration.
trcAInd	(trace cumulative Animals Indirect contact) Running total number of animals in indirectly exposed and successfully traced units over the course of an iteration.
trcUDirp	(trace cumulative Units Direct contact possible) Running total number of directly exposed units that possibly could have been traced over the course of an iteration.
trcADirp	(trace cumulative Animals Direct contact possible) Running total number of animals in all directly exposed units that possibly could have been traced over the course of an iteration.
trcUIndp	(trace cumulative Units Indirect contact possible) Running total number of indirectly exposed units that possibly could have been traced over the course of an iteration.
trcAIndp	(trace cumulative Animals Indirect contact possible) Running total number of animals in all indirectly exposed units that possibly could have been traced over the course of an iteration.

Table B-3. Daily epidemiological outputs. (continued)

Output name	Description
detnUClin	(detection new Units Clinical signs) Number of units newly detected by clinical signs on the indicated day.
detnAClin	(detection new Animals Clinical signs) Total number of animals in newly detected clinical units.
<u>desnUAll</u>	(destructions new Units All) Number of units newly destroyed for any reason on the indicated day.
<u>desnAAll</u>	(destructions new Animals All) Total number of animals in newly destroyed units.
<u>vaccnUAll</u>	(vaccination new Units All) Number of units newly vaccinated for any reason on the indicated day.
<u>vaccnAAll</u>	(vaccination new Animals All) Total number of animals in newly vaccinated units.
detcUClin	(detection cumulative Units Clinical signs) Running total number of clinical units detected over the course of an iteration.
detcAClin	(detection cumulative Animals Clinical signs) Running total number of animals in detected clinical units over the course of an iteration.
descUIni ⁶	(destructions cumulative Units Initially destroyed) Total number of units in the “destroyed” state at the outset of an iteration (<i>i.e.</i> , units destroyed prior to the point in time when the simulation began).
descAIni ⁶	(destructions cumulative Animals Initially destroyed) Total number of animals in units in the “destroyed” state at the outset of an iteration.
descUDet	(destructions cumulative Units Detected by clinical signs) Running total number of infected units destroyed because they were detected based on clinical signs over the course of an iteration.
descADet	(destructions cumulative Animals Detected by clinical signs) Running total number of animals in units destroyed for the reason listed immediately above.
descUDir	(destructions cumulative Units Direct contact) Running total number of units destroyed because of direct contact with an infected unit over the course of an iteration.
descADir	(destructions cumulative Animals Direct contact) Running total number of animals in units destroyed for the reason listed immediately above.

Appendix B. NAADSM outputs

Table B-3. Daily epidemiological outputs. (*continued*)

Output name	Description
descUInd	(destructions cumulative Units Indirect contact) Running total number of units destroyed because of indirect contact with an infected unit over the course of an iteration.
descAInd	(destructions cumulative Animals Indirect contact) Running total number of animals in units destroyed for the reason listed immediately above.
descURing	(destructions cumulative Units Ring destruction) Running total number of units destroyed because they were within a destruction ring over the course of an iteration.
descARing	(destructions cumulative Animals Ring destruction) Running total number of animals in units destroyed for the reason listed immediately above.
vaccUIni ⁶	(vaccination cumulative Units Initially vaccinated) Total number of units in the “vaccine immune” state at the outset of an iteration (i.e., units vaccinated prior to the point in time when the simulation began).
vaccAIni ⁶	(vaccination cumulative Animals Initially vaccinated) Total number of animals in vaccine immune units at the outset of an iteration.
vaccURing	(vaccination cumulative Units Ring) Running total number of units vaccinated because they were within a vaccination ring over the course of an iteration.
vaccARing	(vaccination cumulative Animals Ring) Running total number of animals in units vaccinated for the reason listed immediately above.

³ This is NOT the number of units/animals that *enter* the specified disease state on the given day, but rather the total number of units/animals in the population that *have* the specified state on the given day. The sum of units/animals in all states on a given day is equal to the total number of units/animals in the population.

⁴ Since it is possible for a unit to repeat the disease cycle within a single iteration, these values may exceed the number of units/animals in the population.

⁵ It is not possible for a unit to be destroyed more than once in a single iteration. Consequently, these values can never exceed the number of units/animals in the population.

⁶ Units/animals that are initially infected(i.e., in an infected state at the beginning of an iteration), initially destroyed, or initially vaccinated appear as initially infected, destroyed, or vaccinated (respectively) on day 1. The cumulative number of units/animals initially infected, destroyed, or vaccinated cannot change during an iteration.

B4. Epidemic curves

Information regarding the number of infected herds on each day of each iteration, as well as the number of newly detected cases on each day of each iteration, is saved for the purpose of generating epidemic curves. (For example, the summary epidemic curves discussed in Section 10.4.2 are generated from this data.) The daily number of infections may be used to generate an actual epidemic curve; the daily number of detections may be used to generate an apparent epidemic curve. The total number of units as well as the total number of animals in those units is recorded. This production type-specific data is stored in the database table **outEpidemicCurves**, as shown in Appendix E, Table E-16.

Appendix C. Plain text comma-delimited (*.csv) files used in NAADSM

C1. Files containing population (herd or unit) data

C1.1. Importing population data

Plain text comma-delimited (*.csv) population files intended for import by *NAADSM* must have a header row that gives field names. Files must contain the fields with field names as shown in Table C-1 (field names are not case-sensitive and do not contain spaces).

Table C-1. Required fields for *NAADSM* *.csv files.

Field name	Field description
<i>HerdID</i> or <i>ID</i> ¹	Unique integer identifier for each unit. ID must be greater than 0.
<i>ProductionType</i>	Identifier for the unit's production type. This value may be either the numeric ID of the production type or the name of the production type.
<i>HerdSize</i>	Integer indicating the number of animals in the unit.
<i>Lat</i>	Real (floating point) number indicating the latitude of the unit. Values must be between -90 and 90, inclusive.
<i>Lon</i>	Real (floating point) number indicating the longitude of the unit. Values must be between -180 and 180, inclusive.
<i>Status</i>	Code indicating the unit's disease transition state at the beginning of the simulation. Values may be numeric or single character codes as shown in Table C-2.

¹ *HerdID* is the preferred name for this field as of version 3.0.63, but CSV files containing the field name *ID* will be imported. This change was made to work around an odd behavior in Microsoft Excel: please see <http://support.microsoft.com/default.aspx?scid=kb;en-us;323626&Product=xlw> for more information.

Appendix C. *.csv file formats

Table C-2. Codes used for disease transition states

Transition state	Single character code	Numeric code
Susceptible	S	0
Latent	L	1
Subclinical	B	2
Clinical	C	3
Naturally immune	N	4
Vaccine immune	V	5
Destroyed	D	6

Comma-delimited population files may contain an optional field, *DaysLeftInStatus*, which contains an integer value for the number of days the unit has remaining in its current status². A value of -1 may be used to indicate that the number of days remaining is unspecified (if the number of days remaining is unspecified, the model will choose a number based on the appropriate probability density functions). *DaysLeftInStatus* applies to units in all disease states except Susceptible: if *DaysLeftInStatus* is specified for a susceptible unit, the value is ignored.

Comma-delimited population files intended for import by NAADSM may contain other fields. Fields with names that do not correspond to one of those listed above will be ignored.

C1.2. Population files exported by NAADSM

Comma-delimited population files generated by NAADSM have the seven fields described above: *HerdID*, *ProductionType*, *HerdSize*, *Lat*, *Lon*, *Status*, and *DaysLeftInStatus*. The user may specify whether production types are written as production type ID or production type name. The user may also specify whether disease transition state is written as a numeric code or single character code.

² For NAADSM 3.0.62, if any record contains the field *DaysLeftInStatus*, then all records must contain the field *DaysLeftInStatus*, or file parsing will fail. This limitation was corrected in NAADSM 3.0.63.

C1.3. Sample population files

Figures C-1 through C-3 show sample *.csv files generated by or suitable for import into NAADSM.

```
HerdID,ProductionType,HerdSize,Lat,Lon,Status,DaysLeftInStatus
1,"Swine",19000,42.9,-94.899999,L,10
2,"Swine",15000,42.8,-94.581999,L,-1
3,"Swine",20000,42.7,-94.187899,S,-1
4,"Swine",3000,42.6,-93.663299,S,-1
5,"Swine",63000,42.5,-92.866699,S,-1
6,"Swine",52000,42.4,-91.177999,S,-1
7,"Swine",45000,42.3,-91.299999,S,-1
8,"Swine",4000,43.2,-95.499999,S,-1
9,"Swine",28000,41.1,-90.699999,S,-1
```

Figure C-1. Default *.csv file format generated by NAADSM 3.0.63 (and higher).

```
HerdID,ProductionType,HerdSize,Lat,Lon,Status,DaysLeftInStatus
1,4,19000,42.9,-94.899999,L,10
2,4,15000,42.8,-94.581999,L,-1
3,4,20000,42.7,-94.187899,S,-1
4,4,3000,42.6,-93.663299,S,-1
5,4,63000,42.5,-92.866699,S,-1
```

Figure C-2. A *.csv file generated by NAADSM 3.0.63 (and higher) using production type ID number.³

```
HerdID,ProductionType,HerdSize,Lat,Lon,Status,DaysLeftInStatus
1,"Swine",19000,-42.9,-94.899999,1,10
2,"Swine",15000,-42.8,-94.581999,1,-1
3,"Swine",20000,-42.7,-94.187899,0,-1
4,"Swine",3000,-42.6,-93.663299,0,-1
5,"Swine",63000,-42.5,-92.866699,0,-1
6,"Swine",52000,-42.4,-91.177999,0,-1
7,"Swine",45000,-42.3,-91.299999,0,-1
8,"Swine",4000,-43.2,-95.499999,0,-1
9,"Swine",28000,-41.1,-90.699999,0,-1
```

Figure C-3. A *.csv file generated by NAADSM 3.0.63 (and higher) using numeric transition state code.³

Appendix C. *.csv file formats

Figure C-4 shows a *.csv file generated by the USDA disease modeling application *SpreadModel*³, a predecessor to *NAADSM*. Files with this format may be imported by *NAADSM*. Most fields will be ignored. *NAADSM* does not export files in this format.

```
ID,ProductionType,HerdSize,Lat,Lon,Status,DaysInStatus,DaysLeftInStatus,  
Prevalence,Mortality,HowInfected,DetectionReason,DaysSinceDetected,  
VaccinationReason,DaysSinceVaccinated,DestructionReason,HoldingReason,  
HoldingDays,SurvRingHits,SurvVisits,SurvTests  
1,1,80,54.8819046,-3.1844039,3,0,18,1,0,3,0,0,0,0,0,0,0,0,0,0,0,0  
2,1,70,54.766201,-2.99002695,0,50,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0  
3,1,8,54.8438072,-3.0588901,1,5,4,1,0,3,0,0,0,0,0,0,0,0,0,0,0,0  
4,1,22,54.7988701,-3.05926991,3,7,20,1,0,3,0,0,0,0,0,0,0,0,0,0,0
```

Figure C-4. A *.csv file generated by *SpreadModel*. Note that the unit with ID 2 has **Status** (disease transition state) of **Susceptible** and **DaysLeftInStatus** of 0. When imported, **DaysLeftInStatus** will be unspecified. Also note that, due to page width constraints, the single header row in this particular example is displayed across four rows.

