User's Guide for the
North American Animal Disease Spread Model 4.0

For *NAADSM* version 4.0.10
4th edition

Aaron Reeves*, Ric Hupalo,
Kelly A. Patyk, and Ashley E. Hill

* *Author for correspondence*
Department of Clinical Sciences
College of Veterinary Medicine & Biomedical Sciences
Colorado State University
Fort Collins, Colorado 80523
<*Aaron.Reeves@colostate.edu*>

Table of contents

List of figures

List of tables

List of hints and warnings

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 **`SampleScenario.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.

1. Introduction to *NAADSM*

The *North American Animal Disease Spread Model* (*NAADSM*) is a computer program for the development of simulation models of the spread of highly contagious animal diseases. *NAADSM* provides a way for researchers, policy makers, and animal health emergency responders to devise and evaluate strategies for the control of diseases like foot-and-mouth disease, classical swine fever, avian influenza, and other similar diseases in commercial livestock and poultry populations. 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, and to estimate the resources that might be needed for disease control in the event of an outbreak.

NAADSM is not a single model of a single disease or population. Instead the *NAADSM* application provides a flexible framework in which user-established parameters are used to define model behavior in terms of disease progression; disease spread by animal-to-animal contact, contact with contaminated personnel or equipment, and airborne dissemination; and the implementation of control measures such as destruction and vaccination. Users may construct scenarios ranging from relatively simple to quite complex, to assist policy development and decision-making, to conduct training, or to develop emergency preparedness exercises.

NAADSM 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 over a decade. The animal disease modeling project at CEAH began with the evaluation of models developed elsewhere and grew to encompass the creation of several versions of a new animal disease model for use in North America (Schoenbaum and Disney, 2003). Each of these new versions was developed with the primary objective of producing a comprehensive, user-friendly modeling application.

NAADSM was first released to the public in 2006. The following year, a description of *NAADSM* appeared in the scientific journal *Preventive Veterinary Medicine* (Harvey *et al.*, 2007). Version 4.0 is the latest in this line of applications, and is a much enhanced successor to earlier versions. This version has been developed through a continuing international collaboration involving researchers from the United States and Canada, along with invaluable contributions from subject matter experts from around the world.

Although the *NAADSM* project originated in North America, its application is not restricted to North America. Since its release, hundreds of investigators have evaluated and used *NAADSM* to explore and evaluate epidemiologic questions of interest in over 30 countries on every continent except Antarctica. A current bibliography of published studies conducted with *NAADSM* may be found on the *NAADSM* website at <*http://www.naadsm.org*>.

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 visit the *NAADSM* website at <*http://www.naadsm.org*>. The website is updated continually with new or improved software

packages, valuable data, and materials for epidemiologic modelers in general, and specifically for users of *NAADSM*.

1.1. Purposes of this guide

This guide is designed to acquaint researchers, analysts, and decision makers in the fields of veterinary epidemiology and animal health economics 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 a *NAADSM* simulation.

This guide describes the mechanisms simulated by *NAADSM* and the assumptions behind these mechanisms. 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.

This manual is also 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 rather than a beginning-to-end "how to" resource. The appendices and other supplementary information provided online offer comprehensive lists 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.

A final purpose of this guide is to describe the assumptions used by *NAADSM* regarding mechanisms of disease progression, spread, and control. It is critical for every model user and every interpreter of model output to have a thorough understanding of these assumptions in order to assess the validity of the scenarios being modeled and of their outcomes. A discussion of model validity is beyond the scope of the *NAADSM User's Guide*, but readers are directed to several other sources and the references cited therein for more on this very important topic (*NAADSM* Development Team, 2009; Reeves et al., 2011).

Box 1-1. Assumptions made during modeling

NAADSM is certainly not unique in making a set of assumptions about disease processes, spread, and control. Every epidemiologic model, regardless of its form or the level of detail that it incorporates, is built on a similar set of assumptions.

Some of these assumptions, like those discussed throughout this guide, are inherent in the modeling framework (in this case, within the framework provided by *NAADSM*). Other assumptions are imposed by model users in their selections of parameters and parameter values. It is the responsibility of the model user first to understand these assumptions for him/herself; and second, to clearly document and justify them, so that others can independently assess the validity of the assumptions made.

1.1.1. Changes for the fourth edition of the *User's Guide*

Every chapter of the fourth edition of the *User's Guide for the North American Animal Disease Spread Model* has been updated to discuss new capabilities of *NAADSM* that have been introduced since the last edition. *NAADSM* 4.0 represents, in some ways, a substantial break from earlier versions. New and dramatically altered features of *NAADSM* 4.0 are described in detail. Chapter 3, which is devoted to a discussion of the differences between *NAADSM* 3.*x* and *NAADSM* 4.0, is entirely new.

Additionally, Chapter 9 has been updated to describe new features of *NAADSM* 4.0. A completely new section (Section 9.6.4) describes local-area spread. Sections 9.6.5 and 9.7 have been completely revised to describe new mechanisms incorporated in *NAADSM* 4 for airborne spread and the use of within-herd prevalence.

As in the third edition of the *Guide*, nearly every section of this chapter begins with a subsection that discusses a particular capability of *NAADSM* in general, conceptual terms (for example, see Section 9.9.1 for a general description of tracing capabilities in *NAADSM*), which provide context for the more detailed descriptions of model parameters that follow.

Throughout the guide, the notes, hints, and warnings of the third edition have been preserved and supplemented. These reflect some of the lessons learned by experience with the model framework over the six years since its introduction. This commentary is intended to provide practical advice; to help users to avoid commonly encountered pitfalls; and to guide careful, critical thinking about the application of model parameters and the techniques of epidemiological simulation modeling in general.

As with the third edition, several of the appendices found in previous versions of the guide (such as the *NAADSM* model specification and definitions of model outputs) have been replaced with online content, so that this information can be updated with less effort, ensuring that users always have the most recent and complete information available. References to the relevant pages on the *NAADSM* website, <*http://www.naadsm.org*>, have been integrated into the text.

1.2. 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- or flock-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. A fictitious sample dataset is included with the program, so that users may familiarize themselves with model parameters and capabilities.

NAADSM is intended to be used as a research and planning tool in advance of an incursion of a highly contagious disease, and to serve as one source of information for the formulation of disease preparedness and response strategies.

1.3. Limitations of *NAADSM*

As noted above, *NAADSM* is based on a specific set of assumptions about disease progression, disease spread, and mechanisms of disease control. These assumptions will not be valid for every situation. *NAADSM* is inappropriate, for example, for the simulation of chronic, endemic, vertically transmitted, sexually transmitted, or vector-borne diseases; diseases with environmental or biologically adapted reservoirs; or diseases that do not confer post-infection immunity. Although it may be tempting to try to apply *NAADSM* as it currently exists to such situations, such an attempt would be inadvisable.

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, and sufficient detail for a simulation model like *NAADSM* is rarely available in emergency situations. Simulated outbreaks based on inaccurate information are not likely to be useful, and may provide a false sense of security regarding the scope and control of the outbreak.

Current versions of *NAADSM* are most suitable for use within regions that can be reasonably represented by a single set of model parameters, *i.e.*, regions that are characterized by consistent similarity throughout their extent with respect to the types of animals in the population, animal management practices, and disease control policies and capabilities. It is the responsibility of the modeler to determine the extent of a region for which application of *NAADSM* is appropriate.

1.4. Overview of simulation parameters

Users should be familiar with the parameters described in the *NAADSM* model specification see <*http://naadsm.org/documentation/specification*>. This document gives a complete, detailed list of all input parameters, and describes how each parameter is used in the course of a simulation.

> *Box 1-2. The* NAADSM *model specification*
>
> The *NAADSM* specification, available at <*http://naadsm.org/documentation/specification*>, is the most definitive source regarding the kinds of information required for the *NAADSM* framework. It also describes in great detail how the model functions.
>
> *Every* NAADSM *user should have a high degree of familiarity with the model specification.* Although ancillary sources, like this guide and other technical papers (a variety of which are available at <*http://naadsm.org/techpapers*> contain valuable information and commentary, they should be considered secondary to the specification.

Chapter 9 of this guide gives detailed information regarding how these parameters are entered into *NAADSM*. Input parameters apply to the following broad categories:

Animal populations. Individual herds, flocks, or other operational units or premises are referred to as "herds" in this document. *NAADSM* operates on a population of herds, which 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. Production types are typically defined by animal species and/or by the management practices that are applied to each herd. Production types may be broad (*e.g.*, cattle, swine, poultry) or narrow (*e.g.*, beef sale yard, beef cow-calf operations with fewer than 50 animals, beef cow-calf operation with more than 50 animals, or beef feedlot), depending on the needs or preferences of the modeler. The user defines the rates of contact and the probabilities of airborne transmission of disease within and between production types to mimic disease spread to or from herds of different types.

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.

Disease transmission. *NAADSM* was originally 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 establish 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. User-defined parameters specify the probabilities of observing clinical signs of disease and reporting an observation. Additionally, once a herd is determined to be infected, tracing backward and forward to herds having direct and indirect contacts with the infected herd can be performed. Herds identified by tracing can be examined for clinical signs, or subjected to diagnostic testing for detection of disease.

Disease control. *NAADSM* simulates several methods of disease control, including quarantine, movement restriction, destruction/depopulation, and ring vaccination. 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 destroyed or vaccinated per day.

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

1.5. 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, numbers of herds and animals infected, numbers of herds and animals vaccinated, costs associated with vaccination, *etc.*). Model outputs are discussed in more detail in Chapter 11. A complete description of all available model outputs is provided on the *NAADSM* website, <*http://www.naadsm.org/documentation*>.

NAADSM summarizes the results over all iterations using histograms, measures of central tendency and spread, and other relevant statistics.

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 C.

NAADSM/PC is supported on *Microsoft Windows XP*, *Windows Vista*, *Windows 7*, and higher. Although basic systems that meet the minimum requirements for these operating systems are sufficient to run *NAADSM*, faster computers with more available memory will allow users to run larger and more sophisticated scenarios.

Box 2-1. Limitations on NAADSM *scenario database file size*

Each *NAADSM/PC* scenario (which includes input parameters and outputs generated when the scenario is run) is stored in its own *Microsoft Access*-compatible database file. The maximum size of each database file, as imposed by *Microsoft Access*, is two gigabytes. In some situations, this cap will limit the number of herds that can be included in a scenario or the number of iterations of the scenario that can be run.

If you encounter these limitations, you may wish to contact a member of the *NAADSM* Development Team. Contact information is provided on the *NAADSM* website.

3. Changes from *NAADSM* 3.*x* to *NAADSM* 4.0

This chapter is intended for users familiar with *NAADSM* 3.x, in order to point out the major differences between previous versions and *NAADSM* 4.0. Users new to *NAADSM* with version 4 may safely skip this chapter.

3.1. Compatibility of *NAADSM* 3.*x* and *NAADSM* 4.0

NAADSM 4 is the first public release of the *NAADSM* modeling framework that is not 100% backward-compatible with previous releases. (*NAADSM* 3.1 and 3.2 were supersets of capabilities proved by the initial public releases of *NAADSM* 3.0.x in 2006).

Several substantial improvements to the modeling framework in *NAADSM* 4 have been made, as described in the following sections. These changes have necessitated, however, the removal or substantial revision of some parameters and capabilities present in *NAADSM* 3. It is possible to import an existing *NAADSM* 3 scenario in *NAADSM* 4, but it is not possible to run such a scenario without first modifying several key parameters for the new version.

The definitive sources of information for changes between *NAADSM* 3 and *NAADSM* 4 are the updated model description document (*NAADSM* Development Team, 2012) and the accompanying requests for comment, all of which may be found at <*http://www.naadsm.org/documentation/specification*>. This chapter provides several brief overviews of the modifications between these two major versions.

3.1.1. Continued support for *NAADSM* 3.*x*

Although the *NAADSM* Development Team believes that the changes introduced in *NAADSM* 4 are helpful (and it is the opinion of the authors of this guide that users should consider moving to *NAADSM* 4 as soon as possible), it is anticipated that *NAADSM* 3.2 and *NAADSM* 4 will be maintained and simultaneously supported for the foreseeable future. Apart from bug fixes, however, no further improvements to the *NAADSM* 3 line are expected. The development of any new capabilities will likely occur in *NAADSM* 4 and subsequent versions.

3.2. Changes to local-area and airborne spread

In previous versions of *NAADSM*, users were restricted in their ability to represent non-directional local-area spread of disease (the spread of disease among premises in relatively close proximity to one another by mechanisms that are not readily traced, such as localized aerosol transmission, insects or vermin acting as mechanical fomites, or "across the fence" contact). Users could simulate directional airborne spread of disease, or non- directional local-area spread, but not both. In *NAADSM* 4, these two mechanisms are now parameterized separately.

In addition, the approach to calculating the probability of local-area and longer-distance directional airborne transmission of disease has been completely revamped. In previous versions of *NAADSM*, local-area/airborne spread was dependent not only on the source and recipient units involved, but was also influenced by the size of all other units in the study population. This dependence on the entire population not only had little biological or epidemiological justification, but made it very difficult to legitimately use parameters developed for one particular study in any other study that was not based on exactly the same study population. Also, using the previous approach to handling local-area/airborne spread, it was possible to obtain nonsensical probabilities of disease transmission of more than 100%. The new approach used in *NAADSM* 4 eliminates all of these problems.

3.3. Changes to within-herd prevalence of disease

The capability to represent the impact of within-unit prevalence of disease on spread by direct contact and by airborne/local-area transmission was introduced in *NAADSM* 3.1. This version did not, however, account for the different ways in which within-unit prevalence affected these distance mechanisms. Epidemiologically, it makes sense that local area/airborne transmission might be influenced by the prevalence of animals in a unit that are shedding the infectious agent. By contrast, as discussed in Section 9.6.3.1, latent individuals can also contribute to disease spread by direct contact (*i.e.*, the introduction of an infected animal from one unit into another). The prevalences of shedding animals versus all infected animals may be quite different. Furthermore, the probability of disease spread by direct contact is influenced not only by the prevalence of infected animals, but also by the number of animals introduced from an infected unit into another: all other parameters being equal, shipments of more animals are more likely to result in disease transmission than shipments involving fewer animals. *NAADSM* 4 provides improved capabilities for simulating the effects of within-unit prevalence on disease spread.

3.4. Introduction of a “dead from disease” state

Prior versions of *NAADSM* did not incorporate any notion of disease mortality, which limited their applicability to diseases associated with high levels of mortality, such as avian influenza. *NAADSM* 4 incorporates a new herd-level disease state called “dead from disease”. Users can now specify the probability with which infected units will die from disease or recover and progress to the naturally immune state.

Because *NAADSM* 4 is still a herd-based model, the most appropriate interpretation of the “dead from disease” state is that all or essentially all animals in the unit have died. This disease state is most applicable to small herds or flocks, which are be more likely to be entirely wiped out by an outbreak of disease. The utility of the “dead from disease” state for production types characterized by large units should be carefully considered by the modeler.

3.5. Elimination of exposure delay parameters

NAADSM 3 incorporated several delay parameters for direct, indirect, and airborne transmission of disease, initially designed to account for the time it takes for disease to be transmitted from a source to a recipient unit: for example, long-distance shipments might take longer to arrive at their destinations than short-distance shipments.

In *NAADSM 3*, however, there was no association between distance and delay: these parameters operated independently and in an unrealistic way. Additionally, because of the way the *NAADSM* is structured, there is already an implicit one-day delay: the effect of exposures occurs on the simulation day after the exposure takes place. Further complications in *NAADSM* 3 arose when control measures that would impact the outcome of a contact (*e.g.*, quarantine or destruction) took place while such contacts between units were pending. Behavior of the model under these conditions was not well defined.

Readers of earlier editions of this guide (Hill and Reeves, 2006) will recall that we consistently advised users not to use these parameters. In *NAADSM* 4, they have been eliminated altogether.

3.6. Clarification of disease spread by direct and indirect contact

In *NAADSM* 3, users could choose whether latent and subclinical units could spread disease by direct contact, and whether subclinical units could spread disease by indirect contact. In a previous edition of this guide (Reeves *et al.*, 2012), we pointed out that, based on definitions of direct and indirect contact, latent and subclinical units should always be able to spread disease by direct contact. Similarly, subclinical units should always be able to spread disease by indirect contact. In *NAADSM* 4, these two situations are now the case, and the user options described above have been eliminated.

3.7. Queuing herds to be vaccinated

Due to a quirk in the model specification for *NAADSM* 3 (*NAADSM* Development Team, 2010a, 2010b), it was possible for herds in queue to be vaccinated (Section 9.14) to be listed in the queue multiple times. This behavior has been changed in *NAADSM* 4. This change not only more accurately represents expert opinion with respect to how vaccination is actually carried out, but it also makes it possible to provide useful information to model users about how limited vaccination capacity affects disease control efforts. Readers are referred to *NAADSM* technical paper #4, *Understanding vaccination queues in* NAADSM *3 and* NAADSM *4* (Reeves, 2011), available at <*http://www.naadsm.org/techpapers*>, for more information.

4. 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 4-1, 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 4-1, 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).

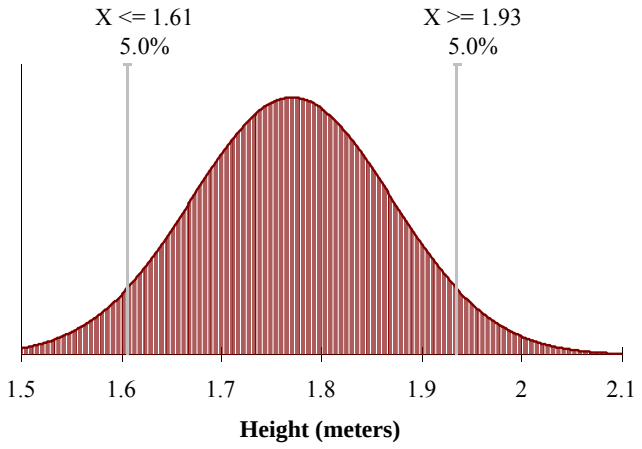


Figure 4-1. Distribution of height among males in the United States.

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

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 4-2). 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 be affected by waist size and pants style).

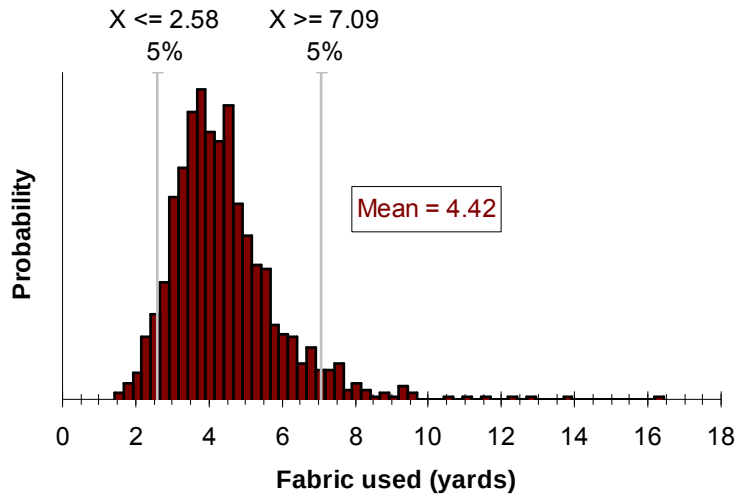


Figure 4-2. Output of 1000 iterations of a stochastic model estimating the amount of fabric needed to make a pair of pants.