C2. Files containing piecewise probability density functions

Point arrays for piecewise *pdfs* may be imported from *.csv files (see Section 6.1.5.1). The file must have a header row to indicate the *x* and *y* columns. Values along the *x* axis should be in units appropriate for the *pdf* (for example, the *x* axis should be in days for a *pdf* describing the length of the latent period). Values along the *y* axis are in units of probability density such that the area under the curve of the entire function equals 1. *NAADSM* will adjust *y* axis values as needed to ensure that the area under the curve equals 1. Figure C-5 shows sample data suitable for import into *NAADSM*. Point arrays for piecewise *pdfs* are exported (Section 6.1.5.2) in the same format.

X	Y
0	0
0.5	0.012198
1.19	0.052892
2.16	0.178874
2.72	0.349649
3	0.406341
3.56	0.353648
4.28	0.186973
5.48	0.052892
6.59	0.012198
10	0

Figure C-5. A *.csv file for a piecewise *pdf*. The first row is a header row, which identifies the *x* and *y* columns. (Columns shown here are evenly spaced to make them easier to read; uniform spacing is not required in the file to be imported.)

³ Schoenbaum, M.A., and Disney, W.T. 2003. Modeling alternative mitigation strategies for a hypothetical outbreak of foot-and-mouth disease in the United States. *Prev. Vet. Med.* 58: 25-52.

C3. Files containing relational functions

Point arrays for relational functions may be imported from *.csv files (see Section 6.1.6.4). The file must have a header row to indicate the *x* and *y* columns. Values along the *x* and *y* axes should be in units appropriate for the *rel*. Figure C-6 shows sample data suitable for import into NAADSM. Point arrays for *rels* are exported in the same format.

x,	y
0,	10
20,	20
50,	40
80,	75
100,	95
120,	100

Figure C-6. A properly formatted *.csv file for importing a relational function. Note that the first row is a header row, which identifies the *x* and *y* columns. (Columns shown here are evenly spaced to make them easier to read: uniform spacing is not required in the file to be imported.)

Appendix D. Probability density functions supported by *NAADSM*

As described in Section 3 and Section 6.1.5, many parameters in a *NAADSM* scenario are probability density functions (*pdfs*). Probability density functions are distributions of values representative of the natural range of possible values for some parameter. Values are drawn stochastically from these distributions as a simulation runs.

NAADSM supports 15 general *pdf* types, described in the following sections. References are provided for users who wish to obtain more detailed information about these distributions: in particular, Vose (2000) provides a very helpful discussion regarding suitable applications of the different types of *pdfs*.

All but one of the distributions described here are continuous distributions. Recall, though, that *NAADSM* operates in discrete time steps of one day. Consequently, for parameters that have *x* axis units of days (*e.g.*, the disease transition state periods described in Section 8.5.1), values obtained from these distributions will be rounded to the nearest whole day.

D1. The Beta distribution

A Beta distribution (Figure D-1) is defined by four parameters: α_1 , α_2 , a minimum value, and a maximum value. Parameters α_1 and α_2 must be greater than 0, and the minimum value must be less than the maximum value. The probability density $f(x)$ is calculated as:

$$f(x) = \frac{(x - Min)^{\alpha_1-1} (Max - x)^{\alpha_2-1}}{B(\alpha_1, \alpha_2)(Max - Min)^{\alpha_1+\alpha_2-1}}$$

where B denotes the Beta function.

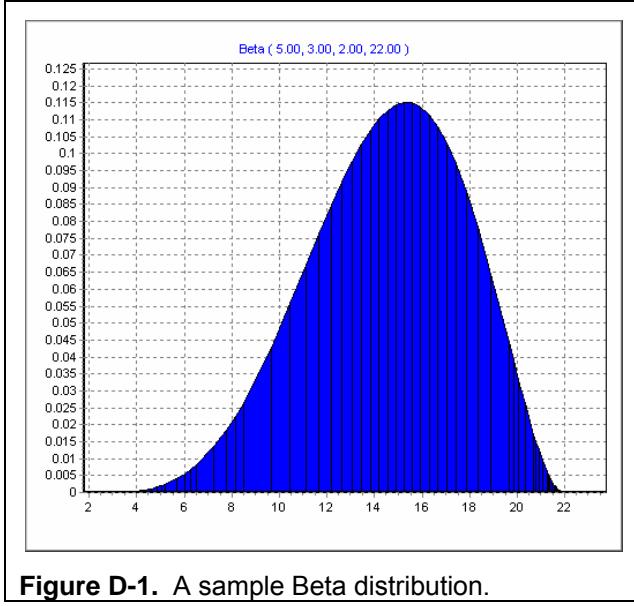


Figure D-1. A sample Beta distribution.

D2. The BetaPERT distribution

The BetaPERT distribution (Figure D-2) is defined by its minimum, its most likely value (mode), and its maximum. In this way, the BetaPERT distribution is similar to the Triangular distribution.

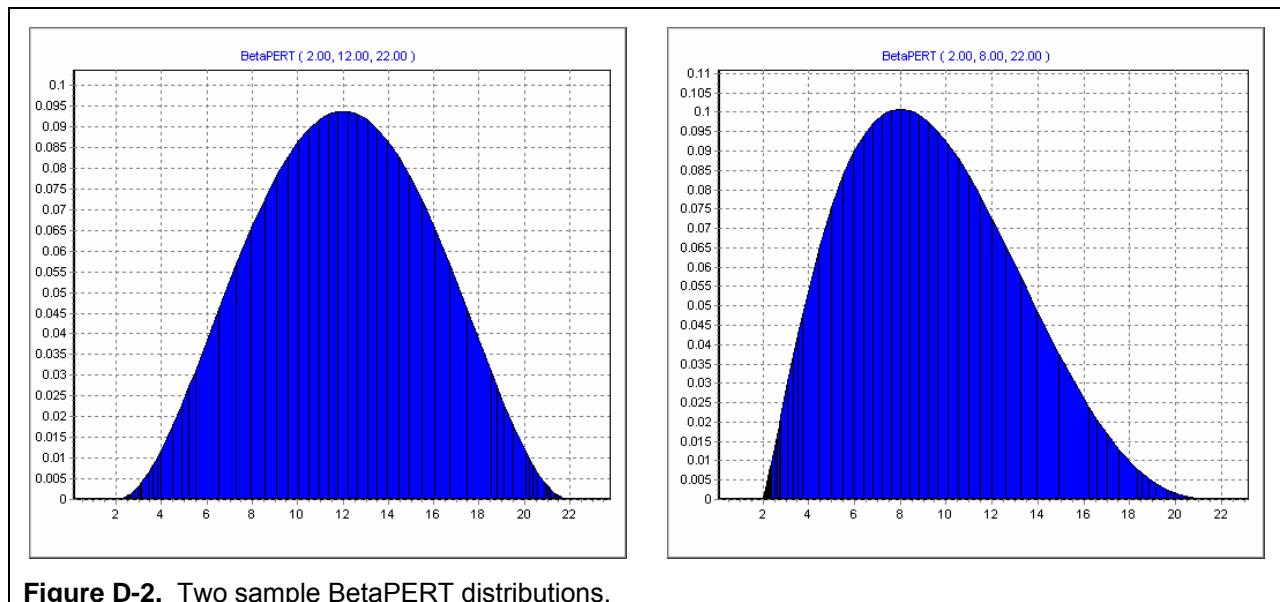
The BetaPERT distribution is related to the Beta distribution: the α_1 and α_2 parameters used to define a Beta distribution are obtained from the minimum, mode, and maximum values of a BetaPERT. There are several slight variations regarding how α_1 and α_2 are calculated from the minimum, mode, and maximum: *NAADSM* uses the same approach as the risk analysis package *@Risk* (Palisade Corporation, 2002). In this approach, α_1 and α_2 are calculated with the following formulas:

$$\alpha_1 = 6 \left(\frac{\mu - Min}{Max - Min} \right)$$

$$\alpha_2 = 6 \left(\frac{Max - \mu}{Max - Min} \right)$$

where μ is the distribution mean, defined as:

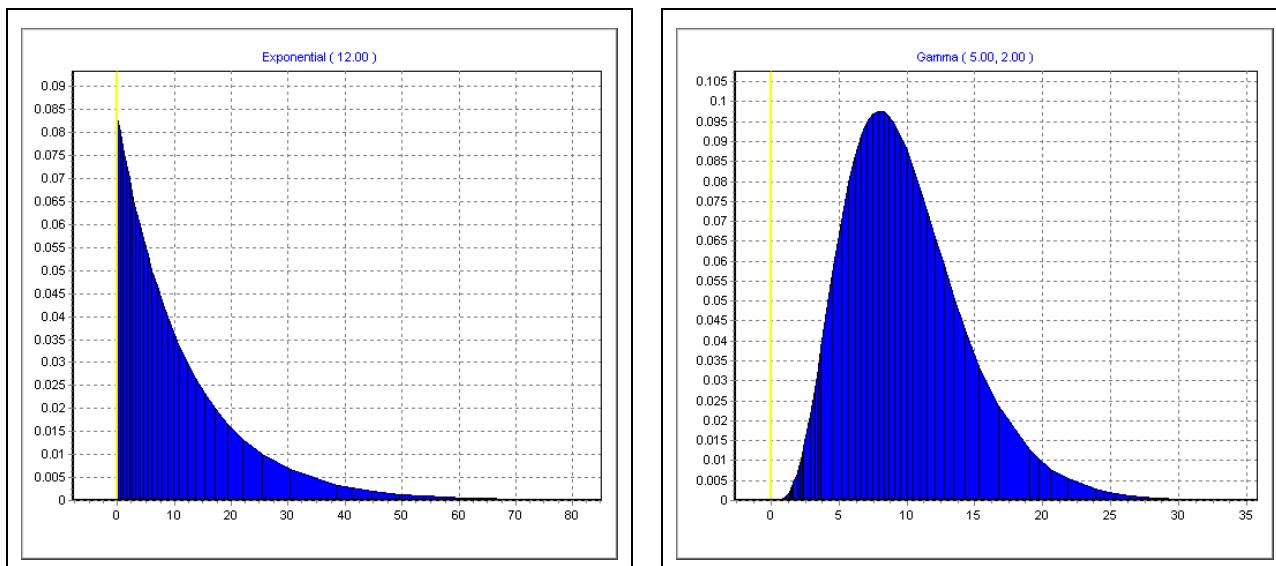
$$\mu = \frac{Min + 4Mode + Max}{6}$$

**Figure D-2.** Two sample BetaPERT distributions.

D3. The Exponential distribution

An Exponential distribution (Figure D-3) is a highly skewed distribution defined by its mean μ , which must be greater than 0. The probability density function $f(x)$ is calculated as:

$$f(x) = \frac{e^{-x/\mu}}{\mu}$$

**Figure D-3.** A sample Exponential distribution.**Figure D-4.** A sample Gamma distribution.

D4. The Gamma distribution

A Gamma distribution (Figure D-4) is defined by two parameters: its shape (α), and its scale (β), where α and β must be greater than 0. This distribution has a mean of $\alpha \times \beta$. The probability density function $f(x)$ is calculated as:

$$f(x) = \frac{1}{\beta \Gamma(\alpha)} \left(\frac{x}{\beta} \right)^{\alpha-1} e^{-x/\beta}$$

where Γ is the Gamma function.

D5. The Gaussian (Normal) distribution

A Gaussian or Normal distribution (Figure D-5) is a bell-shaped curve described by two parameters: its mean μ , and its standard deviation σ . The standard deviation must be greater than 0. The Gaussian distribution is inherently symmetric. The probability density function $f(x)$ is calculated as:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

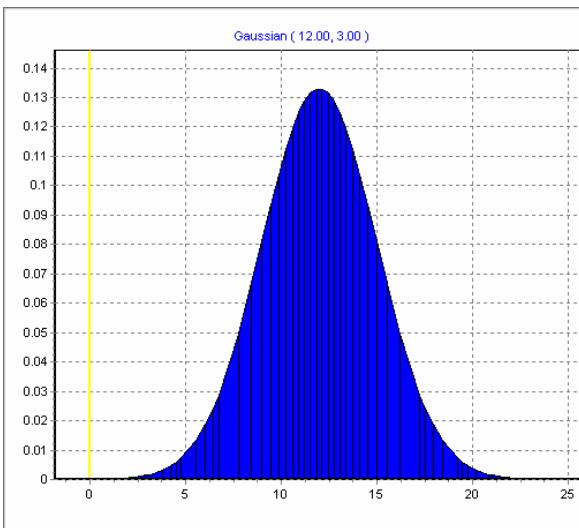


Figure D-5. A sample Exponential distribution.

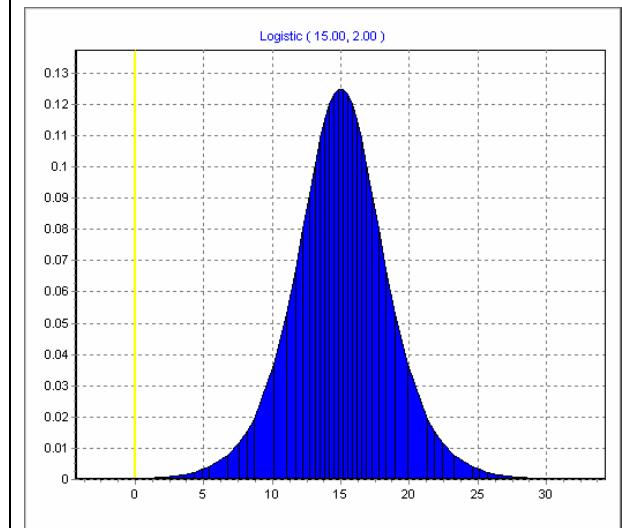


Figure D-6. A sample Logistic distribution.

D6. The Logistic distribution

A Logistic function (Figure D-6) is defined by two parameters: its location α and scale β . The scale parameter must be greater than 0. The probability density function $f(x)$ is calculated as:

$$f(x) = \frac{\text{sech}^2\left[\frac{1}{2}\left(\frac{x-\alpha}{\beta}\right)\right]}{4\beta}$$

where sech is the hyperbolic secant function

D7. The Loglogistic distribution

The Loglogistic function (Figure D-7) is defined by three parameters: its shape α , scale β , and location γ . Scale and shape must be greater than 0. The probability density function $f(x)$ is calculated as:

$$f(x) = \frac{\alpha\left(\frac{x-\gamma}{\beta}\right)^{\alpha-1}}{\beta\left[1+\left(\frac{x-\gamma}{\beta}\right)^\alpha\right]^2}$$

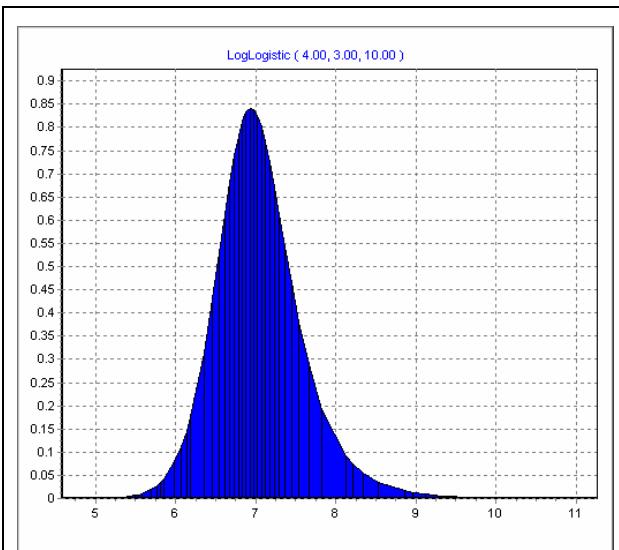


Figure D-7. A sample Loglogistic distribution.

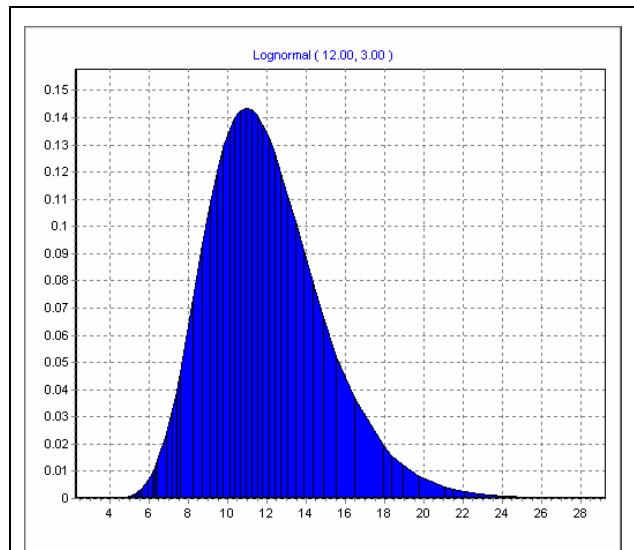


Figure D-8. A sample Lognormal distribution.

D8. The Lognormal distribution

The Lognormal distribution (Figure D-8) is a logarithmic transformation of the normal distribution; it is described using the same parameters as the normal distribution: its mean μ and its standard deviation σ , where both μ and σ are greater than 0.. The Lognormal distribution is extremely asymmetric (skewed to the right) when the mean is close to 0; the further the mean is from 0, the more the Lognormal distribution approaches the symmetry and shape of a Normal distribution.

The probability density function $f(x)$ for a Lognormal distribution is calculated as:

$$f(x) = \frac{1}{x\sqrt{2\pi\sigma'}} e^{-\frac{1}{2}\left(\frac{\ln(x)-\zeta}{\sigma'}\right)^2} \quad \text{where } \zeta = \ln\left(\frac{\mu^2}{\sqrt{\sigma'^2 + \mu^2}}\right) \quad \text{and } \sigma' = \sqrt{\ln\left[1 + \left(\frac{\sigma}{\mu}\right)^2\right]}$$

A Lognormal distribution may also be defined by zeta(ζ) and σ' , as shown above: *NAADSM* supports both sets of parameters for Lognormal distributions, and automatically handles the conversion of μ and σ to ζ and σ' .

D9. The Pearson 5 distribution

The Pearson 5 distribution (Figure D-9) is defined by its shape α and scale β , both of which must be greater than 0. A Pearson 5 distribution has a mean of $\beta/(\alpha-1)$. The probability density function $f(x)$ is calculated as:

$$f(x) = \frac{1}{\beta\Gamma(\alpha)} \cdot \frac{e^{-\beta/x}}{(x/\beta)^{\alpha+1}} \quad \text{where } \Gamma \text{ is the Gamma function.}$$

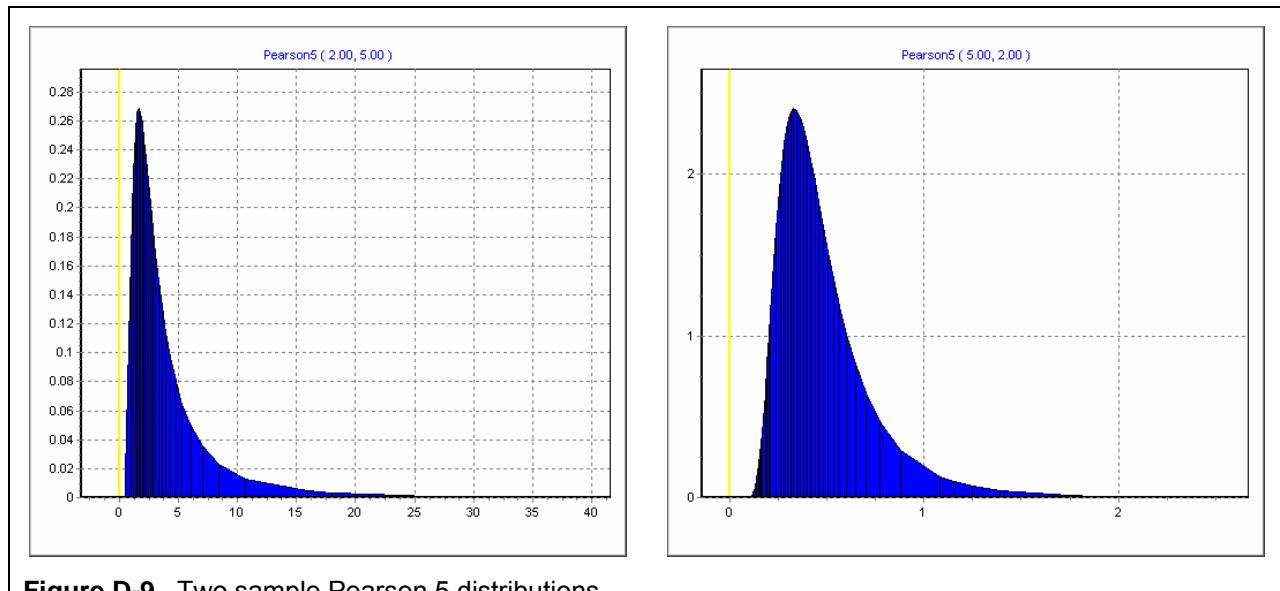


Figure D-9. Two sample Pearson 5 distributions.