NAADSM is a stochastic model. Variability is incorporated in the model using *probabilities*, *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 4-3). 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).

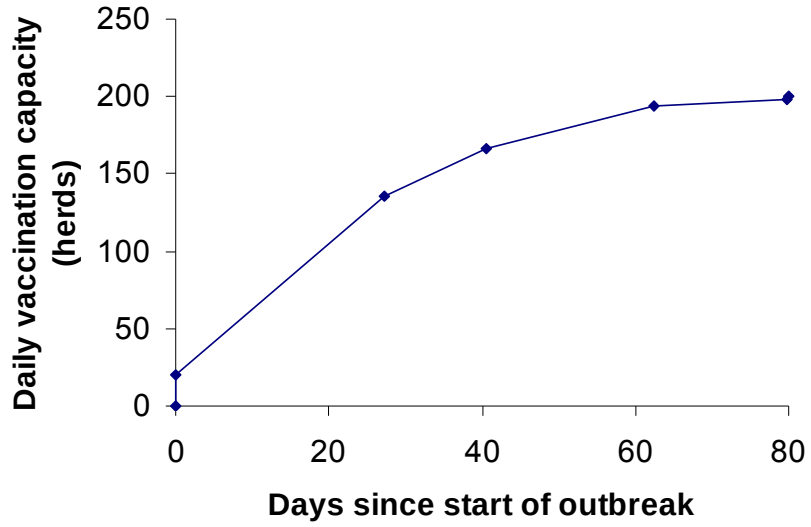


Figure 4-3. Number of herds that can be vaccinated per day as a function of time since the start of an outbreak.

5. Installing and registering *NAADSM*

5.1. Installing *NAADSM*

Installation of *NAADSM* is straight-forward. Simply download the most recent installation package from the *NAADSM* website, <*http://www.naadsm.org*>. Follow the relevant links to download the *NAADSM* application and select the latest version of *NAADSM/PC*. The installation package will have a file name similar to **`NAADSMSetup-4_0_x.exe`**, where the more recent full installation version number is given in place of "x". Save this installation package to your computer.

Next, run this setup package, *e.g.*, by double-clicking on its icon. A screen like the one shown in Figure 5-1should be displayed.

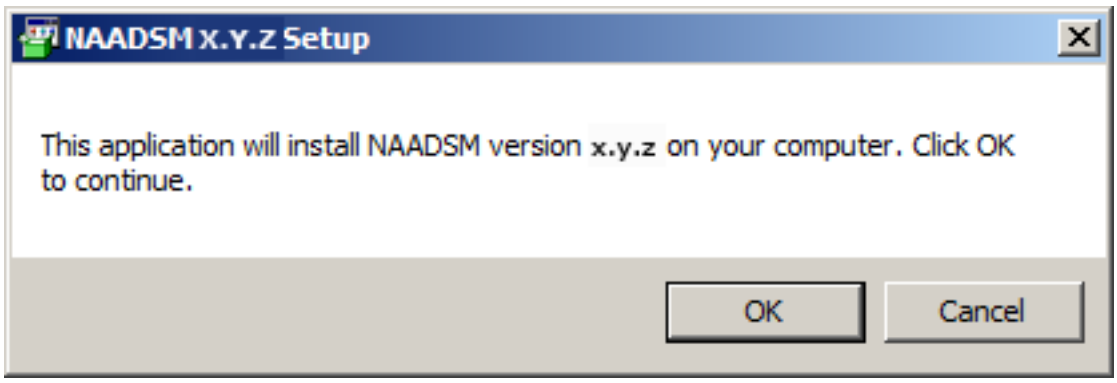


Figure 5-1. The first installation message displayed by the *NAADSM* setup package.

Click **OK** to begin the installation process. On the following screen, click **I Agree** to accept the license agreement. By default, the *NAADSM* application and related library files will be installed to the folder **`C:\Program Files\NAADSM 4.0.x`** on your computer (again, where "x" will be the number of the most recent release) unless you choose a different location. You will have the choice of creating shortcuts for *NAADSM* on your *Windows* **Start** menu, on your *Windows* desktop, or both. Follow the prompts as they are displayed on your screen. If installation proceeds properly, the message shown in Figure 5-2 will be shown. (If an installation error occurs, please consult with one of the *NAADSM* developers, as listed on the *NAADSM* website at <*http://www.naadsm.org/contacts*>).

Box 5-1. Permissions required for NAADSM *installation*

In order to install *NAADSM* to the default location, users must have "write" permission for the **`Program Files`** folder. Some organizations do not allow users sufficient privileges to install software themselves. If your user account does not have sufficient privileges, installation may fail with an error message that reads in part:

> Error opening file for writing. Click Abort to stop the installation, Retry to try again, or Ignore to skip this file.

If you see this message, you will need to re-run the installation program and select a different folder for installation, or check with your computer support personnel to acquire sufficient permission to install *NAADSM* on your computer.

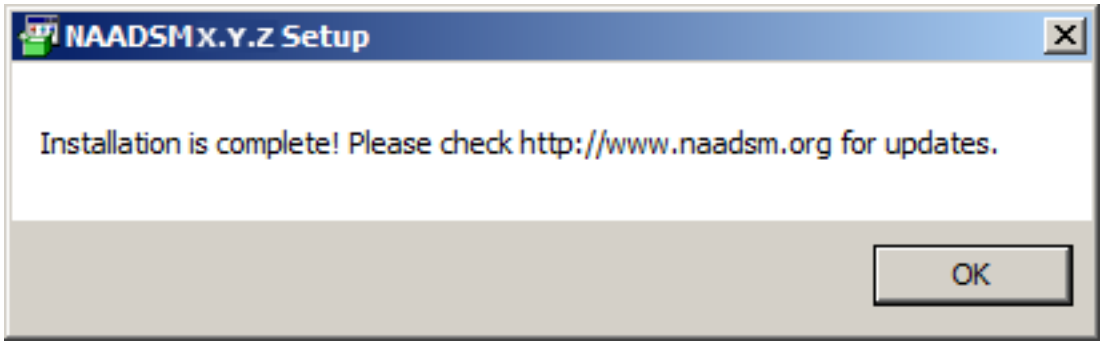


Figure 5-2. The final installation message displayed by the *NAADSM* setup package.

When installation is complete, click on your newly created shortcut to *NAADSM* to launch the application.

5.2. Registering *NAADSM*

While registration is not required, users are encouraged to register their copies of *NAADSM* through the *NAADSM* website at <*http://www.naadsm.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*.

6. Running your first simulation

NAADSM is supplied with a sample scenario, which includes artificial herd population data and parameters for a fictitious disease. To view and run this sample scenario, start the *NAADSM* application, go to the **File** menu, choose **New scenario…**, select **Sample scenario**, and select a folder in which to save the new scenario when prompted.

> *Box 6-1.* ***Warning****: The* NAADSM *sample scenario*
>
> The sample scenario file included with *NAADSM* is exactly that: it is only a sample. The parameters included in the sample scenario are not representative of any actual population, disease, or situation. A quick examination of the sample dataset will show that this livestock population exists somewhere in the North Atlantic Ocean: hardly a situation to be taken seriously!

6.1. 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 6-1). Each dot represents the location of a particular herd. 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 in the center of 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 9.4.

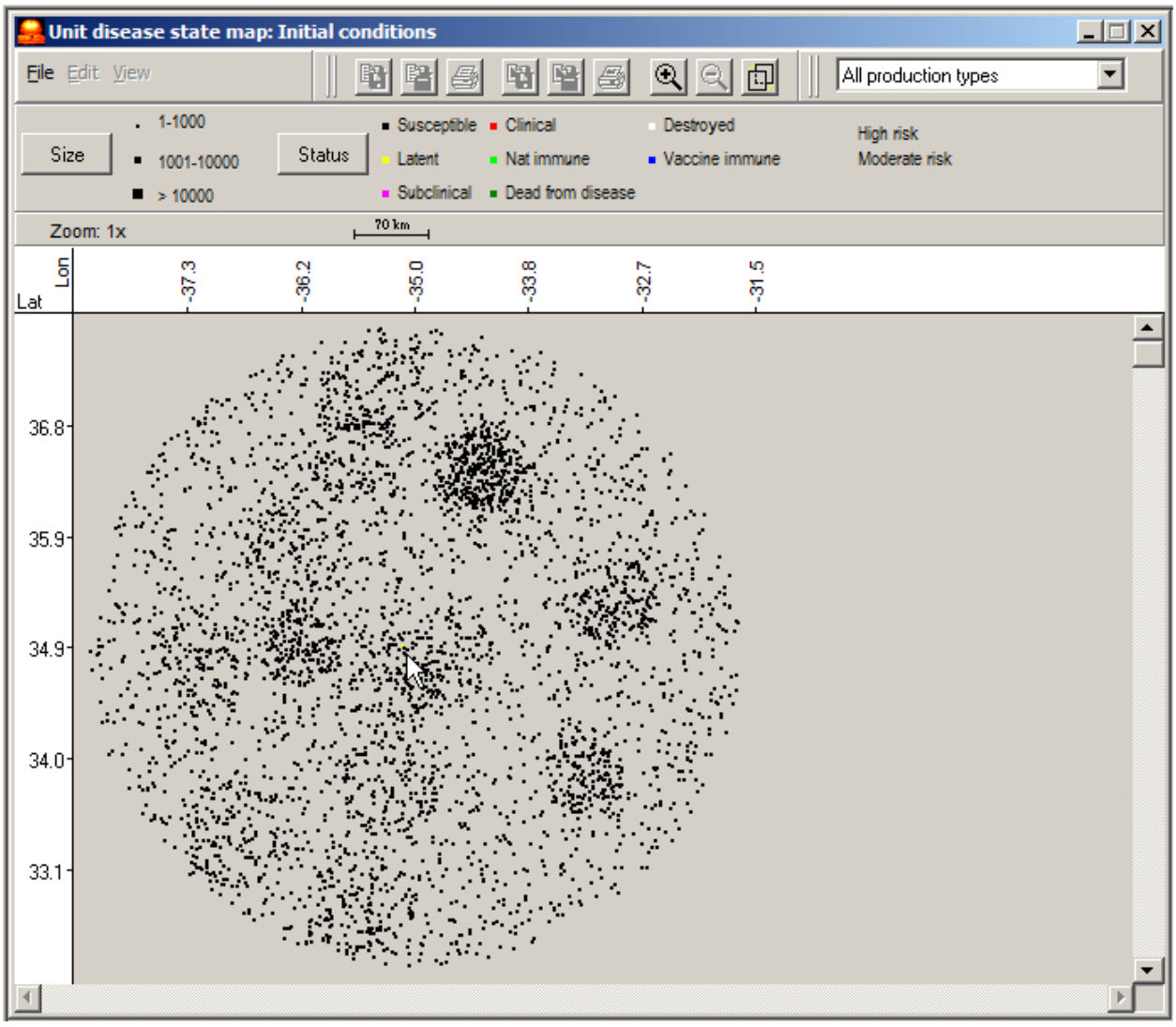


Figure 6-1. Initial map of herds in the sample scenario.

The cursor points to the latently infected herd (shown in yellow). All other herds (shown in black) are susceptible at the start of this simulation.

6.2. 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 6-2). 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 Chapter 9.

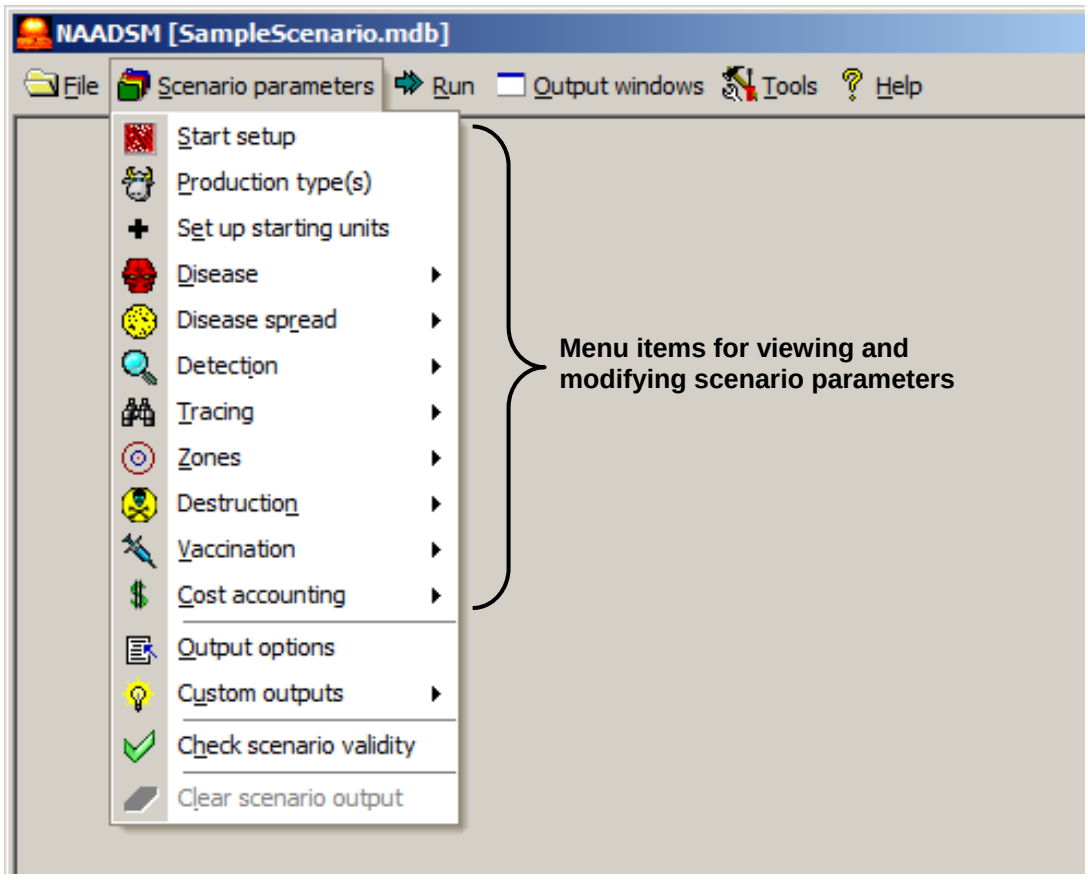


Figure 6-2. Menu items for viewing scenario parameters.

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.

6.3. 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 6-3). 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 Chapter 10.

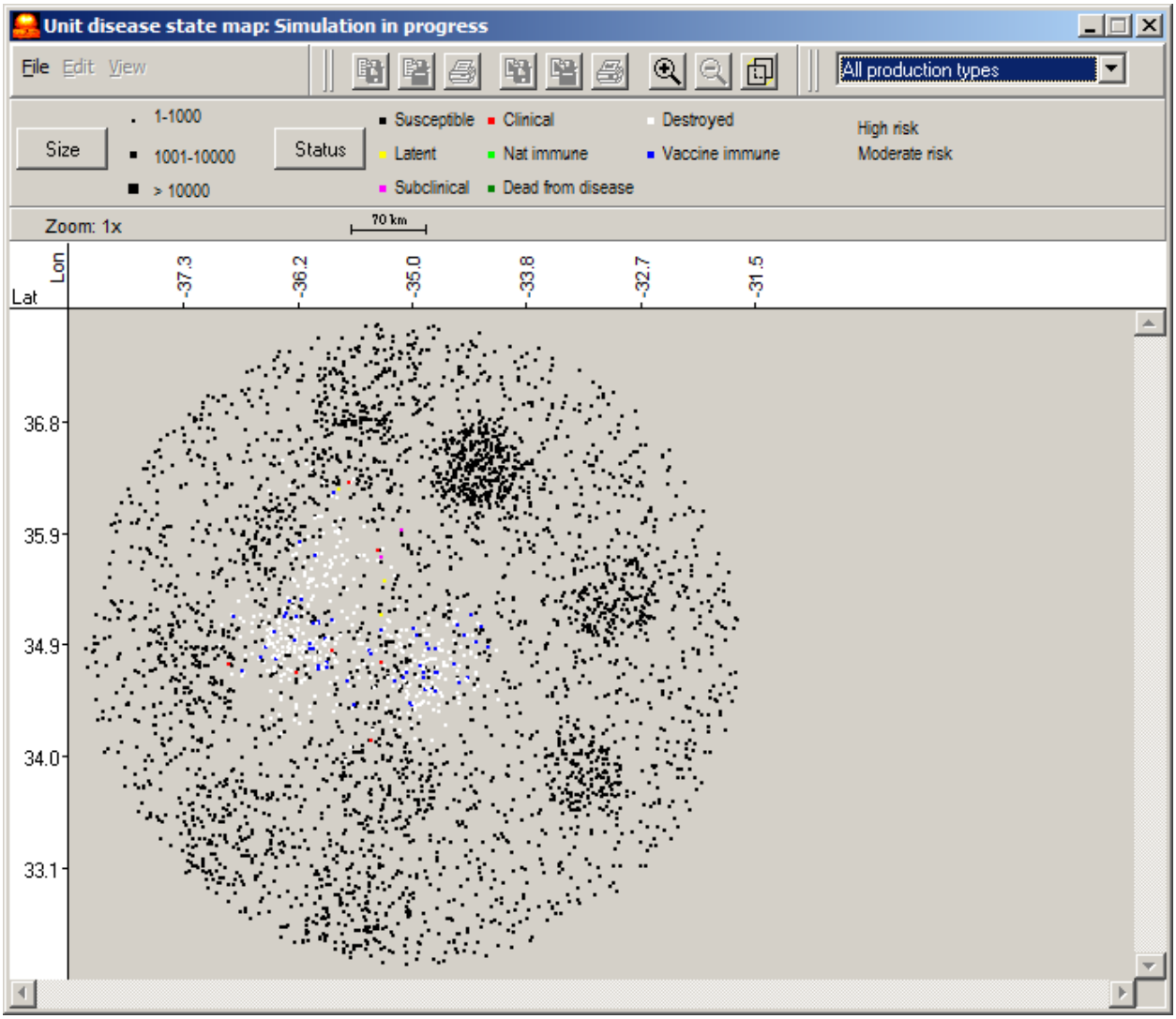


Figure 6-3. Map showing herds and their disease states for an iteration in progress.

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; light green: herds that have progressed to naturally immune after disease; dark green: herds that have progressed to the dead from disease state after disease; blue: herds that have become immune to disease as a result of vaccination; white: herds that have been destroyed as a control measure in the simulation.

6.4. 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**. The **Tabular view** tab in this

window shows the number of herds that were infected, the number of herds that were detected, the number of animals in infected and detected herds, and the numbers of herds and animals destroyed or vaccinated. Additionally, the **Graphical view** tab (Figure 6-4) shows two epidemic curves: the actual epidemic curve displays the number of units infected on each day of a simulation, while the "apparent" epidemic curve shows the number of infected units that are detected on each day. Note that the apparent epidemic curve underestimates the total number of infected units and lags behind the actual epidemic curve. This is a result of imperfect detection of disease (discussed further in Sections 9.8 and 11.1).

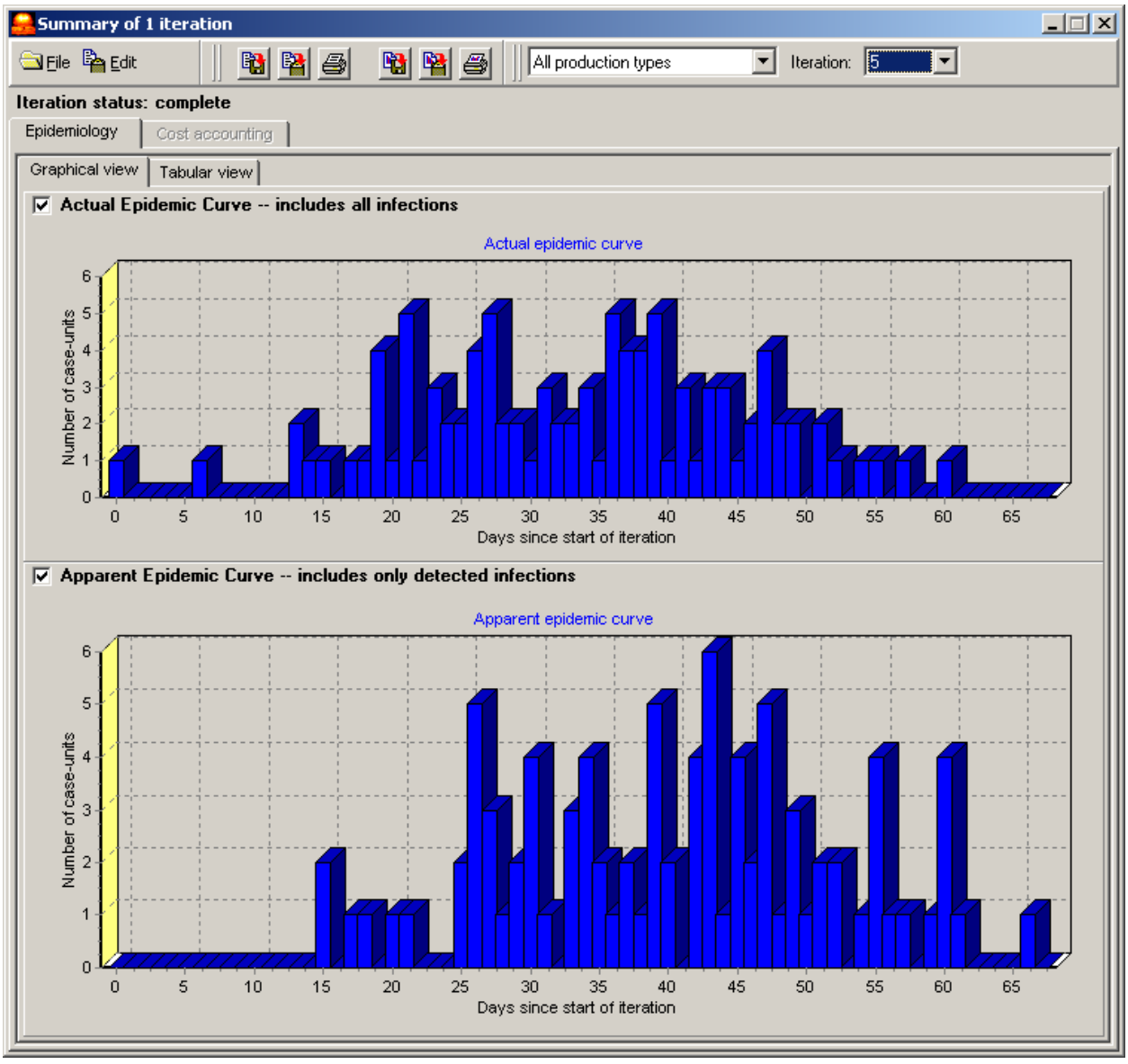


Figure 6-4. Epidemic curves showing the number of infected ("actual") and detected ("apparent") herds on each day of a single simulated outbreak.

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 6-5 will be displayed. Note in Figure 6-5 how the number of susceptible herds 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.

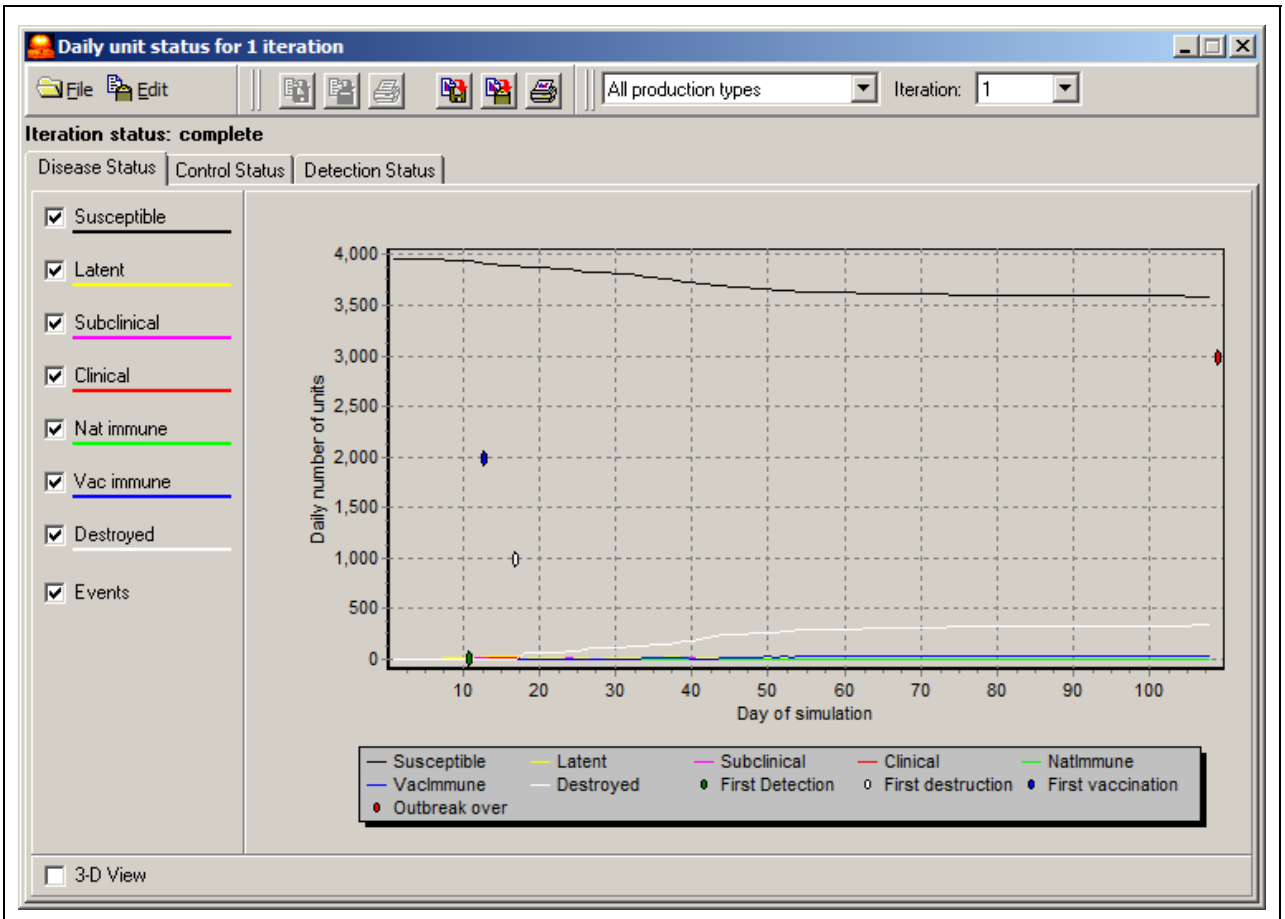


Figure 6-5. Chart of the number of herds in the various disease stages over the course of an iteration.

Several other selections are available on the **Output windows** menu. These items, which will be discussed in Chapter 11, 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 a *NAADSM* scenario, and will show how these parameters may be set by the user.

7. 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 a *NAADSM* scenario. All input parameters and their uses are described in detail in Chapter 9.

7.1. 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 7.2.

7.1.1. 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.

7.1.2. 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.

7.1.3. 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.

7.1.4. 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.

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

7.1.5. Probability density functions

Probability density functions (*pdf*s) 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. Chapter 4 provides additional background information regarding the use of distributions in stochastic models.

7.1.5.1. Types of *pdfs*

Probability density functions generally fall into two broad classes: *pdfs* based on theoretical distributions are specified by defined parameters, and have mathematical formulas that determine the shape of the *pdf* based on the values of those parameters. A Gaussian (or normal) probability density function is a good example: this function is defined by two parameters, a mean and a standard deviation. The shape and location of the curve are determined by the values of these parameters (Figure 7-1). Other common *pdfs* based on theoretical distributions are triangular, uniform, lognormal, exponential, and beta distributions. Appendix A lists the *pdf* types supported by *NAADSM*.

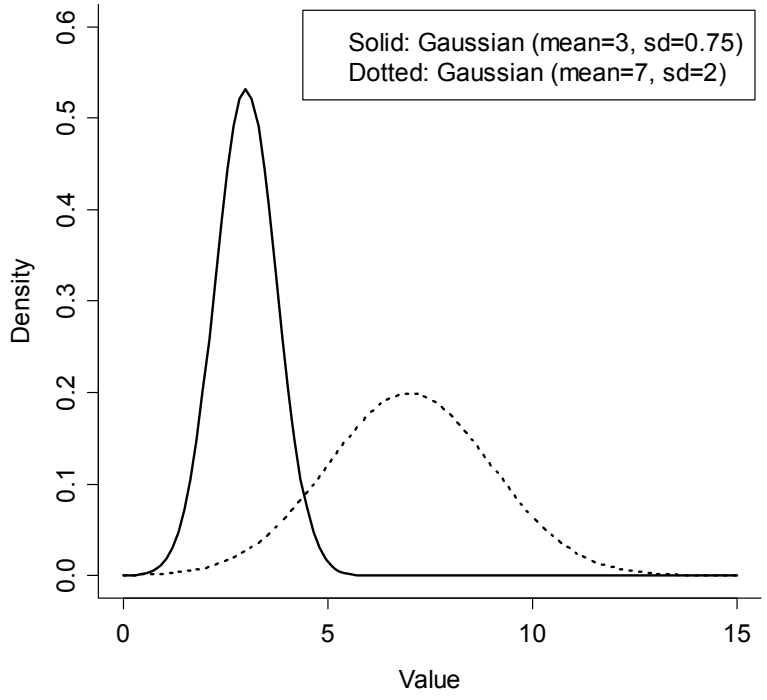


Figure 7-1. Two Gaussian (normal) distributions with different parameters.

Note that the location (center) and shape of each curve are determined by the mean and the standard deviation, respectively.

Theoretical distributions are often fit to observed data, as shown in Figure 7-2. This guide does not cover the topic of distribution fitting, but good discussions are presented elsewhere (Law, 2006; Vose, 2000).

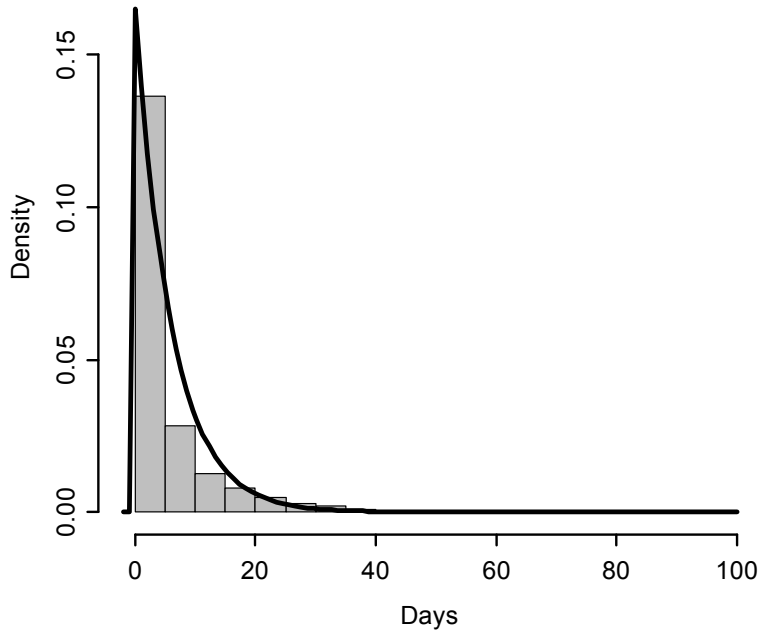


Figure 7-2. Fitting a *pdf* based on a theoretical distribution to data.

Sample data for an estimated duration of a unit-level clinical disease state (Section 9.5) is plotted in the histogram. Specialized software was used to determine the best fitting theoretical distribution (indicated by the black line). In this case, the best fitting theoretical distribution was found to be an exponential distribution with a mean of just over 6 days.

Other probability density functions are based not on defined theoretical distributions, but can be based directly on data or some arbitrarily defined shape. The shapes of these *pdfs* are not determined by mathematical formulas, and can be quite variable. These *pdfs*, referred to as empirical *pdfs*, are also supported by *NAADSM*. Two types of empirical *pdfs* are piecewise (or general) *pdfs* (Section 7.1.5.3), and histogram *pdfs* (Section 7.1.5.4).

New *pdfs* may be entered in *NAADSM* by providing parameters for known theoretical distributions, by importing data, or by using the provided tools to create your own shapes, as described in the following sections.

7.1.5.2. Creating and editing *pdfs* based on theoretical distributions

7.1.5.2.1. The *pdf* selection box

Each *pdf* parameter in *NAADSM* is associated with a selection box like the one shown in Figure 7-3. The selection box shows a list of *pdf*s defined in each scenario, and provides buttons for creating, editing, and removing *pdf*s.

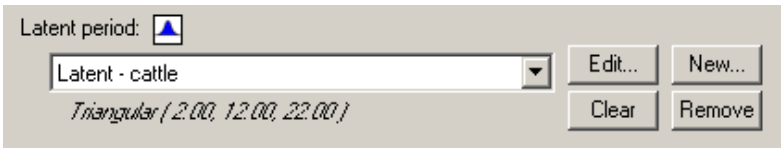


Figure 7-3. Selection box for probability density functions.

Use the drop-down box to select a *pdf* that has already been defined. Click **Edit…** to alter an existing *pdf*, or **New…** to define a new *pdf*. Note that a description of the selected *pdf* is displayed below the drop-down box.

7.1.5.2.2. 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 7-4. The left side of the window shows the type of the selected *pdf* (*e.g.*, triangular, normal, uniform, *etc.*), the parameters required to define the selected type, and the values of those parameters. Simply enter values for all of the required parameters in order to specify a *pdf*. The right side of the window graphically depicts the selected *pdf*. Changing the parameters will cause the graphical depiction to change as well (Figure 7-5). If the parameters are changed to inappropriate values, an error message is displayed (Figure 7-6). The user must address the error problem in order to save the changes to the newly altered or created *pdf*.

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 7-4). 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: use the **Save** command in the lower right corner of the *pdf* editor window, or click on the  button in the toolbar.

If you wish to exit the *pdf* editor window without applying the changes you have made, use the **Cancel** button in the lower right corner of the *pdf* editor window, or click on the  button in the toolbar.

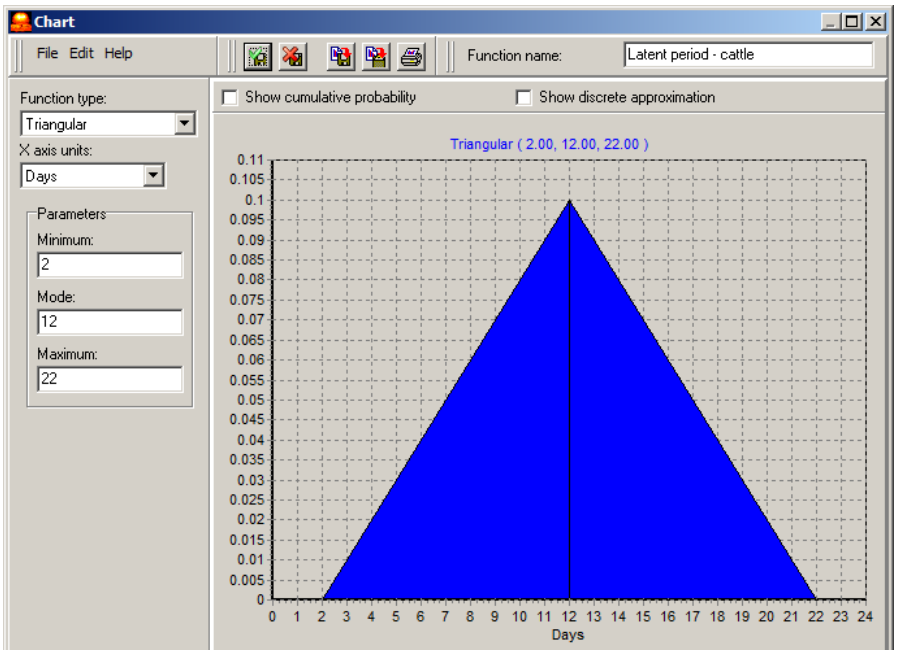


Figure 7-4. The *pdf* editor window.

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.

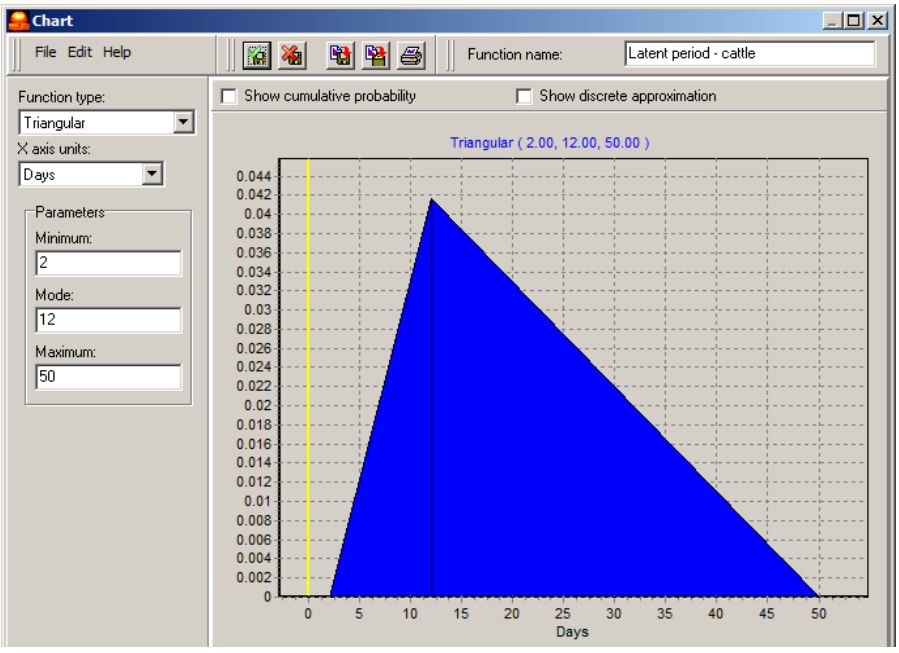


Figure 7-5. Changing the parameters of a triangular *pdf*.