D10. The Piecewise (General) distribution

A Piecewise or General distribution (Figure D-10) is defined by an array of points, each of which have an x and a y value. Each x value must be larger than the previous x value. Each y value must be at least 0. Finally, the y values of the first and last points must be 0. The probability density function $f(x)$ is calculated as.

$$f(x) = y_i + \left[\frac{x - x_i}{x_{i+1} - x_i} \right] (y_{i+1} - y_i) \text{ where } x_i \leq x \leq x_{i+1}$$

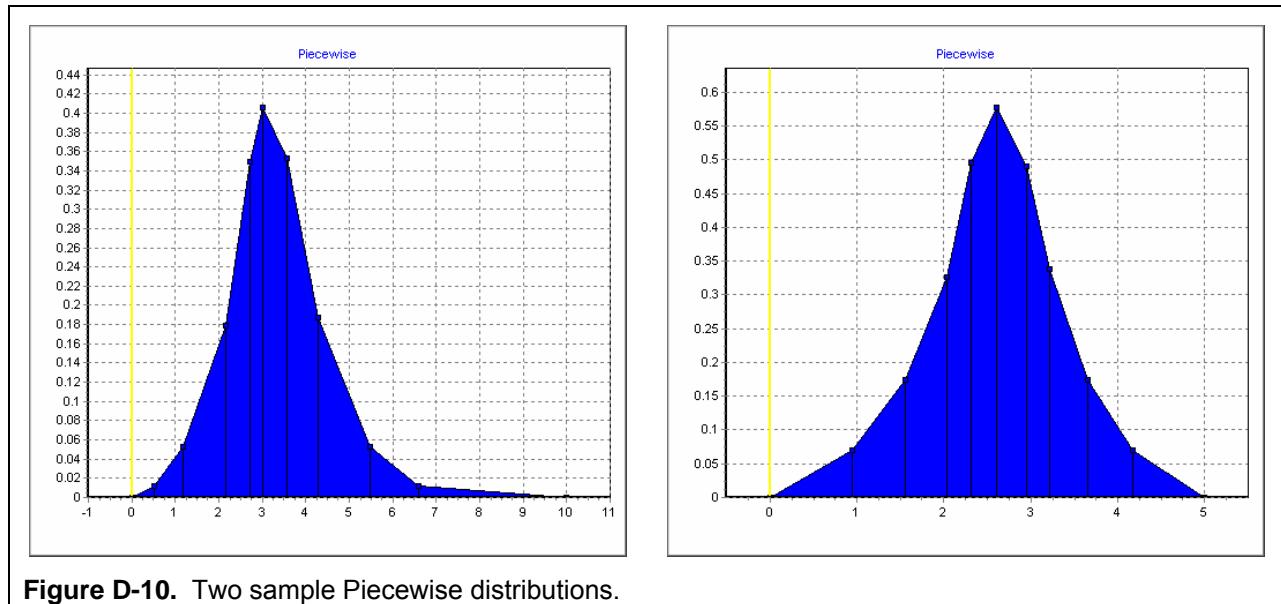


Figure D-10. Two sample Piecewise distributions.

D11. The Point “distribution”

Point values are not distributions, but are used in much the same way by NAADSM. A Point “distribution” is defined by a fixed value y . The probability density function $f(x)$ for a Point “distribution” always returns this fixed value:

$$f(x) = y$$

D12. The Poisson distribution

Unlike any of the other distributions described in this appendix, the Poisson distribution (Figure D-11) is discrete rather than continuous. Poisson distributions have one specific role in an NAADSM scenario: Poisson distributions are used to determine the number of contacts that will be initiated by each herd that is a source of disease: see Section 8.6.3.2.

Appendix D. Probability density functions

A Poisson distribution is defined by its mean, designated λ . The probability mass function $f(x)$ is calculated as:

$$f(x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

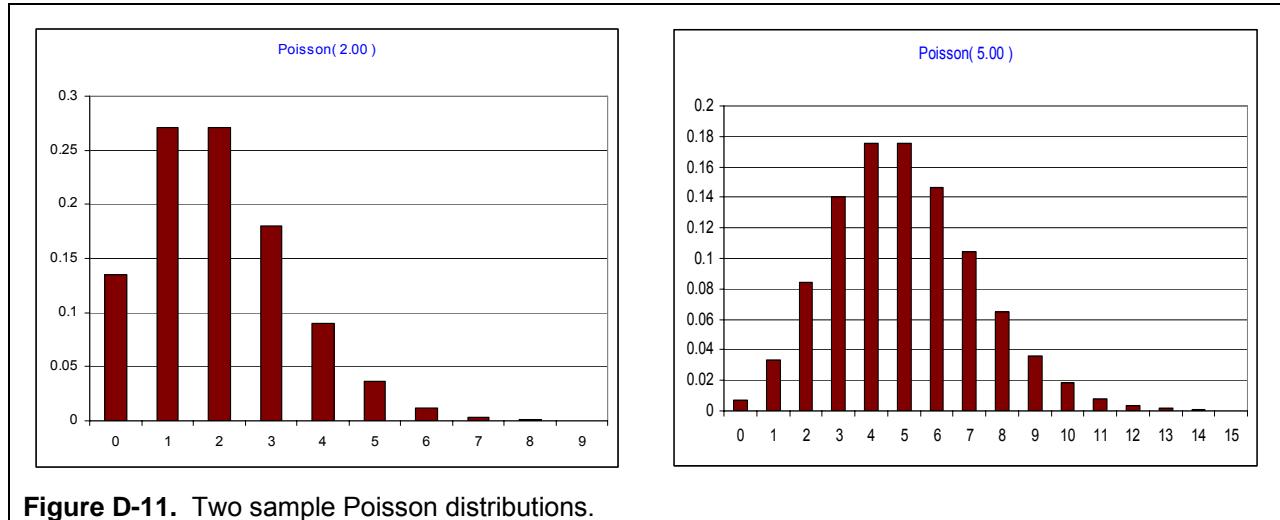


Figure D-11. Two sample Poisson distributions.

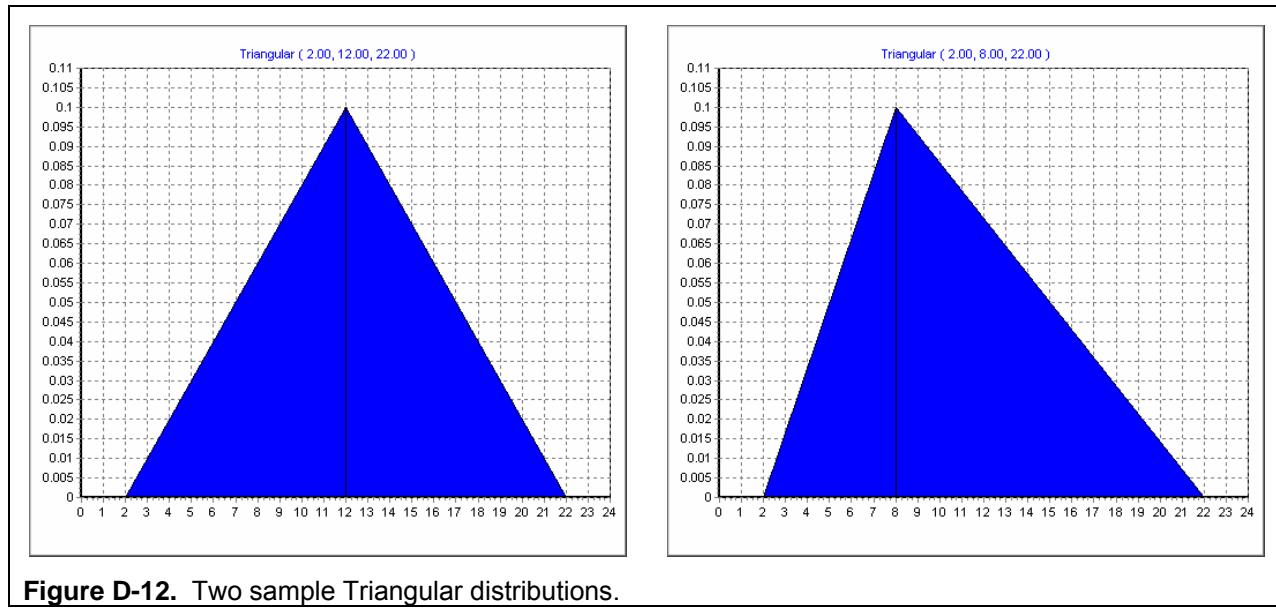
D13. The Triangular distribution

A Triangular distribution (Figure D-12) is, as its name implies, a triangle. It is described by three parameters: the minimum, peak (mode or “most likely”), and maximum values. The Triangular distribution can be symmetric or asymmetric depending on the relation of the peak to the minimum and maximum values. The probability density function $f(x)$ is calculated as:

$$f(x) = \frac{2(x - Min)}{(Mode - Min)(Max - Min)} \text{ if } Min \leq x \leq Mode$$

or

$$f(x) = \frac{2(Max - x)}{(Max - Mode)(Max - Min)} \text{ if } Mode \leq x \leq Max$$



D14. The Uniform distribution

The Uniform distribution (Figure D-13) is a rectangular block, indicating that all values within a range occur with equal frequency; it is described by two parameters: the minimum and maximum of the range. The Uniform distribution is inherently symmetric. The probability density function $f(x)$ is calculated as:

$$f(x) = 1/(Max-Min)$$

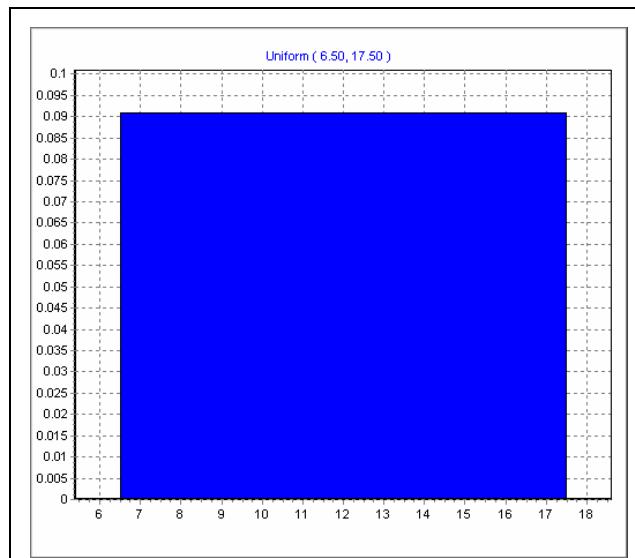


Figure D-13. A sample Uniform distribution.

D15. The Weibull distribution

A Weibull distribution (Figure D-14) is defined by its shape α and scale β , both of which must be greater than 0. The probability density function $f(x)$ is calculated as:

$$f(x) = \frac{\alpha x^{\alpha-1}}{\beta^\alpha} e^{-(x/\beta)^\alpha}$$

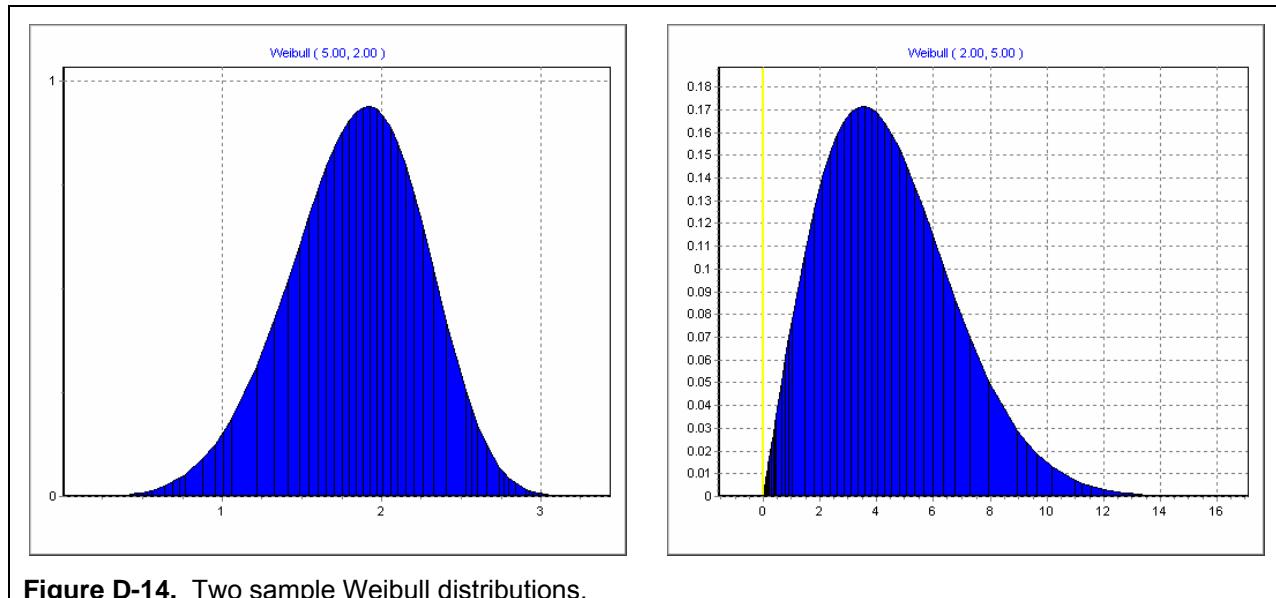


Figure D-14. Two sample Weibull distributions.