The graphical preview has been updated to correspond to the new parameters.

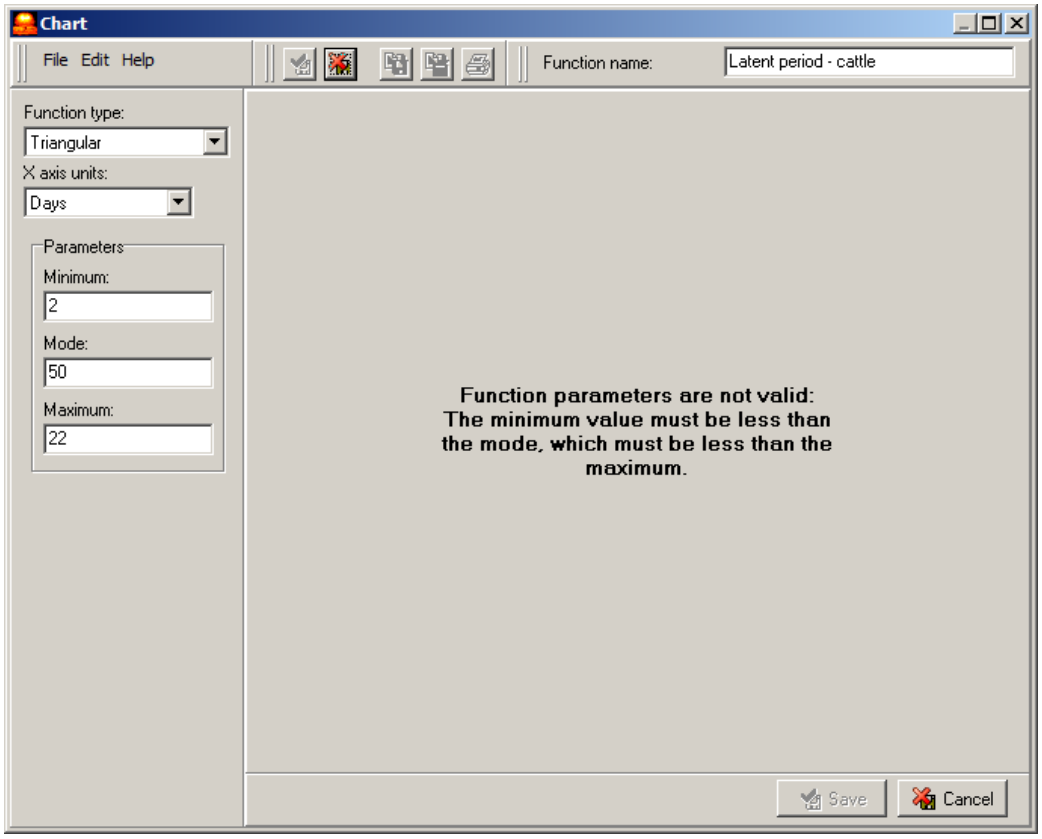


Figure 7-6. Message displayed in the *pdf* editor when input parameters are not valid.

When there is an error in the specified parameters the source of the error is shown instead of the graphical preview.

7.1.5.3. Working with piecewise *pdf*s

Piecewise *pdf*s can reflect nearly any shape that the user wishes. Piecewise distributions are assembled from a series of (*x*,*y*) points as shown in Figure 7-7. Specifying a few points will produce a rather coarse piecewise *pdf*, while specifying many points can produce a piecewise *pdf* that is nearly smooth and can closely approximate a *pdf* based on a theoretical distribution. The only restrictions for these points are that each *x* value must be greater than the previous *x* value, and that the start and end points have *y* values of 0.

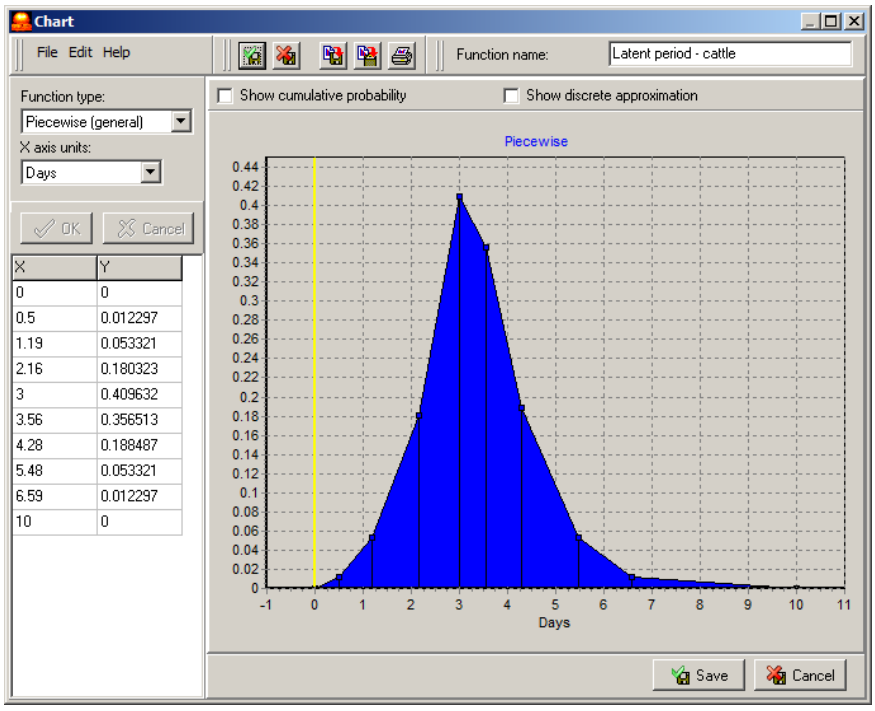


Figure 7-7. A piecewise *pdf*.

The x and y values of each individual point are shown in the left panel. A graphical preview is shown on the right.

7.1.5.3.1. Creating and editing piecewise *pdfs*

To create a piecewise *pdf*, select **Piecewise (general)** from the list of function types. Three points are created automatically (Figure 7-8). 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.

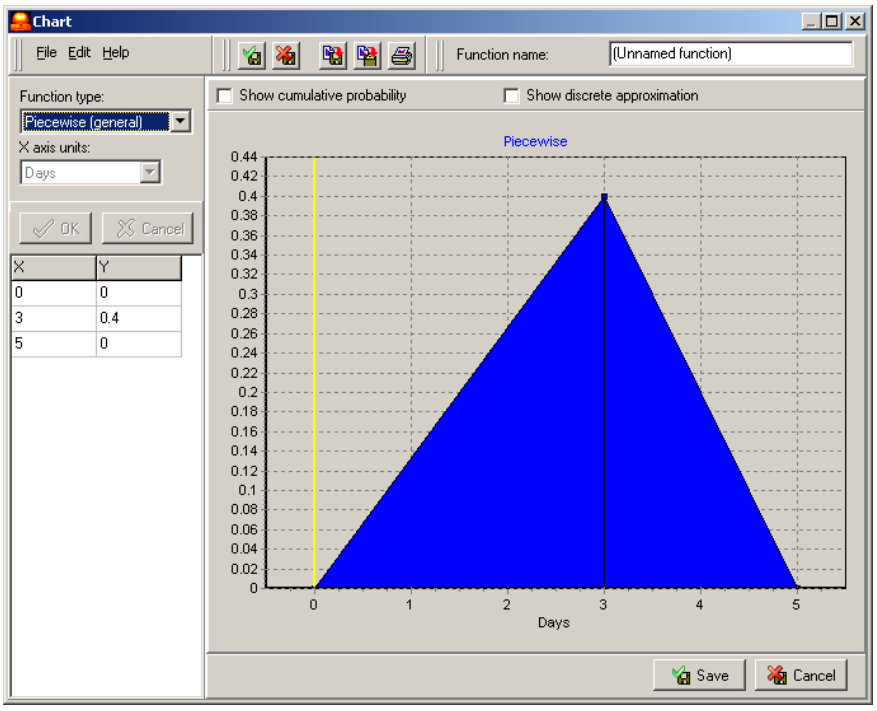


Figure 7-8. A new piecewise *pdf*.

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 7-9).

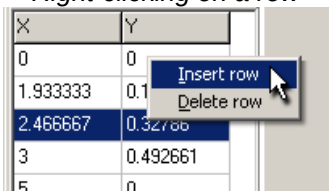
Alternatively, right-click on the table on the left side of the *pdf* editor window, and select **Insert row**. A new row will be added to the table, and you can edit the *x* and *y* values to set the desired location of the new point.

Box 7-1. Right-clicking in the table of points

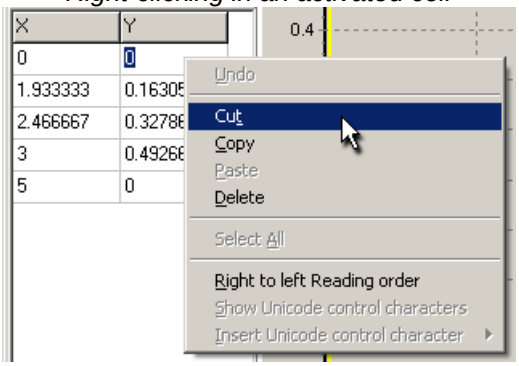
Depending on exactly where you are in the points table when you right-click, you may see one of two possible context menus. If you select a row, the entire row will be highlighted, and you will see the context menu shown on the left below. This menu will let you insert or remove rows.

If a cell in the table is activated for editing (*i.e.*, only a single cell is highlighted), then a different context menu is shown, which will let you copy and paste cell values (below right). It may take a little practice to determine whether you are selecting a row or a cell.

Right-clicking on a row



Right-clicking in an activated cell



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 7-10).

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**.

Alternatively, right-click on the appropriate row in the on the table on the left side of the *pdf* editor window, and select **Delete row**.

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.

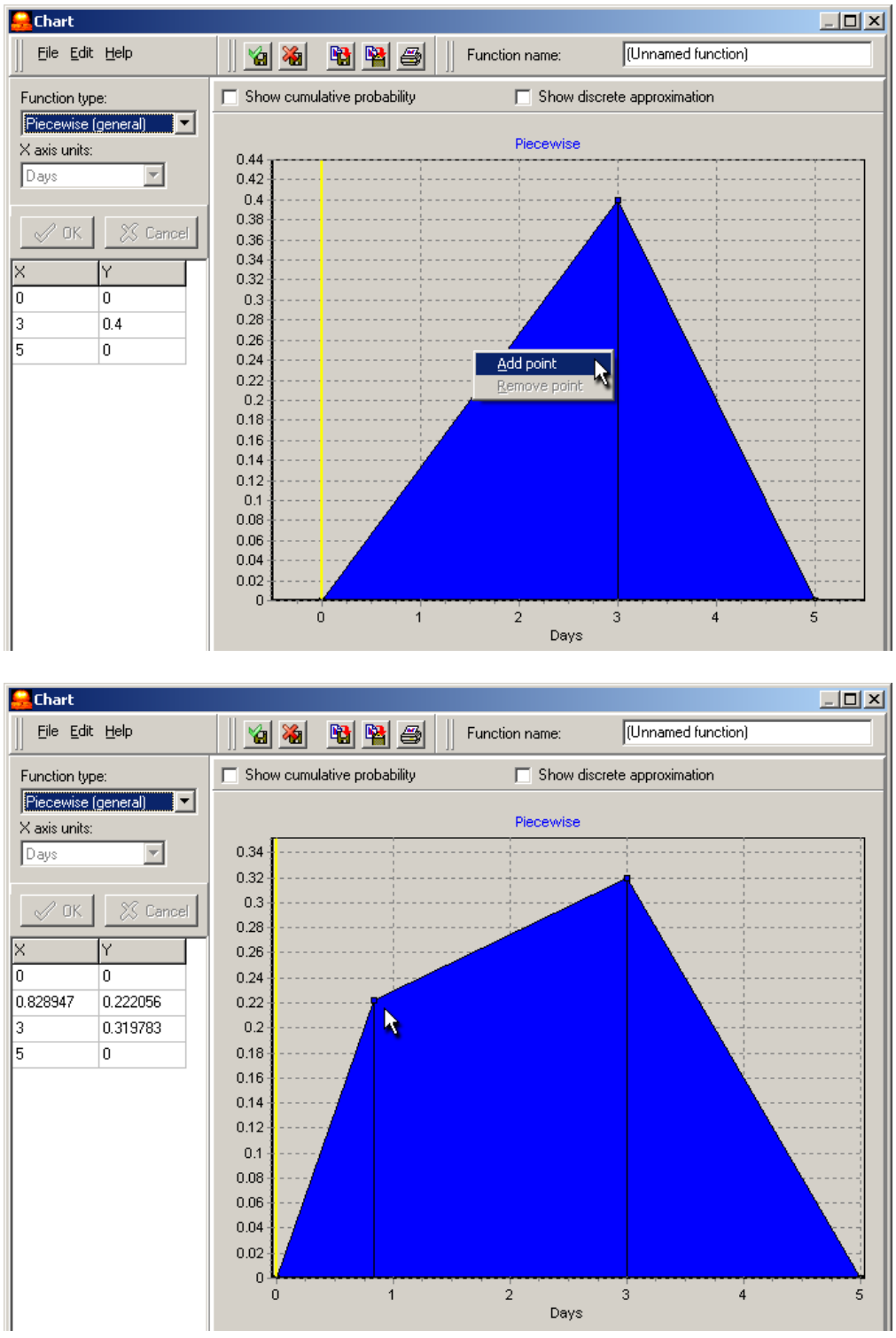


Figure 7-9. Adding a new point to a piecewise *pdf*.

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.

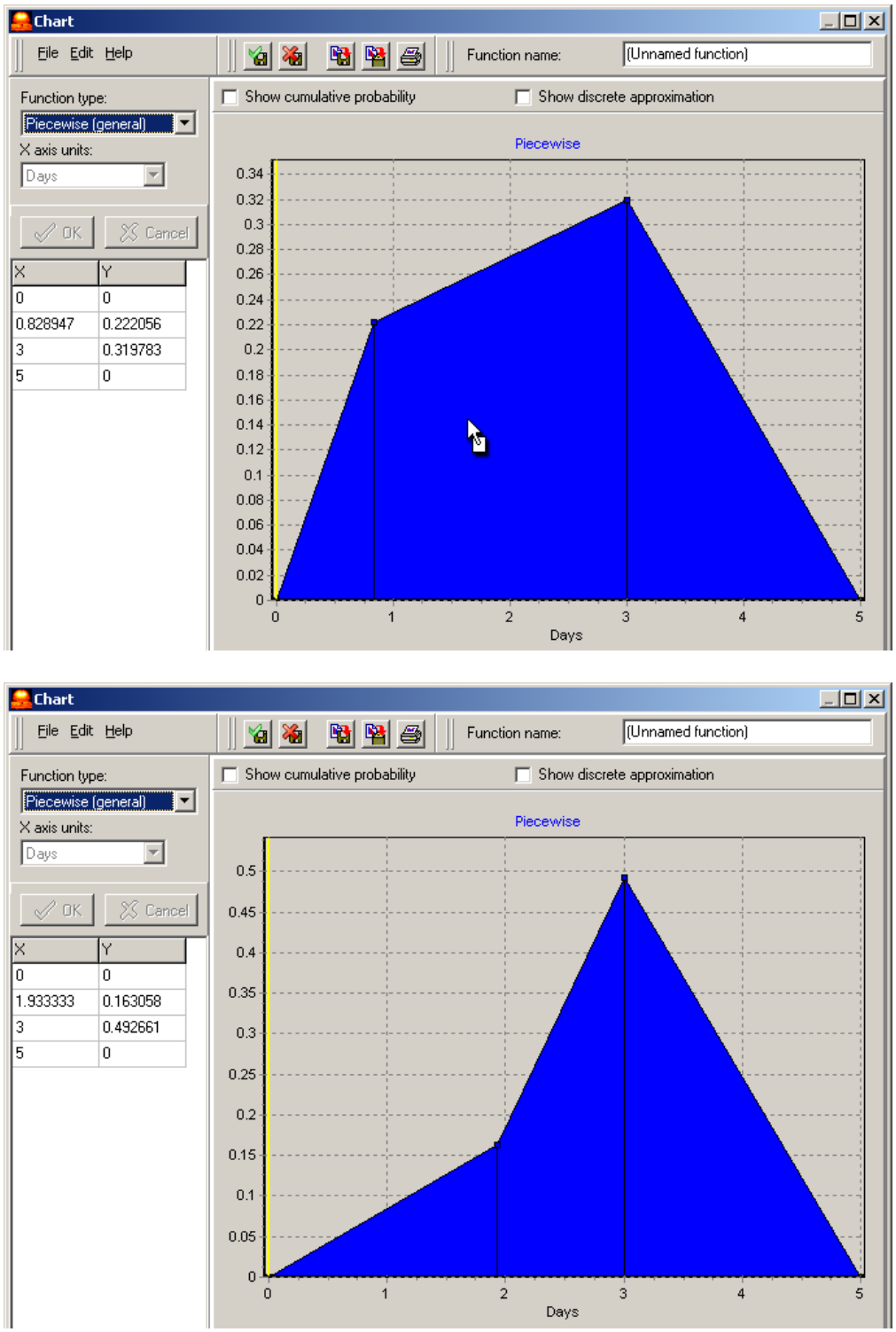


Figure 7-10. Moving an existing point of a piecewise *pdf*.

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.

7.1.5.3.2. 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 7-11 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 B.

To import points from a file, use the menu command **File → Import from file… → Import piecewise pdf** in the *pdf* editor window. To import points from the clipboard, use the menu command **Edit → Import from clipboard → Import piecewise pdf**. 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 **Save** button in the lower right corner of the pdf editor window or use the  button on the tool bar across the top of the screen.

```
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 7-11. 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 for the file to be imported.)

7.1.5.3.3. 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.

7.1.5.4. Working with histogram *pdf*s

A histogram *pdf* is a second type of commonly used empirical *pdf*. As the name implies, a histogram of the data values of interest is used to define a probability density function. The left panel of Figure 7-12 shows a simple histogram generated from a hypothetical dataset of 35 values. Each histogram bin is defined by its starting and ending cut-off points, and the number of values that fall between those cut-off points. The histogram shows that two data points were between the cut-off values of 0 and 1 (*i.e.,* in the first histogram bin), three data points were between the cut-off 1 and 2 (in the second bin), six data points were between the cut-off values 2 and 3 (the third histogram bin), and so on.

The right panel of Figure 7-12 shows the same histogram rescaled to give a probability density function. Recall that the area under a *pdf* is always equal to 1. The densities shown on the right are simply calculated by dividing the count for each histogram bin by the number of data points represented. The first bin then represents 2/35 = 5.71% of the total area, the second bin represents 3/35 = 8.57% of the total area, the third bin represents 6/35 = 17.14% of the total area, and so on, to give a total of 100%. When values are generated from this distribution, there is a 5.71% chance that the value will be between 0 and 1, an 8.57% chance that the value will be between 1 and 2, and so on.

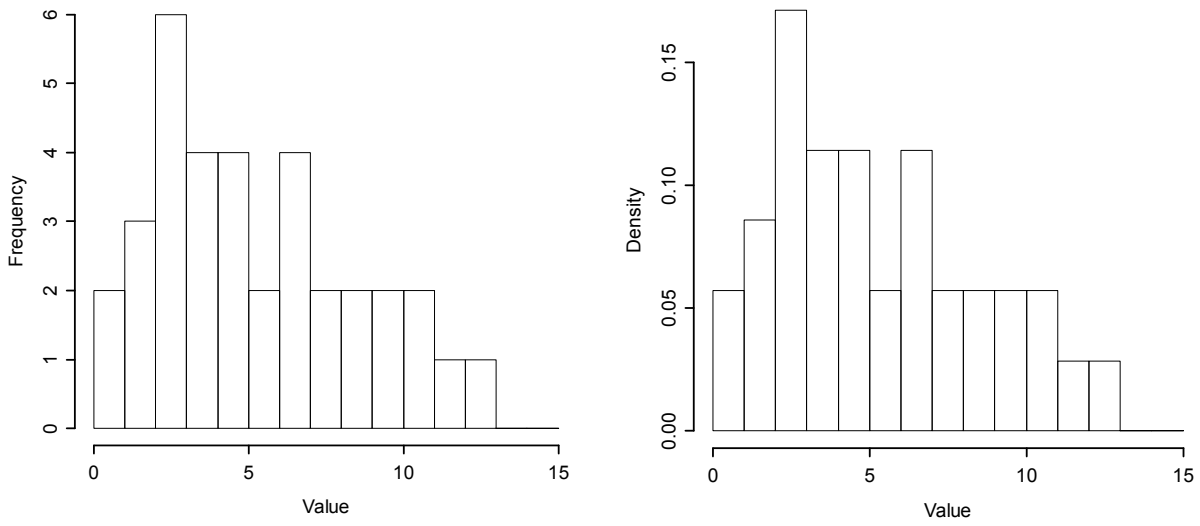
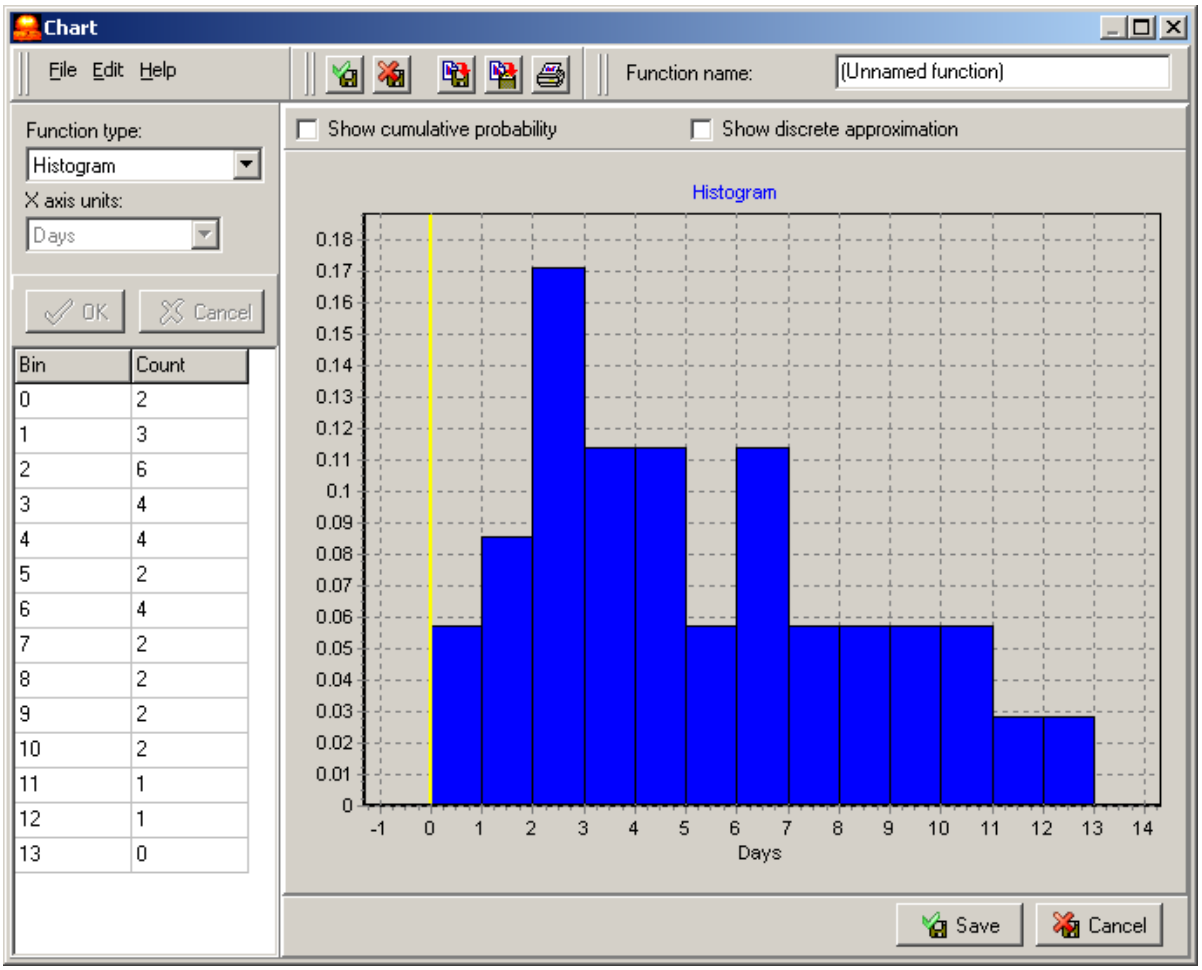


Figure 7-12. A histogram and a histogram *pdf*.

The *pdf* is generated by rescaling the frequencies (the counts of data points in each bin) to densities.

Figure 7-13 shows this histogram *pdf* in the *NAADSM pdf* editor window. Note the values in the table on the left side of the window. The first bin has its starting cut-off value at 0. The ending cut-off for the first bin is the starting cut-off for the second bin, or 1 in this case. The count of data points in first bin is 2. Every other bin is defined in the same way.



Bin	Count
0	2
1	3
2	6
3	4
4	4
5	2
6	4
7	2
8	2
9	2
10	2
11	1
12	1
13	0

Figure 7-13. A histogram *pdf* in *NAADSM*.

Bin ranges (starting and ending cut-off points) and counts are specified as described in the text.

Also note the last row in the table, which has a starting cut-off point of 13. No ending cut-off point is specified, but it is not really needed: the count for this last bin is 0, *i.e.*, there are no data points with values greater than 13. *NAADSM* always assumes that the last value in the **Bin** column is the last cut-off point (the maximum possible value), and the last value in the **Count** column is always assumed to be 0 (*i.e.*, there can be no values higher than the maximum possible value).

From the information shown in the table in Figure 7-13, which includes the starting and ending cut-off points for each histogram bin and the count of values in each histogram bin, *NAADSM* automatically generates a histogram *pdf*.

7.1.5.4.1. Creating and editing histogram *pdfs*

To create a histogram *pdf*, select **Histogram** from the list of function types. A histogram with two bins is created by default. You can alter the number of bins by right-clicking on the

table, and selecting either **Insert row** or **Delete row** from the context menu, similar to that shown in Section 7.1.5.3.1. Edit the bin ranges (by changing values in the **Bin** column) and counts by changing values in the **Count** column), and click **OK** to view the *pdf*. If you are editing an existing histogram *pdf*, **Histogram** will already be selected, and the currently defined bin ranges and counts will be displayed.

7.1.5.4.2. Importing a histogram *pdf*

If you have defined bin ranges and counts for your histograms *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, labeled *x* (for the bin cut-off values) and *y* (for bin counts), 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). Counts (y values) may be in any appropriate scale, for example, whole-number counts or proportions. *NAADSM* will calculate the histogram *pdf* so that the area under the curve equals 1. Data can be imported from a file or from the clipboard. Figure 7-14 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* are given in Appendix B.