D16. References

- Galassi, Mark, Jim Davies, James Theiler, Brian Gough, Gerard Jungman, Micheal Booth, and Fabrice Rossi. 2004. *GNU Scientific Library Reference Manual*. The GNU Scientific Library (GSL) is a collection of routines for numerical computing. *NAADSM* uses the GSL for many of its probability density functions. GSL source code is distributed under the GNU General Public License. The *Reference Manual* is available online at <<http://www.gnu.org/software/gsl>>. An Adobe PDF file is available for download from <<http://gnuwin32.sourceforge.net/packages/gsl.htm>> (follow the documentation download link).
- McLaughlin, Michael P. 2000. *A Compendium of Common Probability Density Functions*. A reference document assembled by the creators of the mathematical modeling program *Regress+*. The *Compendium* is available for download as an Adobe PDF file from <http://www.causascientia.org/math_stat/Dists/Compendium.html>.
- Palisade Corporation. 2002. *A Concise Summary of @RISK Probability Distribution Functions*. A guide published by the creators of *@Risk*, a risk analysis package for *Microsoft Excel*. The Summary is available for download as an Adobe PDF file from <<http://project.zf.jcu.cz/risk/data/distfunc.pdf>>.
- Vose, David. 2000. *Risk Analysis: A Quantitative Guide*, 2nd edition. New York: John Wiley & Sons. Vose's book includes a very useful chapter on probability density functions and their applications.

Appendix E. Scenario file database schema

E1. Table types in *NAADSM* scenario files

The PC version of the *North American Animal Disease Spread Model (NAADSM)* stores scenario input and simulation output in a *Microsoft Access*-compatible (*.mdb) database file. This appendix describes the tables and fields that appear in *NAADSM* database files for advanced users, database developers, and programmers.

Tables in the database fall into four categories:

- Tables that contain scenario input parameters (indicated with the prefix *in*). Input tables are not changed as a simulation runs.
- Tables that contain scenario input and also have fields that are modified as the simulation runs (dynamic tables, indicated with the prefix *dyn*).
- Tables that contain simulation output (indicated with the prefix *out*). Output tables are updated as a simulation runs.
- Read-only lookup tables that serve as references for codes used elsewhere in the database file (indicated with the prefix *read*).

Figure E-1 shows all of the tables and their relationships. The following sections describe the fields of the individual tables.

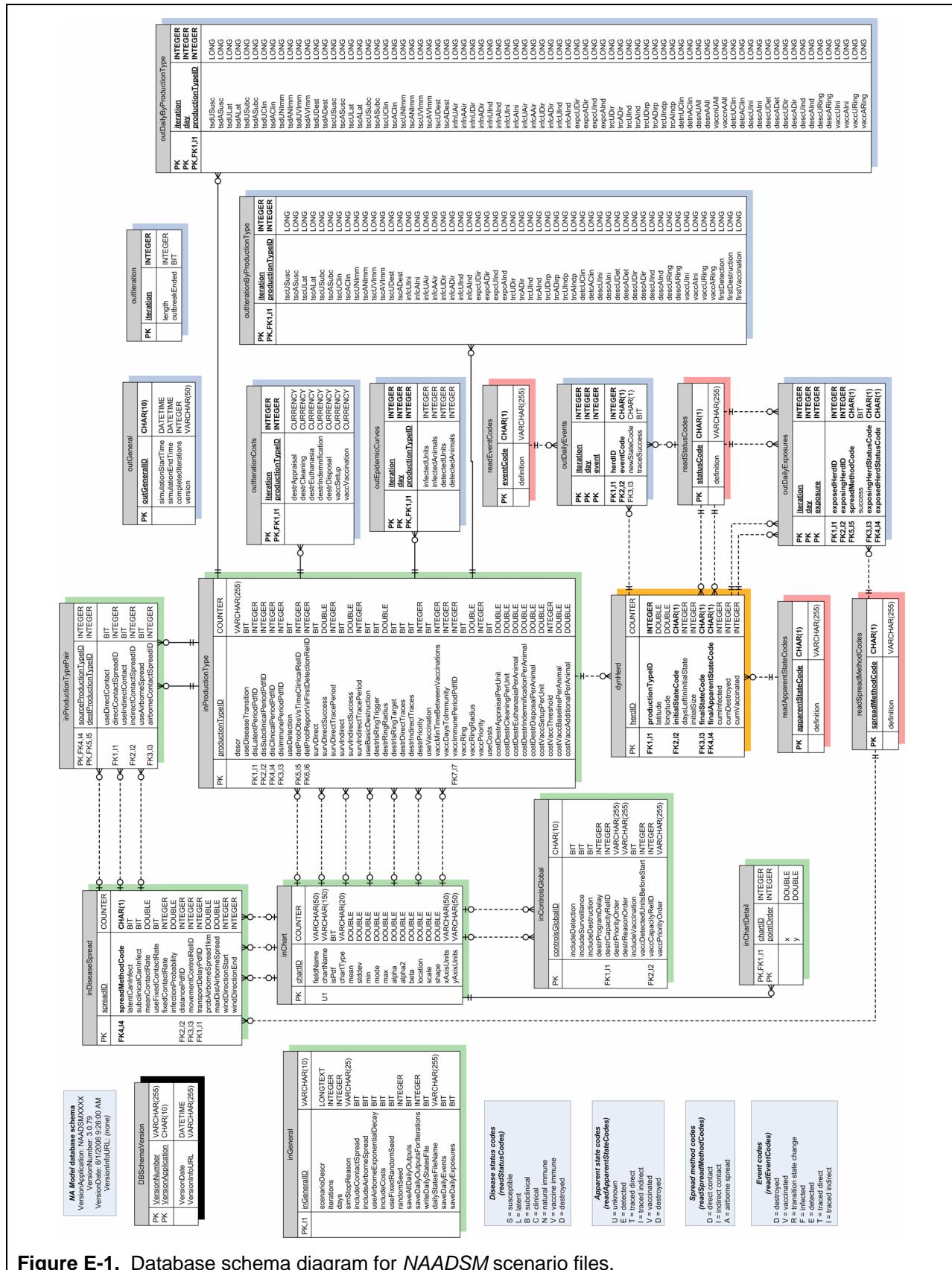
E2. The **DBSchemaVersion** database table

In addition to the tables that fit into the four categories listed above, there is one additional special table: table **DBSchemaVersion** contains information about the version of the database schema that is implemented by the database file. This table is used internally by *NAADSM* and related applications, but has no role in the definition of a scenario or the collection of simulation output. **DBSchemaVersion** contains only a single record.

Table E-1. Fields of database table **DBSchemaVersion**

Field name	Data type	Description
VersionNumber	Text(255)	The minimum version of <i>NAADSM</i> capable of using this database.
VersionApplication	Char(10)	This value is always "NAADSMXXXX".
VersionDate	Datetime	The date and time that the version indicated above was released.
VersionInfoURL	Text(255)	An Internet address where more information about this version may be found.

Appendix E. Scenario file database schema



E3. Database tables for scenario input

E3.1. Database table inGeneral

The database table **inGeneral** contains information that pertains to the entire simulation (*e.g.*, the number of iterations to run, the number of days for each iteration, and whether to include contact spread in the scenario). Most fields in this table (Table E-2) correspond to settings made in the **Start setup** window (Section 8.2) and the **Output options** window (Section 8.12). This table contains only one record.

Table E-2. Fields of the database table **inGeneral**.

Field name	Data type	Description
inGeneralID	Text(10)	A primary key for the single record stored in this table. This value is always "NAADSMXXXX".
scenarioDescr	Longtext	The description of the scenario.
iterations	Long	The number of iterations of this scenario that should be run.
days	Long	The number of days that iterations of this scenario should run, if the command Run → Start and run until specified day... is used.
simStopReason	Text(25)	The criterion used to end each iteration. This may be that the specified number of days has passed, the first detection has occurred, or the outbreak has ended.
includeContactSpread	Bit	Indicates whether disease spread by direct or indirect contact is used in the scenario.
includeAirborneSpread	Bit	Indicates whether airborne spread is used in the model.
useAirborneExponentialDecay	Bit	Indicates whether the decrease in probability of spread by airborne transmission is simulated by the exponential (TRUE) or linear (FALSE) algorithm: see Section 8.6.4.
includeCosts	Bit	Indicates whether direct costs should be tracked in the model.
useFixedRandomSeed	Bit	Indicates whether a specified seed value for the random number generator should be used.
randomSeed	Long	The specified seed value for the random number generator.
saveAllDailyOutputs	Bit	Indicates whether daily outputs should be stored for every iteration.

Appendix E. Scenario file database schema

Table E-2. Fields of the database table **inGeneral**. (*continued*)

Field name	Data type	Description
saveDailyOutputsForIterations	Long	The number of iterations for which daily outputs should be stored. The minimum value is 3.
writeDailyStatesFile	Bit	Indicates whether a plain text file with the state of each unit on each day of each iteration should be written.
dailyStatesFileName	Text(255)	The file name of the plain text file described above.
saveDailyEvents	Bit	Indicates whether all events should be recorded in the scenario database.
saveDailyExposures	Bit	Indicates whether all exposures should be recorded in the scenario database.

E3.2. Database table **inProductionType**

The database table **inProductionType** contains data that applies to individual production types present in the scenario (*i.e.*, disease parameters; detection, destruction, and vaccination parameters; and cost parameters). Each production type will have its own record in this table, with fields as shown in Table E-3.

Table E-2. Fields of the database table **inProductionType**.

Field name	Data type	Description
productionTypeID	Counter	A unique ID for each production type present in the scenario.
descr	Text(255)	A unique description of the production type.
useDiseaseTransition	Bit	Indicates whether units of this production type will undergo disease transition.
disLatentPeriodPdfID	Long	ID number of the probability density function used to define the latent period for units of this production type. This field references <i>inChart.chartID</i> .
disSubclinicalPeriodPdfID	Long	ID number of the probability density function used to define the subclinical period for units of this production type. This field references <i>inChart.chartID</i> .
disClinicalPeriodPdfID	Long	ID number of the probability density function used to define the clinical period for units of this production type. This field references <i>inChart.chartID</i> .
disImmunePeriodPdfID	Long	ID number of the probability density function used to define the natural immune period for units of this production type. This field references <i>inChart.chartID</i> .