```
x,   y                    x,   y
0,   2                    0,   0.117647059
1,   3                    1,   0.176470588
2,   6                    2,   0.352941176
3,   4                    3,   0.235294118
4,   4                    4,   0.235294118
5,   2                    5,   0.117647059
6,   4                    6,   0.235294118
7,   2                    7,   0.117647059
8,   2                    8,   0.117647059
9,   2                    9,   0.117647059
10,  2                    10,  0.117647059
11,  1                    11,  0.058823529
12,  1                    12,  0.058823529
13,  0                    13,  0
```

Figure 7-14. Two properly formatted **`*.csv`** files for importing histogram *pdfs*.

These two sets of values will produce identical histogram *pdfs*: although the *y* values differ, each set has the same relative scale. When *NAADSM* calculates the histogram pdf, this scale is automatically accounted for. Note that the first row is always 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 for the file to be imported.)

7.1.5.4.3. Exporting a histogram *pdf*

You may want to export the bin ranges and proportions of a histogram *pdf* for use in a spreadsheet or statistical package. 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.

7.1.5.5. Copying and pasting *pdfs* between applications

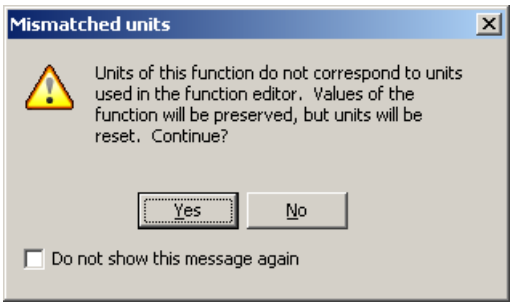
If you are creating or modifying several *NAADSM* parameters, you may find yourself re-entering the same values to create identical *pdfs* in several places. This manual process is not only time-consuming, but error-prone. It is more convenient to copy and paste *pdfs* from one place to another.

To copy a *pdf* for use elsewhere, use the **Edit → Copy** menu command from the *pdf* editor window. To paste a *pdf* from the clipboard into the *pdf* editor window, use the **Edit → Paste** menu command.

Box 7-2. ***Warning****: Units of x axis values of* pdfs

NAADSM uses *pdfs* to represent many parameters, which often have different units. For example, disease state durations are given in units of days, while distances over which contacts occur are given in units of kilometers.

If you attempt to copy a *pdf* from a parameter that uses one unit and paste it to a parameter that uses a different unit, you will see a warning message like the one shown below:



The values of the *pdf* on the clipboard will not be changed, but the units will be changed to those expected by the parameter to which you are pasting the *pdf*. There is nothing necessarily wrong with this, as long as you know that the units are appropriate.

It is the user's responsibility to ensure that the values are really provided in the proper units. *NAADSM* does not, for example, automatically convert from miles to kilometers, and cannot possibly convert from kilometers to days.

Box 7-3. Using pdfs *across multiple applications*

Pdfs can be copied and pasted not only from one parameter to another within *NAADSM*, but can also be copied and pasted between different applications. If you have several instances of *NAADSM* running at the same time, you can copy and paste *pdfs* from one scenario to another.

You can also copy and paste *pdfs* between different programs that support *NAADSM*-style *pdf* functions. The *NAADSM* Development Team has created and released several other models, utilities, and other related applications that are compatible with *NAADSM*. Several of these programs can be found on the *NAADSM* website at <*http://www.naadsm.org/otherapps*>.

Box 7-4. For advanced users: Manipulating pdf *parameters in plain text*

Pdfs are copied to the *Microsoft Windows* clipboard in plain-text format. You can see this for yourself if you copy a *pdf* from *NAADSM* and paste it into a plain-text editor like the *Notepad* utility included with *Microsoft Windows*. If you *carefully* edit this plain-text representation, you can copy and paste it back into *NAADSM*. This feature may be particularly useful if you use another script or application to produce your *pdfs*: writing a script or program that produces *pdfs* for *NAADSM* can be accomplished just by following the format of the plain-text representation of *NAADSM pdfs*.

7.1.5.6. 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.

7.1.6. Relational functions

Relationships or relational functions (*rel*s) 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 Chapter 4 and Figure 4-3 for a more detailed discussion of *rel*s). In *NAADSM*, *rel*s are created, edited, and handled in ways very similar to those described for *pdf*s (see Section 7.1.5).

7.1.6.1. The *rel* selection box

Each *rel* parameter in *NAADSM* is associated with a selection box like the one shown in Figure 7-10. The selection box shows a list of *rel*s defined in each scenario, and provides buttons for editing, creating, and removing *rel*s.

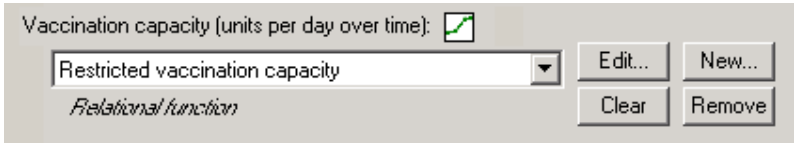


Figure 7-15. Selection box for relational functions.

Use the drop-down box to select a *rel* that has already been defined. Click **Edit…** to alter an existing *rel*, or **New…** to define a new *rel*.

7.1.6.2. 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 7-16.

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 7-16). 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 **Save** button in the lower right corner of the *rel* editor window, or click on the  button in the window's tool bar.

If you wish to exit the *rel* editor window without applying the changes you have made, use the **Cancel** button in the lower right corner of the *rel* editor window, or click on the  button in the toolbar.

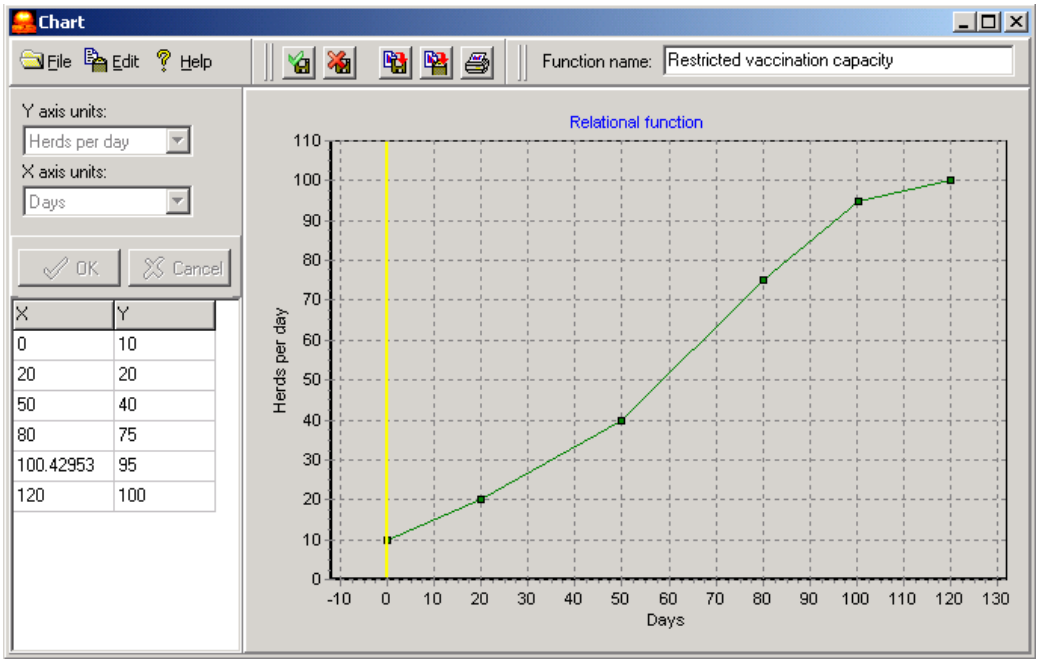


Figure 7-16. The *rel* editor window.

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.

7.1.6.3. 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 7-17). 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 *rel*s in the same way that they are added to piecewise *pdf*s (see Section 7.1.5.3.1). 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 7.1.5.3.1).

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.

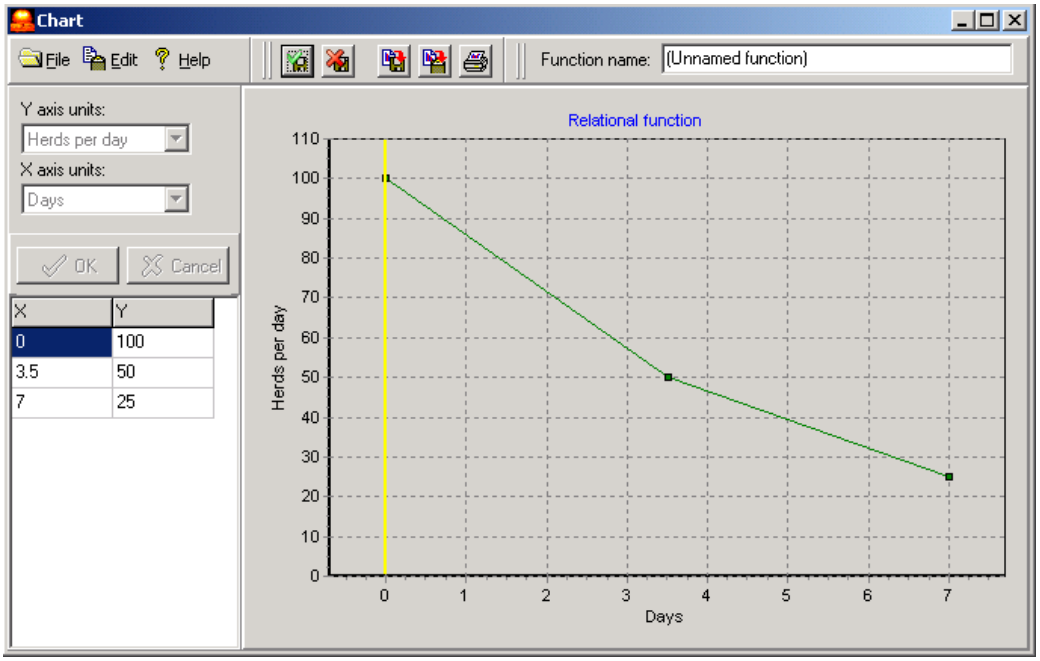


Figure 7-17. A new relational function.

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.

7.1.6.4. 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 7-18 shows sample data 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 B.

To import points from a file, use the menu command **File → Import from file… → Import relational function** in the *rel* editor window. To import points from the clipboard, use the menu command **Edit → Import from clipboard → Import relational function**. 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.

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

Figure 7-18. 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.)

7.1.6.5. Exporting a *rel* function

You may want to export the coordinates of 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.

7.1.6.6. Copying and pasting *rels* between applications

Copying and pasting rels is similar to copying and pasting pdfs (section 7.1.5.5). To copy a *rel* for use elsewhere, use the **Edit → Copy** menu command from the *rel* editor window. To paste a *rel* from the clipboard into the *rel* editor window, use the **Edit → Paste** menu command.

7.1.6.7. 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.

7.2. Complex types

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

7.2.1. Herds or units

The smallest unit in a *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 7.2.2)
- Initial disease state

As described in Section 9.5, 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.

7.2.2. 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.

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, small ruminants) 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 a *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 7.1.5)
- Duration of the vaccine immune stage (a probability density function: see Section 7.1.5)
- Detection and tracing parameters, some of which are relational functions (see Section 7.1.6)
- Control (vaccination and destruction) measures, some of which are relational functions (see Section 7.1.6)
- Direct costs associated with control measures

7.2.3. 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.

The use of production type combinations is discussed in more detail in Chapter 9.

8. 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.

> *Box 8-1.* NAADSM *scenario files and* Microsoft Access
>
> A *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://sourceforge.net/projects/jetsqlconsole/*>), 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.

8.1. New scenario file

The **File → New scenario file** command has two options: **New empty scenario file** 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, **Sample scenario file**, creates the sample scenario which we saw earlier in Chapter 6. This complete sample scenario contains an animal population, production types, and disease and control measures for a hypothetical outbreak of a highly contagious disease. In some cases, it may be easier to modify this sample scenario than to create a completely new scenario from scratch.

> *Box 8-2. A warning worth repeating: The* NAADSM *sample scenario*
>
> The sample scenario file included with *NAADSM* is exactly that: it is only a sample. The parameters included in the sample scenario are not representative of any actual population, disease, or situation. A quick examination of the sample dataset will show that this livestock population exists somewhere in the North Atlantic Ocean: hardly a situation to be taken seriously!

8.2. 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 8.6) 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 with the corrected version. Should this situation arise, please contact a member of the *NAADSM* Development Team (see the *NAADSM* website) for more information about the sources of and solutions for these problems.

> *Box 8-3. 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.

8.3. Importing and exporting scenarios

As noted in Chapter 2 and described in Appendix C, implementations of *NAADSM* exist for *Microsoft Windows*-based PCs (*NAADSM/PC*) as well as *Linux/Unix*-based parallel computing platforms (*NAADSM/SC*). These two versions use exactly the same parameters, and perform exactly the same simulations (in fact, the computer code used for simulation in the two versions is identical). Please see Appendix C for more information about *NAADSM/SC*.

Instead of using a scenario database to store input parameters, *NAADSM/SC* uses two plain-text files in extensible markup language (`*.xml`) format. One of these files contains a list of herds, and the other contains all of the other input parameters used for a scenario. The **File → Import scenario** command can be used to read one of both of these `*.xml` files and store their contents in a *NAADSM/PC* scenario database. The **File → Export scenario** command converts an existing *NAADSM/PC* scenario database file into `*.xml` files suitable for use with *NAADSM/SC*.

8.3.1. The Import scenario... command

Using the **File → Import scenario...** command will create a brand new scenario database file from existing *NAADSM/SC* `*.xml` files. The **Import scenario** command cannot be used to import parameters into an existing scenario database. This command is enabled only when no

scenario database is already open: if a scenario database is already open, close it (using **File → Close scenario file**: see section 8.6).

Figure 8-1 shows the **Import scenario** window. When importing *NAADSM/SC* `*.xml` files, you have the choice of importing a list of herds (units), a set of all other input parameters, or both. Most commonly, both will be imported, but there may be conditions in which only one or the other is required. The check boxes in the upper part of the window allow you to select the type(s) of file(s) to import.

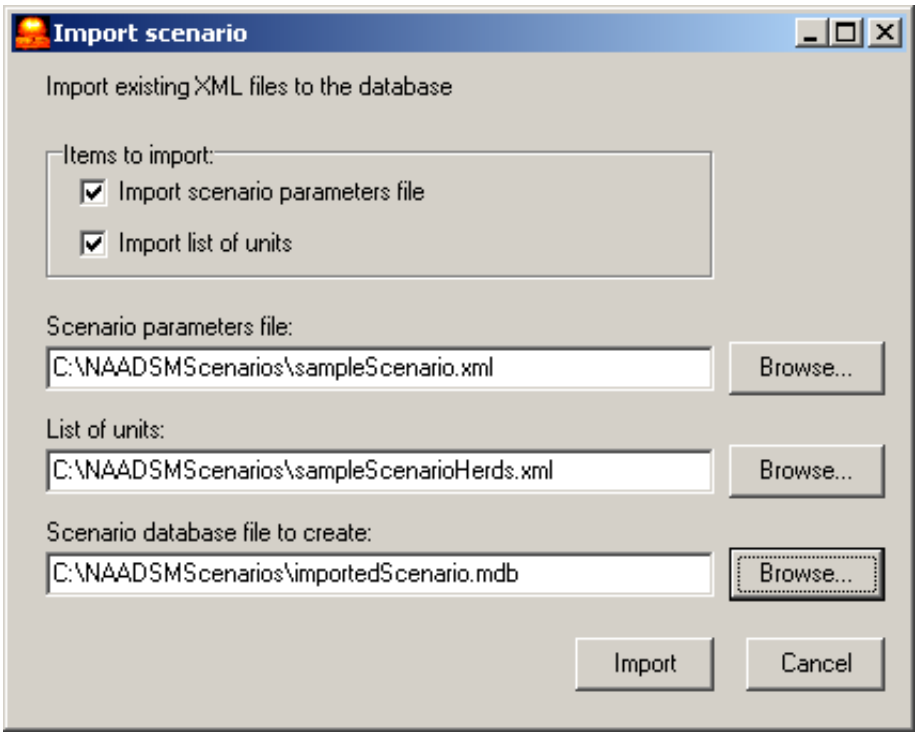


Figure 8-1. The **Import scenario** window.

The lower portion of the **Import scenario** window allows you to select the file or files that you wish to import, and also allows you to choose a name and folder location for the new scenario database to be created. Use the appropriate **Browse...** button to select folders and files, and then click the **Import** button to begin the import process. If file import is successful, the newly created scenario database will be opened. If there is a problem, an error message like the one in Figure 8-2 may be displayed.

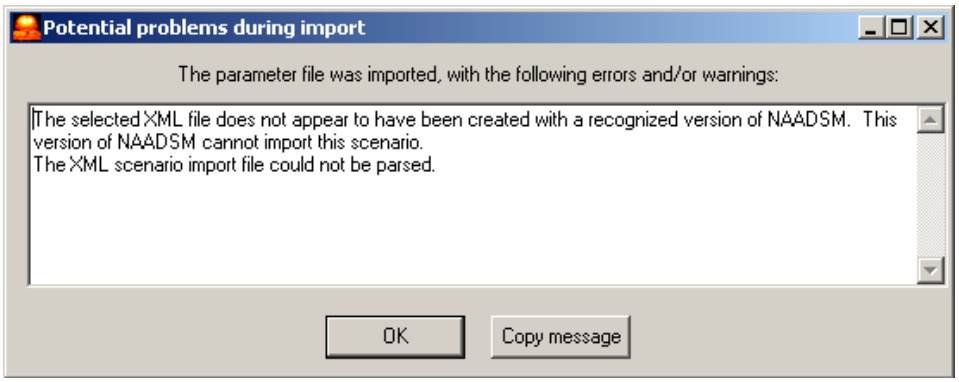


Figure 8-2. Errors generated during scenario input.

Such error messages should offer some guidance for users to determine the source of errors during file import.

Box 8-4. Error messages generated during XML import

NAADSM error messages generated during XML import are intended to be helpful to users as they troubleshoot their XML files to find the source of errors. In some cases, however, these error messages are more cryptic than useful. It is also possible that bugs in the application prevent successful import.

If you encounter problems during XML import and it is not clear how to solve them, please consider contacting a member of the *NAADSM* Development Team (current contact information can be found on the *NAADSM* website at <*http://www.naadsm.org/contacts*>). Such feedback would be useful as the Development Team works to improve future releases.

Box 8-5. Importing XML files from previous versions of NAADSM

It is not possible to import **`*.xml`** files generated by previous versions of *NAADSM* (*e.g.*, *NAADSM* 3.x) directly into *NAADSM* 4.0. If you have an XML file from a previous version that you wish to update to *NAADSM* 4.0, you must first create a *NAADSM* scenario file (a **`*.mdb`** file) by importing the XML file into the appropriate earlier version of *NAADSM*. This scenario file can then be opened with *NAADSM* 4.0, and will be updated as needed.