Table E-2. Fields of the database table **inProductionType**. (continued)

Field name	Data type	Description
useDetection	Bit	Indicates whether disease detection will be modeled for units of this production type.
detProbObsVsTimeClinicalRelID	Long	ID number of the relational function used to define the probability of observing clinical signs in units of this production type. This field references inChart.chartID.
detProbReportVsFirstDetectionRelID	Long	ID number of the relational function used to define the probability of reporting disease in units of this production type. This field references inChart.chartID.
survDirect	Bit	Indicates whether trace surveillance for direct contacts will be attempted.
survDirectSuccess	Double	The probability of success of identifying direct contacts by trace surveillance.
survDirectTracePeriod	Long	The period of time, in days, for which traceback surveillance of direct contacts will be attempted.
survIndirect	Bit	Indicates whether trace surveillance for indirect contacts will be attempted.
survIndirectSuccess	Double	The probability of success of identifying indirect contacts by trace surveillance.
survIndirectTracePeriod	Long	The period of time, in days, for which traceback surveillance of indirect contacts will be attempted.
useBasicDestruction	Bit	Indicates whether detected clinical units of this production type will be destroyed.
destrlsRingTrigger	Bit	Indicates whether detection of a unit of this type will trigger the formation of destruction ring.
destrRingRadius	Double	Radius, in kilometers, of the destruction ring.
destrlsRingTarget	Bit	Indicates whether units of this type will be subject to preemptive ring destruction.
destrDirectTraces	Bit	Indicates whether units of this type identified by traceback of direct contacts will be subject to preemptive destruction.
destrIndirectTraces	Bit	Indicates whether units of this type identified by traceback of indirect contacts will be subject to preemptive destruction.

Appendix E. Scenario file database schema

Table E-2. Fields of the database table **inProductionType**. (continued)

Field name	Data type	Description
destrPriority	Long	The destruction priority of this production type, relative to other production types. A lower number indicates higher priority.
useVaccination	Bit	Indicates whether units of this type will be subject to vaccination.
vaccMinTimeBetweenVaccinations	Long	The minimum time, in days, between vaccinations for units of this production type.
vaccDaysToImmunity	Long	The number of days required for the onset of vaccine immunity in a newly vaccinated unit of this type.
vaccImmunePeriodPdflD	Long	ID number of the probability density function used to define the vaccine immune period for units of this production type. This field references inChart.chartID.
vaccRing	Bit	Indicates whether detection of a clinical unit of this type will trigger a vaccination ring.
vaccRingRadius	Double	Radius, in kilometers, of the vaccination ring.
vaccPriority	Long	The vaccination priority of this production type, relative to other production types. A lower number indicates higher priority.
useCosts	Bit	Indicates whether direct costs will be tracked for units of this production type.
costDestrAppraisalPerUnit	Double	The cost associated with appraisal for each destroyed unit of this type.
costDestrCleaningPerUnit	Double	The cost of cleaning and disinfection for each destroyed unit of this type.
costDestrEuthanasiaPerAnimal	Double	The cost of euthanizing each destroyed animal of this type.
costDestrIndemnificationPerAnimal	Double	The cost of indemnification for each destroyed animal of this type.
costDestrDisposalPerAnimal	Double	The cost of carcass disposal for each destroyed animal of this type.
costVaccSetupPerUnit	Double	The cost of site setup for each vaccinated unit of this type.

Table E-2. Fields of the database table **inProductionType**. (continued)

Field name	Data type	Description
costVaccThreshold	Long	The number of animals of this type that can be vaccinated before the cost of vaccination increases.
costVaccBaselinePerAnimal	Double	The baseline cost of vaccination for each vaccinated animal of this type. This cost applies to all vaccinations before the threshold set in costVaccThreshold is met.
costVaccAdditionalPerAnimal	Double	The additional cost of vaccination for each vaccinated animal of this type, after the threshold is exceeded.

E3.3. Database table **inProductionTypePair**

The table **inProductionTypePair** identifies all selected combinations of source and recipient production type that have a role in disease spread, and identifies the mechanism or mechanisms of disease spread that apply to each production type combination. Each selected production type combination has its own record in this table, with fields as described in Table E-3.

Table E-3. Fields of the database table **inProductionTypePair**.

Field name	Data type	Description
sourceProductionTypeID	Long	The ID number of the production type that will be the source type for this production type combination. This field references inProductionType.productionTypeID .
destProductionTypeID	Long	The ID number of the production type that will be the recipient type for this production type combination. This field references inProductionType.productionTypeID .
useDirectContact	Bit	Indicates whether direct contact will occur from units of the source type to units of the recipient type.
directContactSpreadID	Long	ID number of the disease spread mechanism used to model spread by direct contact between these types. This field references inDiseaseSpread.spreadID .
useIndirectContact	Bit	Indicates whether indirect contact will occur from units of the source type to units of the recipient type.
indirectContactSpreadID	Long	ID number of the disease spread mechanism used to model spread by indirect contact between these types. This field references inDiseaseSpread.spreadID .

Appendix E. Scenario file database schema

Table E-3. Fields of the database table **inProductionTypePair**. (*continued*)

Field name	Data type	Description
useAirborneSpread	Bit	Indicates whether airborne spread will occur from units of the source type to units of the recipient type.
airborneContactSpreadID	Long	ID number of the disease spread mechanism used to model airborne spread between these types. This field references inDiseaseSpread.spreadID .

E3.4. Database table **inDiseaseSpread**

The database table **inDiseaseSpread** contains parameters that apply to all individual disease spread mechanisms (contact or airborne spread) used in the scenario. Records for the production type combinations (on table **inProductionTypePair**) refer to records on this table. Each production type combination for which disease spread occurs will have from one to several records on this table, with fields as shown in Table E-4.

Table E-4. Fields of the database table **inDiseaseSpread**.

Field name	Data type	Description
spreadID	Counter	A unique identifier for each disease spread "model".
spreadMethodCode	Char(1)	Indicates the disease spread mechanism being modeled: A = airborne spread, D = direct contact, I = indirect contact. This field references readSpreadMethodCodes.spreadMethodCode .
latentCanInfect	Bit	Indicates whether latent units of the source type can spread disease by direct contact. Not applicable to airborne spread or indirect contact.
subclinicalCanInfect	Bit	Indicates whether subclinical units of the source type can spread disease by direct or indirect contact.
meanContactRate	Double	The mean contact rate (in recipient herds per source herd per day) for direct or indirect contact models. See Section 8.6.3.2.
useFixedContactRate	Bit	Indicates whether a fixed contact rate will be used instead of the mean contact rate.

Table E-4. Fields of the database table **inDiseaseSpread**. (*continued*)

Field name	Data type	Description
fixedContactRate	Long	The fixed contact rate (in recipient herds per source herd per day) for direct or indirect contact models. See Section 8.6.3.2.
infectionProbability	Double	The probability that a contact will result in disease transmission. Specified for direct and indirect contact models.
distancePdfID	Long	ID number of the probability density function used to define the shipment distances for direct and indirect contact models. This field references inChart.chartID.
movementControlRelID	Long	ID number of the relational function used to define movement control effects for the indicated production type combination. This field references inChart.chartID.
transportDelayPdfID	Long	ID number of the probability density function used to define the shipment delays for direct and indirect contact models or airborne transport delay for airborne spread. This field references inChart.chartID. See Sections 8.6.3.3.1 and 8.6.4.
probAirborneSpread1km	Double	For airborne spread, the probability that disease will be spread to units 1 km away from the source unit. See Section 8.6.4.
maxDistAirborneSpread	Double	The maximum distance, in kilometers, of airborne spread.
windDirectionStart	Long	The start angle, in degrees, of the predominant wind direction for airborne spread. See Section 8.6.4.
windDirectionEnd	Long	The end angle, in degrees, of the predominant wind direction for airborne spread.

E3.5. Database table **inControlsGlobal**

The database table **inControlsGlobal** specifies disease control parameters that apply globally across all production types, such as destruction and vaccination capacity. This table contains only a single record, with fields as described in Table E-5.

Table E-5. Fields of the database table **inControlsGlobal**.

Field name	Data type	Description
controlsGlobalID	Char(10)	A primary key for the single record stored in this table. This value is always "NAADSMXXXX".
includeDetection	Bit	Indicates whether detection of disease in any production type will be modeled.
includeSurveillance	Bit	Indicates whether surveillance of any production type will be modeled.
includeDestruction	Bit	Indicates whether destruction will be used in any production type.
destrProgramDelay	Long	The number of days that must pass after the first detection before a destruction program can begin.
destrCapacityRelID	Long	ID number of the relational function used to define the daily destruction capacity. This field references inChart.chartID.
destrPriorityOrder	Text(255)	A string that identifies the primary priority order for destruction: see Section 8.9.2.1.
destrReasonOrder	Text(255)	A string that identifies the secondary priority order for destruction: see Section 8.9.2.1.
includeVaccination	Bit	Indicates whether vaccination will be used in any production type.
vaccDetectedUnitsBeforeStart	Long	The number of clinical units which must be detected before the initiation of a vaccination program.
vaccCapacityRelID	Long	ID number of the relational function used to define the daily vaccination capacity. This field references inChart.chartID.
vaccPriorityOrder	Text(255)	A string that identifies the priority order for vaccination: see Section 8.10.2.1.

E3.6. Database table **inChart**

The database table **inChart** contains data specific to each probability density function and each relational function used in a scenario (see Section 6.1.5, Section 6.1.6, and Appendix D). Each *pdf* or *rel* defined for a scenario will have a record in this table, with fields as shown in Table E-6.

Table E-6. Fields of the database table **inChart**.

Field name	Data type	Description
chartID	Counter	A unique ID for each function (<i>pdf</i> or relational function) defined for the scenario.
fieldName	Text(50)	Indicates the parameter (e.g. "disease latent period" or "movement effects over time") to which the function corresponds
chartName	Text(150)	Unique name assigned to each function.
isPdf	Bit	True if the function is a probability density function, false if a relational function.
chartType	Text(20)	For <i>pdfs</i> , identifies the type of function: see Appendix D.
mean	Double	The mean for <i>pdf</i> types Gaussian, Lognormal, and Exponential.
stddev	Double	The standard deviation for <i>pdf</i> types Gaussian and Lognormal.
min	Double	The minimum value for <i>pdf</i> types Uniform, Triangular, Beta, and BetaPERT.
mode	Double	The mode for <i>pdf</i> types Point, Triangular and BetaPERT.
max	Double	The maximum value for <i>pdf</i> types Uniform, Triangular, Beta and BetaPERT.
alpha	Double	The alpha parameter for <i>pdf</i> types Gamma, Weibull, and Pearson 5, or the <i>alpha1</i> parameter for Beta <i>pdfs</i> .
alpha2	Double	The alpha2 parameter for Beta <i>pdfs</i> .
beta	Double	The beta parameter for <i>pdf</i> types Gamma, Weibull, and Pearson 5.
location	Double	The location parameter for <i>pdf</i> types Logistic and Loglogistic.
scale	Double	The scale parameter for <i>pdf</i> types Logistic and Loglogistic.
shape	Double	The shape parameter for Loglogistic <i>pdfs</i> .
xAxisUnits	Text(50)	Specifies the units of the x axis for <i>pdfs</i> and <i>rels</i> .
yAxisUnits	Text(50)	Specifies the units of the y axis for <i>rel</i> functions.