8.3.2. The Export scenario... command

Using the **File → Export scenario...** command will create *NAADSM/SC* **`*.xml`** files from an open scenario database. Before a scenario can be exported, *NAADSM* will ensure that all parameters are valid: for example, it will check to see that all required parameters are provided, and that all parameters have appropriate values. If a scenario is not valid, an error message is displayed that provides hints for correcting any problems that the application detected. See Section 9.18 for more about ensuring that you have a valid scenario.

Figure 8-3 shows the **Export scenario** window. When exporting *NAADSM/SC* **`*.xml`** files, you have the choice of exporting a list of herds (units), a set of all other input parameters (called a "scenario parameters file"), or both. The check boxes in the upper part of the window allow you to select the type(s) of file(s) to import.

If you choose to export a scenario parameters file (by checking the first checkbox in the **Export scenario** window), the middle portion of the window is activated, and presents you with some additional choices. If you plan to use the exported file with *NAADSM/SC*, be sure check the box labeled **Include output specification for NAADSM/SC.** You will also need to specify the condition that triggers the end of each iteration that you run. Typically, this condition will be the end of the outbreak, but other options are available. These are discussed in Chapter 10. Finally, select a folder and file name for the file(s) to be exported (using the appropriate **Browse...** button), and then click **Export** to generate the **`*.xml`** file(s).

Box 8-6. Including an output specification for NAADSM/SC

The XML-formatted scenario parameters file for *NAADSM/SC* includes code that specifies the set of outputs to be generated when the scenario is run. Unlike *NAADSM/PC*, which always records a full set of outputs, users of *NAADSM/SC* can customize the outputs generated by modifying the appropriate section of XML.

The best course of action for the vast majority of *NAADSM* users is to check the box labeled **Include output specification for NAADSM/SC** when a scenario database is exported to XML. Advanced users may prefer to create or edit the appropriate sections of XML by hand.

Without an output specification, *NAADSM/SC* will run quite happily, but will not store any output: an outcome that could be somewhat frustrating.

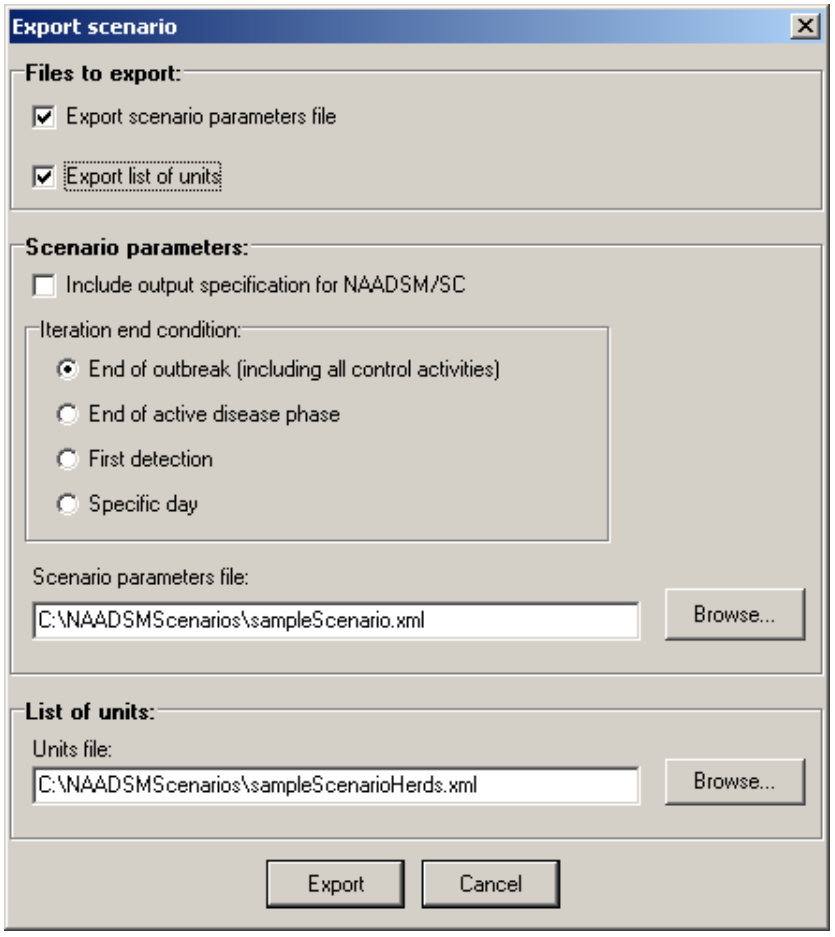


Figure 8-3. The **Export scenario** window.

8.4. 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.

Box 8-7. ***Warning****: Using a* NAADSM *scenario file simultaneously in several applications*

As noted above, *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 a *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.

8.5. 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.

8.6. 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.

8.7. 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.

9. Setting up a scenario: Understanding *NAADSM* and 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.

9.1. 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 9-1). Each of over 30 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, available at <*http://www.naadsm.org/documentation/specification*>. The scenario parameters screens and their various input options correspond very closely to the sections of the model description.

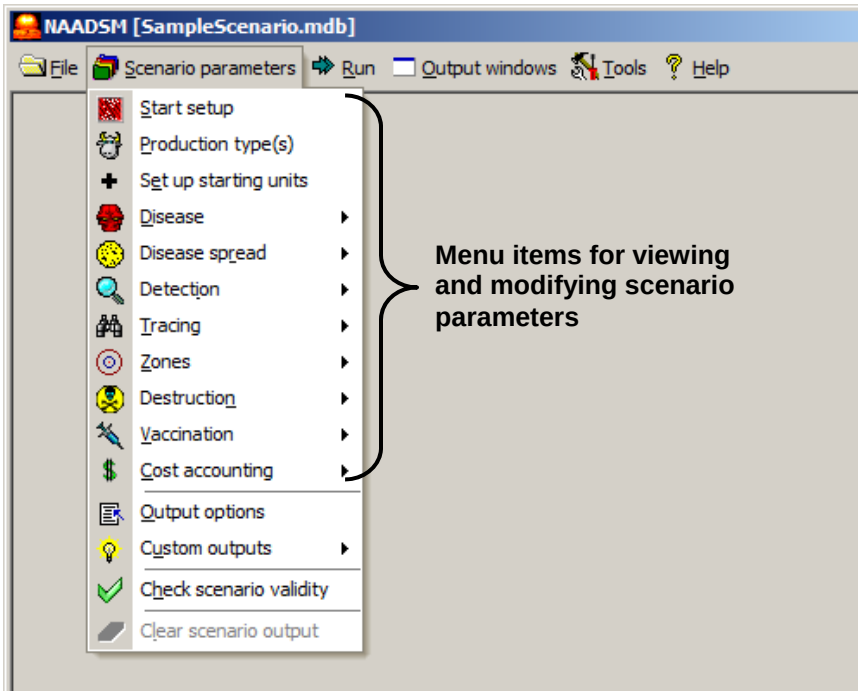


Figure 9-1. The "wizard" commands of the **Scenario parameters** menu.

Each of these menu commands activates a wizard screen, used to view or set scenario parameters associated with the indicated aspect of the scenario.

You can navigate linearly through the wizard screens, using the **< Back** or **Next >** buttons in the lower right corner of each screen (Figure 9-2). Alternatively, you may select one of the other items from the upper section of the **Scenario parameters** menu (Figure 9-1) or use the **Select...** button 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 scenario (see Chapter 6).

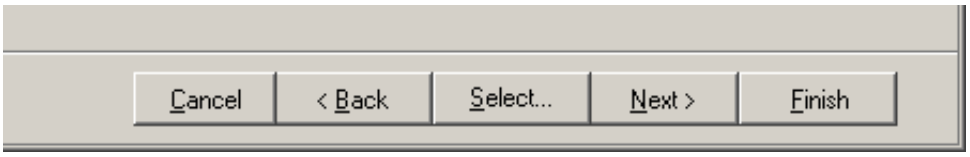


Figure 9-2. The "wizard" buttons, which appear on each of the wizard screens.

Use **< Back** to save changes made in the open window and navigate to the previous wizard screen. Use **Next >** to save changes and navigate to the next wizard screen. The **Select...** button will allow you to skip between all of the wizard screens available via the **Scenario parameters** menu. Use **Finish** to save changes and exit the parameters wizard. Use **Cancel** to discard changes made in the open window and exit the parameters wizard. Two additional "wizard" buttons, **Apply to all** and **Copy...**, appear on some of the wizard screens. The use of these two buttons is described in Section 9.5.2.1.

9.2. The Start setup window

The **Start setup** command and wizard screen (Figure 9-3) display several basic options associated with a scenario.

Scenario description: the scenario description is used to store comments concerning the particular scenario. Providing a description is optional, but potentially very useful and highly recommended.

Number of iterations: the number of iterations is the number of times that you want to run a particular scenario. Recall from Chapter 4 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.

Five to ten iterations are typically sufficient for scenario development and testing: you can watch the iteration in progress (see Section 11.2) to get a "feel" for the way a scenario might progress.

For analytical purposes, several hundred to several thousand iterations may be necessary, depending on the nature of the research question and the precision required, although that range may be impractical for large or complex data sets in *NAADSM/PC*. One hundred or two hundred iterations may be a good starting point.

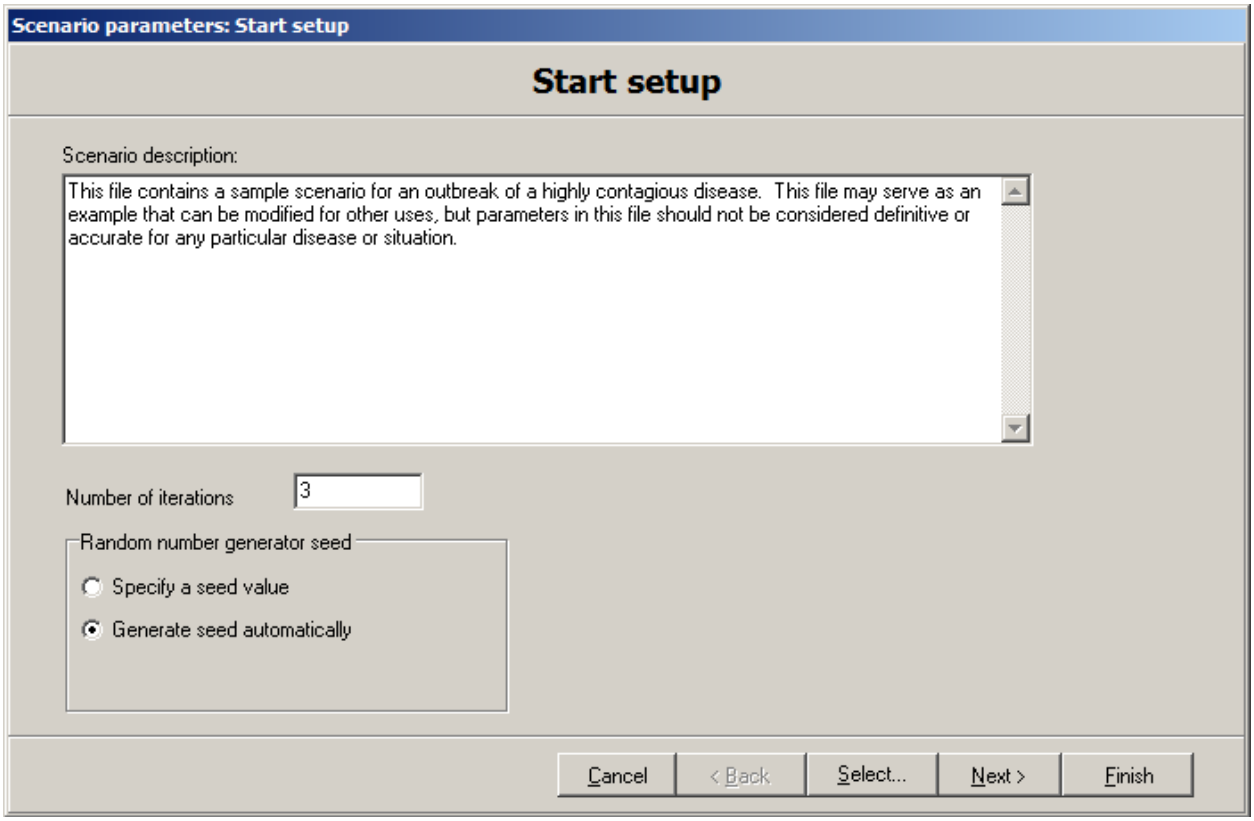


Figure 9-3. The **Start setup** wizard screen.

This form is 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.

9.3. The Production type(s) window

The **Production type(s)** command and wizard screen (Figure 9-4) are used to create (**Add production type**), rename (**Modify selected production type**), or delete (**Remove selected production type(s)**) production types (see Section 7.2.2). Once a production type has been created, subsequent wizard screens may be used to set all of the pertinent disease, contact, and control parameters.

> *Box 9-1.* ***Warning****: removing production types*
>
> When a production type is removed, all herds of that type will be deleted from the scenario as well.

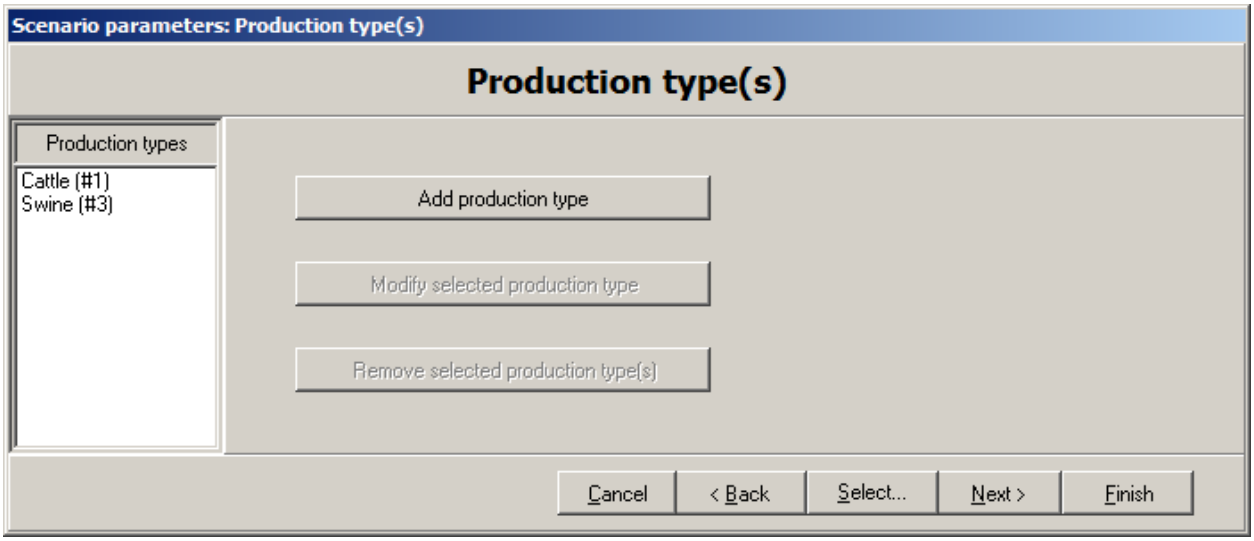


Figure 9-4. The **Production type(s)** wizard screen.

This form is activated by the **Scenario parameters → Production type(s)** command.

> *Box 9-2. 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 9.4.3). For convenience, the **Production type(s)** window displays these ID numbers, but is it not necessary (or possible) to manually enter ID numbers: the *NAADSM* application will assign them as needed.

9.4. Setting up a population of herds: The Set up starting units window

Scenario parameters → Set up starting units is used to activate the **Set up starting units** wizard screen (Figure 9-5). Use this screen to import, edit, and/or export your herd population data.

Box 9-3. 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://sourceforge.net/projects/jetsqlconsole/*>) provides a very efficient way to work with large database files: many records can be updated with just a few commands.

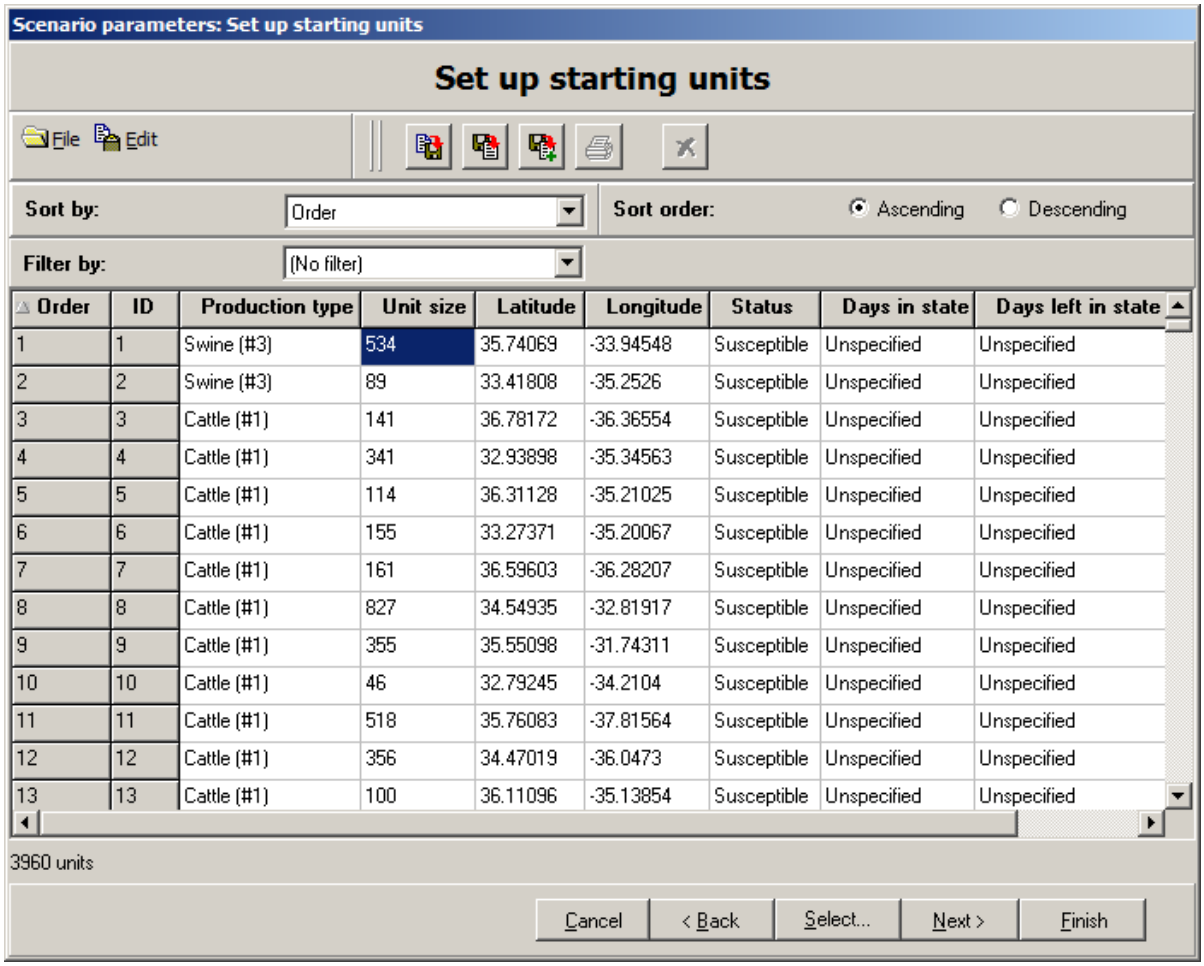


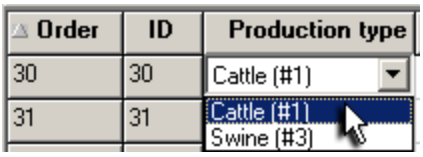
Figure 9-5. The **View/set up starting units** screen.

This form is activated by the **Scenario parameters → Set up starting units** command.

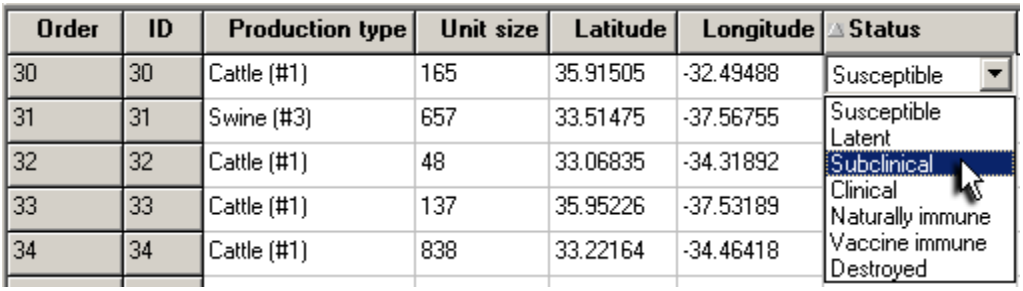
9.4.1. Editing the attributes of a herd

9.4.1.1. Changing basic attributes

Recall from Section 7.2.1 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 9.5). These attributes may be altered for any herd directly in the spreadsheet-like herd list window (Figure 9-5): 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 different option from the dropdown menu that appears when you click on a cell in columns labeled **Production type** or **Status** (Figure 9-6). For latitude, longitude, or herd size, simply type a number in the cell.



(a)



(b)

Figure 9-6. 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 drop-down 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 9.3. **(b)** The initial disease state of unit 31 is being set to "Subclinical", *i.e.*, infectious but without clinical signs of disease.

Box 9-4. Herd order and ID

You may have noticed additional columns in Figure 9-5 and Figure 9-6, labeled **Order** and **ID**. These two numbers may be used for sorting and filtering if desired (see Section 9.4.2), 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.

9.4.1.2. Controlling a herd's time in its initial disease state

The last two columns shown in Figure 9-5 are labeled **Days in state** and **Days left in state**. Recall from Section 7.2.1 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 7.2.1 and the *NAADSM* model specification for more details).

Most of the time, it is probably best to allow *NAADSM* to operate as usual. In some specific cases, though, you may wish to specify exactly the number of days that a herd has been in its initial disease state (this can influence disease detection: see Section 9.8), or how many additional days a herd will remain in its initial state before transitioning to the next disease state. The former can be accomplished by setting the value of **Days in state** to something other than

Unspecified. Likewise, change the value of **Days left in state** to specify the latter. Simply enter a positive whole number in this field to change the initial setting. A value of -1 for either **Days in state** or **Days left in state** is equivalent to **Unspecified**.

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

9.4.2. 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.

9.4.2.1. 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 9-7a). 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 9-7).

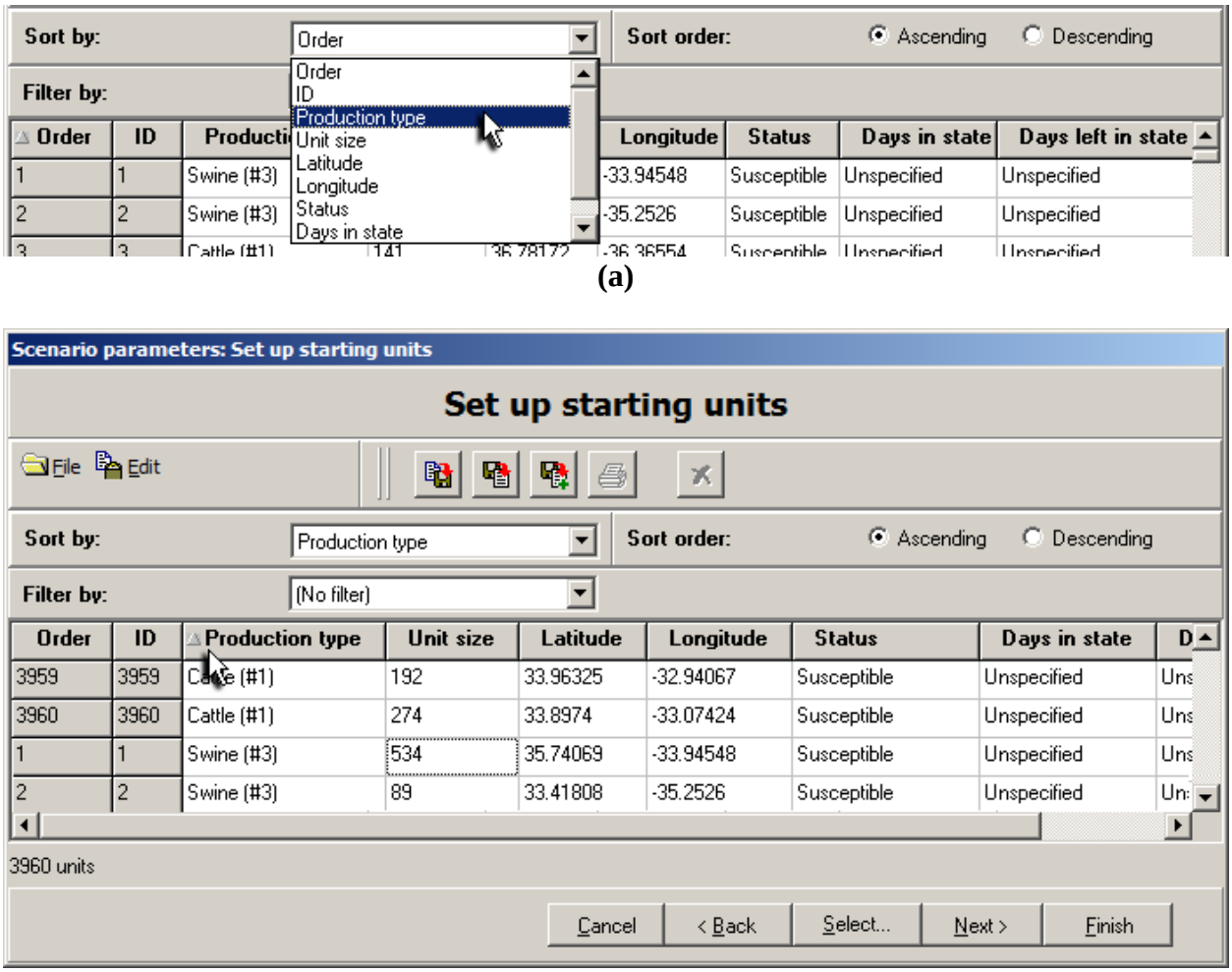


Figure 9-7. Sorting herds by production type.

(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**.

9.4.2.2. Filtering herds by simple criteria

You can display a subset of your population data in the **Set up 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 9-8a). 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 9-8b shows an example of the latter).

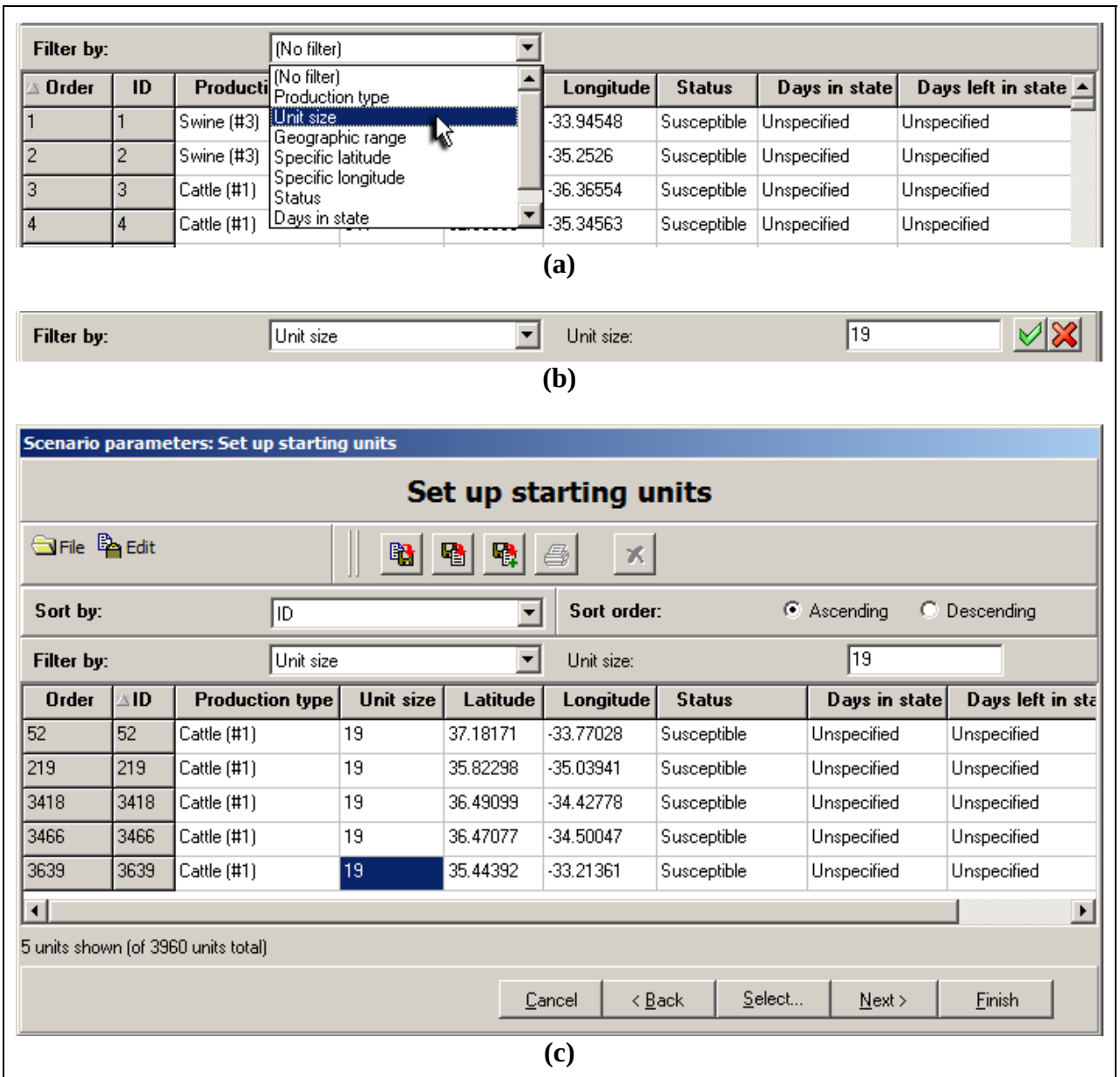


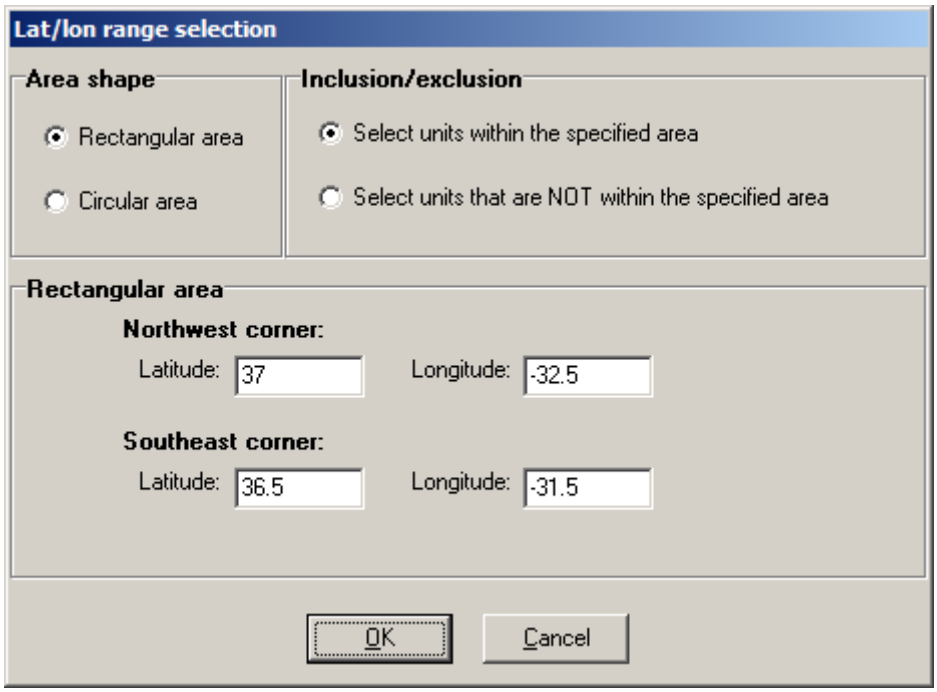
Figure 9-8. Filtering herds by herd size.

(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.

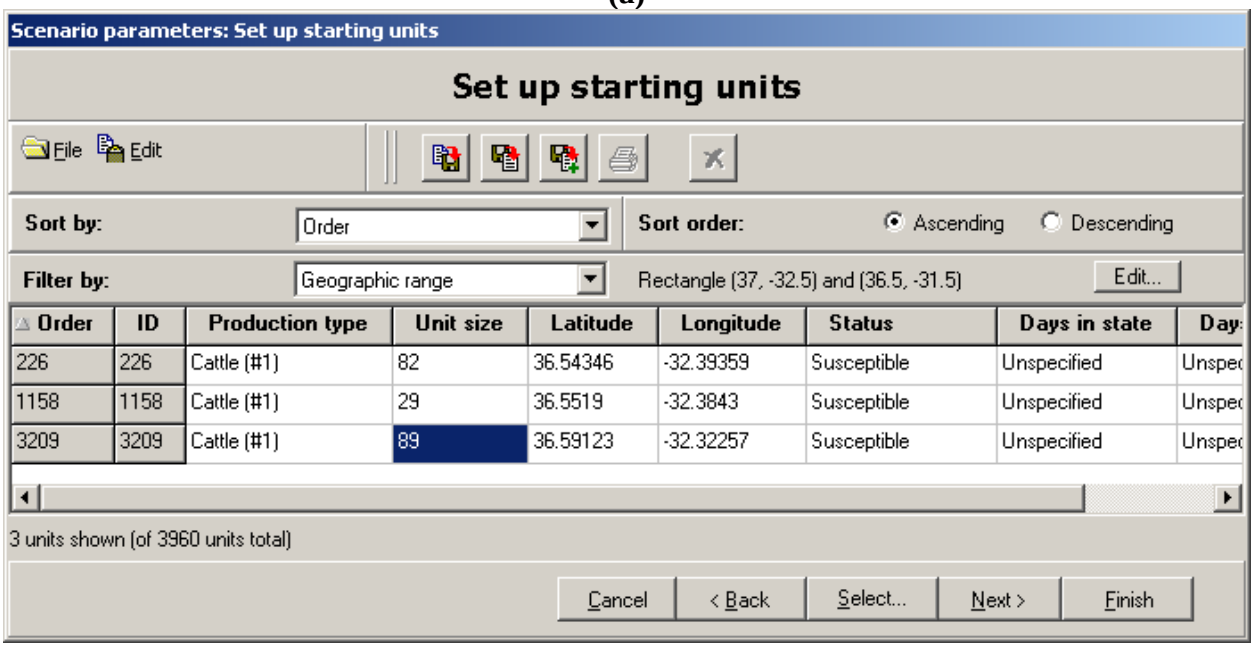
9.4.2.3. Filtering herds by geographic range

When editing herds, it may be useful to restrict the displayed list to herds contained within a specific geographic range. The **View/set up 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 9-9).



(a)



(b)

Figure 9-9. 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 9-10).

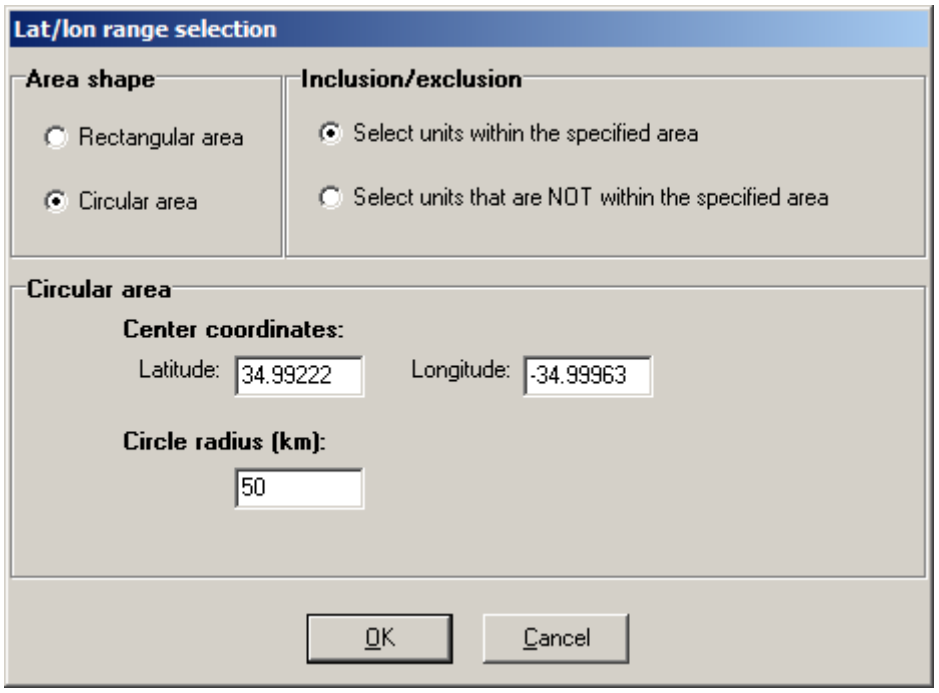


Figure 9-10. Filtering herds by circular geographic range.

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.

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.

9.4.3. Importing and exporting herds

If you have your own herd data, you may want to import it into a *NAADSM* scenario. There are several commands on the **File** menu of the **Set up 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 Chapter 8). 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.

9.4.3.1. 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 fields for days in state and days left in state. Column headers must be included in your **`*.csv`** file for import, as shown in Table 9-1. Columns may be given in any order.

Table 9-1. Fields in *NAADSM* **`*.csv`** files.

Field name	***Field description***
UnitID	Unique integer identifier for each unit. ID must be greater than 0.
ProductionType	Identifier for the unit's production type (see note below).
UnitSize	Integer indicating the number of animals in the unit.
Lat	Real (floating point) number indicating the latitude of the unit. Values must be between -90 and 90, inclusive.
Lon	Real (floating point) number indicating the longitude of the unit. Values must be between -180 and 180, inclusive.
Status	Code indicating the unit's disease transition state at the beginning of the simulation (see Table 9-2).
DaysInState	(Optional) Integer indicating the number of days the unit has been in its initial state (see Section 9.4.1.2).
DaysLeftInState	(Optional) Integer indicating the number of days the unit has remaining in its initial state (see Section 9.4.1.2).

Box 9-5. 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 9.3) 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 9.3). 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 B for details.

Initial disease state is usually indicated by a single upper case character, as shown in Table 9-2.

Table 9-2. 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
Dead from disease	X
Vaccine immune	V
Destroyed	D

Figure 9-11 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 B. A detailed description of the **`*.xml`** file formats used by *NAADSM* is beyond the scope of this guide, but resources are listed in Appendix C and an example is illustrated in Appendix B.