E3.7. Database table **inChartDetail**

The database table **inChartDetail** contains the points used to define piecewise probability density functions and relational functions used in a scenario. Each piecewise pdf and each rel in table **inChart** will have several associated records in table **inChartDetail**, with fields as shown in Table E-7.

Table E-7. Fields of the database table **inChartDetail**.

Field name	Data type	Description
chartID	Long	The unique identifier for the piecewise pdf or relational function. This field references inChart.chartID.
pointOrder	Long	Indicates the order in which points are provided to define the function.
x	Double	The x value of the point indicated by pointOrder.
y	Double	The y value of the point indicated by pointOrder.

E4. The dynamic table **dynHerd**

The table **dynHerd** stores the herd/unit population data for a scenario. Each herd has its own record in this table. Table **dynHerd** plays the role of an input table and an output table: some fields are set up when the scenario is created and do not change as the simulation runs. Other fields on this table are updated as iterations run, for example, to reflect the current disease state of each herd. Some outputs (*i.e.*, the final disease state and the final apparent state) pertain to only a single iteration. Others are “cumulative” across several iterations. The fields of this table are shown in Table E-8.

Table E-8. Fields of the database table **dynHerd**.

Field name	Data type	Description
herdID	Counter	(input) A unique ID number for each herd/unit in the population.
productionTypeID	Long	(input) The ID number of the production type of this herd. This field references inProductionType.productionTypeID.
latitude	Double	(input) The latitude used to georeference this herd.
longitude	Double	(input) The longitude used to georeference this herd.

Table E-8. Fields of the database table **dynHerd**. (*continued*)

Field name	Data type	Description
initialStateCode	Char(1)	(input) The actual disease state of the herd at the beginning of the simulation. This field references readStatusCodes.statusCode.
daysLeftInInitialState	Long	(input) The number of days that the herd will remain in its initial state, unless preempted by other events. See Section 8.4.1.2.
initialSize	Long	(input) The number of animals in the herd.
finalStateCode	Char(1)	(output for the most recent iteration) The actual disease state of the herd at the end of the simulation. This field references readStatusCodes.statusCode.
finalApparentStateCode	Char(1)	(output for the most recent iteration) The apparent state of the herd at the end of the simulation. This field references readApparentStateCodes.apparentStateCode.
cumInfected	Long	(output for all iterations) The total number of iterations in which this herd became infected.
cumDestroyed	Long	(output for all iterations) The total number of iterations in which this herd was destroyed.
cumVaccinated	Long	(output for all iterations) The total number of iterations in which this herd was vaccinated.

E5. Database tables for scenario output

As described in Section 10 and Appendix B, *NAADSM* outputs may apply to a single iteration or may summarize output from multiple iterations. There are eight output tables in each *NAADSM* scenario file. Three of these tables (**outDailyByProductionType**, **outDailyEvents**, and **outDailyExposures**) store data for each day of one or more iterations.

If daily information is stored for very many iterations, the tables mentioned above would be huge. Consequently, a separate mechanism is used to store only cumulative data for each iteration. Four tables (**outIteration**, **outIterationByProductionType**, **outIterationCosts**, and **outEpidemicCurves**) contain this kind of cumulative data.

Finally, one more table (**outGeneral**) stores output pertaining to the complete simulation run, such as the duration of the simulation and the number of complete iterations.

E5.1. Database table **outGeneral**

The database table **outGeneral** contains output pertaining to the complete simulation run, such as the duration of the simulation, and the number of iterations that ran to completion. There is a single record in this table, with fields as shown in Table E-9.

Table E-9. Fields of the database table **outGeneral**.

Field name	Data type	Description
outGeneralID	Char(10)	A primary key for the single record stored in this table. This value is always "NAADSMXXXX".
simulationStartTime	Datetime	The actual clock time according to the computer used to run the simulation when the simulation was launched.
simulationEndTime	Datetime	The actual clock time according to the computer used to run the simulation when the simulation ended.
completedIterations	Long	The number of iterations completed during the simulation run.
version	Text(50)	The number of the NAADSM version used to run the simulation.

E5.2. Daily data tables

E5.2.1. Database table **outDailyByProductionType**

The database table **outDailyByProductionType** stored output generated for each day of the last several iterations, for each production type. By default, daily data for only the last three iterations of a scenario will be stored, although users may select to store more daily information (see Section 8.12). This table will have a separate record for each production type on each day of each iteration. Fields in this table are shown in Table E-10.

Table E-10. Fields of the database table **outDailyByProductionType**.

Field name	Data type	Description
iteration	Long	The iteration during which the outputs in this record were generated. The first iteration is iteration number 1.
day	Long	The day within the iteration on which these outputs were generated. The first day of an iteration is day number 1.
productionTypeID	Long	The ID number of the production type to which the outputs in this record apply. This field references inProductionType.productionTypeID.

All other fields in this table are of data type Long and are described in Appendix B, Table B-3.

E5.2.2. Database table **outDailyEvents**

When data for daily events is stored (see Section 10.3), the database table **outDailyEvents** is populated. Each event will have a separate record in this table, with fields as shown in Table E-11.

Table E-11. Fields of the database table **outDailyEvents**.

Field name	Data type	Description
iteration	Long	The iteration during which the outputs in this record were generated. The first iteration is iteration number 1.
day	Long	The day within the iteration on which these outputs were generated. The first day of an iteration is day number 1.
event	Long	A number used, in conjunction with iteration and day, to uniquely identify each event.
herdID	Long	The ID number of the herd/unit for which this event occurred. This field references dynHerd.herdID.

Appendix E. Scenario file database schema

Table E-11. Fields of the database table **outDailyEvents**. (*continued*)

Field name	Data type	Description
eventCode	Char(1)	An event code indicating the type of event. This field references readEventCodes.eventCode.
newStateCode	Char(1)	For transition state changes, this field indicates the state that results from the event. This field references readStatusCodes.statusCode.
traceSuccess	Bit	For trace events, this field indicates whether the attempted trace succeeded.

E5.2.2. Database table **outDailyExposures**

When data for daily exposures is stored (see Section 10.3), the database table **outDailyExposures** is populated. Each exposure will have a separate record in this table, with fields as shown in Table E-12.

Table E-12. Fields of the database table **outDailyExposures**.

Field name	Data type	Description
iteration	Long	The iteration during which the outputs in this record were generated. The first iteration is iteration number 1.
day	Long	The day within the iteration on which these outputs were generated. The first day of an iteration is day number 1.
exposure	Long	A number used, in conjunction with iteration and day, to uniquely identify each exposure.
exposedHerdID	Long	The ID number of the source herd/unit for the exposure. This field references dynHerd.herdID.
exposingHerdID	Long	The ID number of the recipient herd/unit for the exposure. This field references dynHerd.herdID.
spreadMethodCode	Char(1)	A code indicating the mechanism of disease spread. This field references readSpreadMethodCodes.spreadMethodCode.
success	Bit	Indicates whether the exposure is adequate to transmit disease. See Section 10.3.2 for further discussion.
exposingHerdStatusCode	Char(1)	Disease state of the exposing herd when the exposure occurred. This field references readStatusCodes.statusCode.
exposedHerdStatusCode	Char(1)	Disease state of the exposed herd when the exposure occurred. This field references readStatusCodes.statusCode.

E5.3. Iteration data tables

E5.3.1. Database table **outIteration**

The database table **outIteration** contains output pertaining to each complete iteration. There will be one record in this table for each iteration that is completed, with fields as shown in Table E-13.

Table E-13. Fields of the database table **outIteration**.

Field name	Data type	Description
iteration	Long	The iteration during which the outputs in this record were generated. The first iteration is iteration number 1.
length	Long	The duration in days of the outbreak simulated in the specified iteration.
outbreakEnded	Bit	Indicates whether the simulated outbreak ended before the end of the iteration: see Section B2.

E5.3.2. Database table **outIterationByProductionType**

Database table **outIterationByProductionType** contains epidemiological output generated for each individual iteration, for each production type. Every production type will have a separate record in this table for each completed iteration. Fields in table **outIterationByProductionType** are described in Table E-14.

Table E-14. Fields of the database table **outIterationByProductionType**.

Field name	Data type	Description
iteration	Long	The iteration during which the outputs in this record were generated. The first iteration is iteration number 1.
productionTypeID	Long	The ID number of the production type to which the outputs in this record apply. This field references inProductionType.productionTypeID.

All other fields in this table are of data type Long and are described in Appendix B, Table B-1.

E5.3.3. Database table **outIterationCosts**

Database table **outIterationCosts** contains direct cost accounting output generated for each individual iteration, if direct costs are being tracked (see Section 8.11). Every production type will have a separate record in this table for each completed iteration. Fields in table **outIterationCosts** are described in Table E-15.

Table E-15. Fields of the database table **outIterationCosts**.

Field name	Data type	Description
iteration	Long	The iteration during which the outputs in this record were generated. The first iteration is iteration number 1.
productionTypeID	Long	The ID number of the production type to which the outputs in this record apply. This field references inProductionType.productionTypeID.

All other fields in this table are of data type Currency and are described in Appendix B, Table B-2.

E5.3.4. Database table **outEpidemicCurves**

Information needed to generate an epidemic curve for each iteration is stored in the database table **outEpidemicCurves**. The number of infected herds on each day of each iteration and the number of newly detected cases on each day of each iteration is stored in this table. The daily number of infections may be used to generate an actual epidemic curve; the daily number of detections may be used to generate an apparent epidemic curve. The total number of units as well as the total number of animals in those units is recorded. Every production type has a record for every day of every iteration in this table. Field descriptions are shown in Table E-16.

Table E-16. Fields of the database table **outEpidemicCurves**.

Field name	Data type	Description
iteration	Long	The iteration during which the outputs in this record were generated. The first iteration is iteration number 1.
day	Long	The day within the iteration on which these outputs were generated. The first day of an iteration is day number 1.
productionTypeID	Long	The ID number of the production type to which the outputs in this record apply. This field references inProductionType.productionTypeID.
infectedUnits	Long	The number of units of the specified production type infected by any mechanism on the specified day in specified iteration.
infectedAnimals	Long	Total number of animals in units described immediately above.
detectedUnits	Long	The number of clinically ill units of the specified production type detected by any mechanism on the specified day in specified iteration.
detectedAnimals	Long	Total number of animals in units described immediately above.

E6. Read-only tables

Several read-only or lookup tables are used in each *NAADSM* scenario file. These tables serve as references for codes frequently used in other tables.

E.6.1. Database table **readStatusCodes**

Database table **readStatusCodes** stores codes used to indicate an disease or disease-like transition state. There are two fields and always exactly seven records in **readStatusCodes**, as shown in Tables E-17 and E-18. These codes are used in the database tables **dynHerd**, **outDailyEvents**, and **outDailyExposures**.

Table E-17. Fields of the database table **readStatusCodes**.

Field name	Data type	Description
statusCode	Char(1)	Single character code used to indicate an actual disease state: see Table E-17.
definition	Text(255)	Description of the disease state.

Table E-18. Values in the database table **readStatusCodes**.

statusCode	definition
S	Susceptible
L	Latent
B	Subclinical
C	Clinical
N	Naturally immune
V	Vaccine immune
D	Destroyed

E.6.2. Database table **readApparentStateCodes**

Database table **readApparentStateCodes** stores codes used to indicate the known or apparent state of a herd (as opposed to the actual or potentially undetectable disease state of a herd). There are two fields and always exactly six records in **readStatusCodes**, as shown in Tables E-19 and E-20. These codes are used in the database table **dynHerd**.

Table E-19. Fields of the database table **readApparentStateCodes**.

Field name	Data type	Description
apparentStateCode	Char(1)	Single character code used to indicate an apparent state: see Table E-20.
definition	Text(255)	Description of the apparent state.

Table E-20. Values in the database table **readApparentStateCodes**.

apparentStateCode	definition
D	Destroyed
E	Detected by clinical signs
I	Detected by trace - indirect
T	Detected by trace - direct
V	Vaccinated
U	Unknown ¹

¹ At the start of every iteration, the initial apparent state of every herd is “unknown”

E.6.3. Database table **readEventCodes**

Database table **readEventCodes** stores codes used to indicate the types of events that may affect individual units (see Section 10.3). There are two fields and exactly seven records in **readEventCodes**, as shown in Tables E-21, E-22, and Section 10 Table 10-1. These codes are used in database table **outDailyEvents**.

Table E-21. Fields of the database table **readEventCodes**.

Field name	Data type	Description
eventCode	Char(1)	Single character code used to indicate an event: see Table E-22.
definition	Text(255)	Description of the event.

Table E-22. Values in the database table **readEventCodes**.

eventCode	definition
D	Destruction
E	Detection
F	Infection
I	Trace of indirect contact
R	Transition state change
T	Trace of direct contact
V	Vaccination

E.6.4. Database table **readSpreadMethodCodes**

Database table **readSpreadMethodCodes** stores codes used to indicate the mechanisms that can cause disease spread. There are two fields and exactly three records in **readSpreadMethodCodes**, as shown in Tables E-23, and E-24. These codes are used in the database tables **inDiseaseSpread** and **outDailyExposures**.

Table E-23. Fields of the database table **readSpreadMethodCodes**.

<i>Field name</i>	<i>Data type</i>	<i>Description</i>
spreadMethodCode	Char(1)	Single character code used to indicate a method of disease spread: A = airborne spread, D = spread by direct contact, I = spread by indirect contact.
definition	Text(255)	Description of the method code.

Table E-24. Values in the database table **readSpreadMethodCodes**.

<i>spreadMethodCode</i>	<i>definition</i>
A	Airborne spread
D	Direct contact
I	Indirect contact

Appendix F. *NAADSM/PC* and *NAADSM/SC*

F1. The two implementations of *NAADSM*

There are two distinct but fully compatible implementations of *NAADSM*. *NAADSM/PC* (“personal computer”) is available for *Microsoft Windows*, and includes the complete interactive user interface described in this manual. *NAADSM/SC* (“supercomputer”) runs on a variety of *Linux/Unix* variants, including *Mac OS X*. While *NAADSM/SC* will run on a stand-alone computer with a *Linux/Unix* operating system, it is really intended for use on Message Passing Interface (MPI)-based grid, cluster, or parallel computing systems. *NAADSM/SC* is designed to process a large number (thousands) of iterations involving very large population datasets (tens or hundreds of thousands of herds). An average stand-alone personal computer running either implementation of *NAADSM* will comfortably accommodate 100 iterations of a simulation involving up to roughly 20,000 herds (such a simulation might take between roughly 10 minutes and 2 hours to run to completion).

Precisely the same conceptual model applies to both of these implementations, and the core code of both applications is identical.

F2. Building *NAADSM/SC*

As is the case with much *Linux/Unix* software, *NAADSM/SC* must be compiled on the system on which it will be run. *NAADSM/SC* can be compiled with the GNU Compiler Collection (*GCC*) version 3.3.3 or higher. Other compilers may work, but have not been tested. *NAADSM/SC* source code can be obtained directly from the development team (see Appendix G for current contact information). *NAADSM/SC* also requires several development libraries, most of which are included in standard *Linux/Unix* installations. A complete list of the required libraries is available at <http://hebb.cis.uoguelph.ca/~dastacey/Grid/ERG_ADM/licenses>.

F3. Developing scenarios for *NAADSM/SC*

Most users will find that scenario development is best performed using the graphical interface of *NAADSM/PC*. Once parameters have been defined and the scenario has been tested with a few iterations, the scenario can be exported for use with *NAADSM/SC* (see Section 7.3).

NAADSM/SC parameter input files are created in plain-text extensible markup language (*.**xml**) format. *NAADSM/PC* will automatically generate most of the required XML. Complete document type definitions (DTDs) are included with the *NAADSM/SC* source code.

Users must specify in their *.**xml** documents the outputs that they would like to have returned by *NAADSM/SC*. *NAADSM/PC* does not yet generate XML with a “standard” set of outputs, but this feature may be included in the future.

F4. Viewing output from *NAADSM/SC*

Unlike the *Microsoft Windows* version of *NAADSM*, *NAADSM/SC* does not store output in a database file. Please contact the appropriate members of the development team (see Appendix G) for information about *NAADSM/SC* output formats.

It is not yet possible to import *NAADSM/SC* output into a scenario database file that could be viewed with *NAADSM/PC*. This capability may be added to future versions of *NAADSM*.

Appendix G. Development team contact information

<i>For information or questions concerning:</i>	<i>Please contact:</i>
<ul style="list-style-type: none"> The NAADSM project in the United States Animal disease modeling initiatives at USDA 	Dr. Barbara Corso USDA-APHIS-VS-CEAH-CADIA 2150 Centre Ave., Bldg. B, Mail Stop 2W4 Fort Collins, CO 80526-8117 <Barbara.A.CORSO@aphis.usda.gov>
<ul style="list-style-type: none"> The NAADSM project in Canada Animal disease modeling initiatives at CFIA 	Dr. Caroline Dubé CFIA-Animal Health and Production Division 59 Camelot Ottawa, ON, Canada K1A 0Y9 <dubecm@inspection.gc.ca>
<ul style="list-style-type: none"> NAADSM for Microsoft Windows The NAADSM user interface Unexpected errors or problems running NAADSM on Windows 	Aaron Reeves Animal Population Health Institute Dept. of Clinical Sciences, Colorado State University Fort Collins, CO 80523 <Aaron.Reeves@colostate.edu>
<ul style="list-style-type: none"> NAADSM for Linux/Unix Parallel processing with NAADSM Unexpected errors or problems running NAADSM on Linux/Unix 	Neil Harvey Dept. of Computing & Information Science University of Guelph Guelph, ON, Canada N1G 2W1 <nharvey@uoguelph.ca>
<ul style="list-style-type: none"> This User's Guide 	Aaron Reeves <Aaron.Reeves@colostate.edu> or Dr. Ashley Hill Animal Population Health Institute Dept. of Clinical Sciences, Colorado State University Fort Collins, CO 80523 <Ashley.Hill@colostate.edu>
<ul style="list-style-type: none"> The NAADSM conceptual model and specification (see Appendix A) 	Any team member
<ul style="list-style-type: none"> Advanced use of NAADSM scenario database files Data import/export to other applications or databases 	Aaron Reeves <Aaron.Reeves@colostate.edu>
<ul style="list-style-type: none"> Information for application developers 	Neil Harvey <nharvey@uoguelph.ca> or Aaron Reeves <Aaron.Reeves@colostate.edu>

(continued on the next page)

Appendix G. Contact information

<i>For information or questions concerning:</i>	<i>Please contact:</i>
<ul style="list-style-type: none">• Graduate degree programs in the Animal Population Health Institute at Colorado State University	Dr. M.D. Salman Animal Population Health Institute Dept. of Clinical Sciences, Colorado State University Fort Collins, CO 80523 <M.D.Salman@colostate.edu> < http://www.cvmbs.colostate.edu/aphi >
<ul style="list-style-type: none">• Degree programs in Computing & Information Science at the University of Guelph	Dr. Deborah A. Stacey Dept. of Computing & Information Science University of Guelph Guelph, ON, Canada N1G 2W1 < dastacey@cis.uoguelph.ca > < http://www.cis.uoguelph.ca >

Appendix H. The Creative Commons Public License

General terms of the Attribution-NonCommercial-ShareAlike 2.5 license

Permission is granted to copy, distribute, display, and/or to make derivative works of this document, under the following conditions: the work must be attributed in the manner specified by the author or licensor; the work may not be used for commercial purposes; and any derived works may be distributed only under a license identical to this one.

Creative Commons web site

The Creative Commons web site is <<http://creativecommons.org>>.

More information regarding the **Attribution-NonCommercial-ShareAlike 2.5 license** is available at <<http://creativecommons.org/licenses/by-nc-sa/2.5>>.

Legal code of the Attribution-NonCommercial-ShareAlike 2.5 license

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

- a. "**Collective Work**" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.
- b. "**Derivative Work**" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.
- c. "**Licensor**" means the individual or entity that offers the Work under the terms of this License.
- d. "**Original Author**" means the individual or entity who created the Work.
- e. "**Work**" means the copyrightable work of authorship offered under the terms of this License.
- f. "**You**" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.
- g. "**License Elements**" means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, Noncommercial, ShareAlike.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

- a. to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
- b. to create and reproduce Derivative Works;
- c. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;
- d. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works;

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved, including but not limited to the rights set forth in Sections 4(e) and 4(f).

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

- a. You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(d), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(d), as requested.
- b. You may distribute, publicly display, publicly perform, or publicly digitally perform a Derivative Work only under the terms of this License, a later version of this License with the same License Elements as this License, or a Creative Commons iCommons license that contains the same License Elements as this License (e.g. Attribution-NonCommercial-ShareAlike 2.5 Japan). You must include a copy of, or the Uniform Resource Identifier for, this License or other license specified in the previous sentence with every copy or phonorecord of each Derivative Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Derivative Works that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder, and You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Derivative Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Derivative Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Derivative Work itself to be made subject to the terms of this License.
- c. You may not exercise any of the rights granted to You in Section 3 above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation. The exchange of the Work for other copyrighted works by

Appendix H. The Creative Commons Public License

means of digital file-sharing or otherwise shall not be considered to be intended for or directed toward commercial advantage or private monetary compensation, provided there is no payment of any monetary compensation in connection with the exchange of copyrighted works.

- d. If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.
- e. For the avoidance of doubt, where the Work is a musical composition:
 - i. **Performance Royalties Under Blanket Licenses.** Licensor reserves the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work if that performance is primarily intended for or directed toward commercial advantage or private monetary compensation.
 - ii. **Mechanical Rights and Statutory Royalties.** Licensor reserves the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions), if Your distribution of such cover version is primarily intended for or directed toward commercial advantage or private monetary compensation.
- f. **Webcasting Rights and Statutory Royalties.** For the avoidance of doubt, where the Work is a sound recording, Licensor reserves the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions), if Your public digital performance is primarily intended for or directed toward commercial advantage or private monetary compensation.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING,
LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR

WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

- a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.
- b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

- a. Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
- b. Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
- c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
- d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

Appendix H. The Creative Commons Public License

- e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

Appendix I. The GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Appendix I. The GNU General Public License

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a)** You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b)** You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c)** If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a)** Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b)** Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c)** Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

Appendix I. The GNU General Public License

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this

License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