```
UnitID,ProductionType,UnitSize,Lat,Lon,Status,DaysInState,DaysLeftInState
1,"Swine",534,35.740688,-33.945477,S,-1,-1
2,"Swine",89,33.418079,-35.252598,S,-1,-1
3,"Cattle",141,36.781723,-36.365536,S,-1,-1
4,"Cattle",341,32.938984,-35.345634,S,-1,-1
5,"Cattle",114,36.311275,-35.210247,S,-1,-1
6,"Cattle",155,33.273712,-35.200668,S,-1,-1
7,"Cattle",161,36.596031,-36.282066,S,-1,-1
8,"Cattle",827,34.549351,-32.819172,S,-1,-1
9,"Cattle",355,35.55098,-31.743107,S,-1,-1
```

Figure 9-11. Part of a **`*.csv`** file suitable for import into a *NAADSM* scenario.

Recall from Section 9.4.1.2 that a value of -1 for **DaysInState** or **DaysLeftInState** is equivalent to **Unspecified**.

9.4.3.2. 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 **Set up starting units** toolbar (Figure 9-12). Then select the **`*.csv`** or the **`*.xml`** file that you wish to import.

If you append new units to an existing unit list, the herd ID numbers in your 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.

9.4.3.3. 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 **Set up starting units** toolbar (Figure 9-12). Then select the **`*.csv`** or **`*.xml`** file that you wish to import.

If you replace an existing unit list, the herd ID numbers in your `*.csv` file (see Section 9.4.3.1) will be preserved: contrast the situation when new herds are appended to an existing list, described in Section 9.4.3.2.

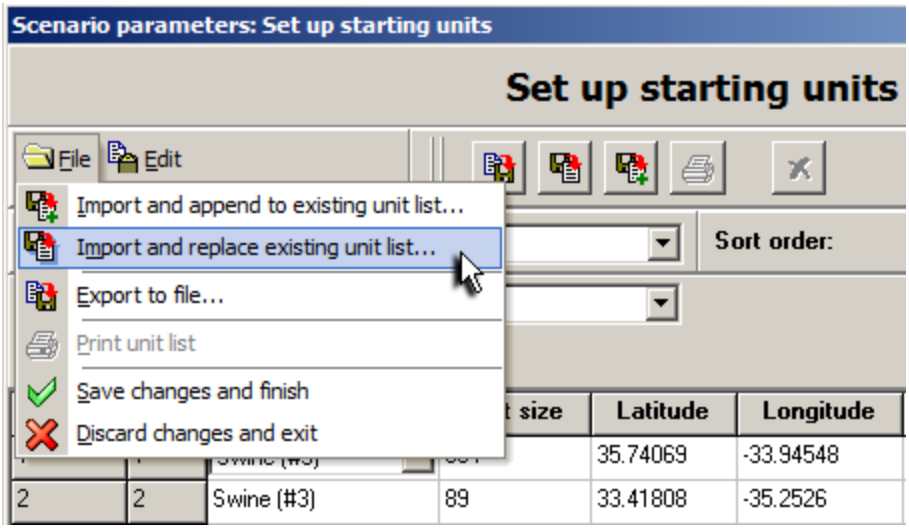


Figure 9-12. The **Set up starting units** toolbar.

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 B, 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.

9.4.3.4. 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 9-13 will be displayed. Using the options shown in Figure 9-13 will produce a `*.csv` file in the format discussed above (Section 9.4.3.1). Other options are described in Appendix B. Simply enter a file name and click **OK** to save your herd list as a file.

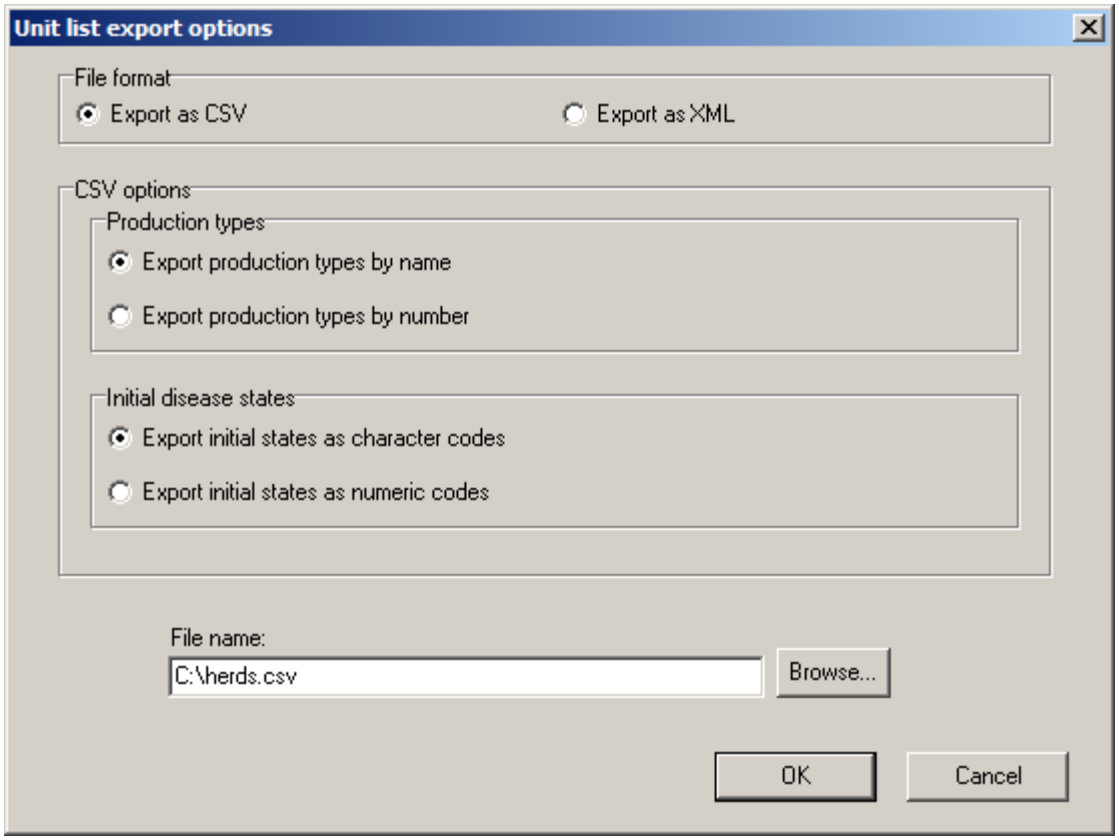


Figure 9-13. Herd file export options.

9.4.4. 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 9-14). 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 that appear consecutively in the list, hold the SHIFT key while clicking in the **Order** or **ID** column for the first and last units that you wish to select.

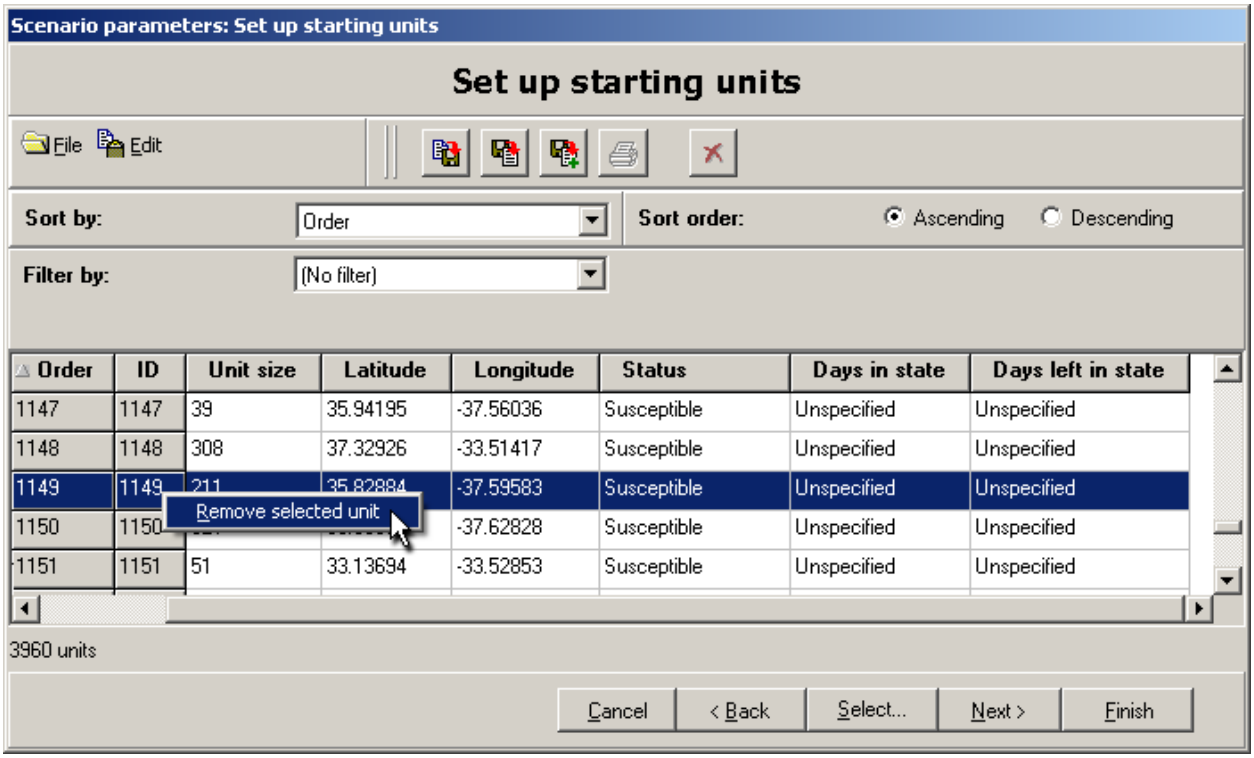


Figure 9-14. Selecting and deleting a herd.

This herd was selected by clicking in the **Order** column. Right-click to display the context menu. Use **Remove selected unit** to delete this herd from the scenario file.

9.4.5. Finalizing changes to starting units

As in all of the wizard screens described in Chapter 9, 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.

9.5. Modeling disease progression

9.5.1. Understanding disease states and progression in *NAADSM*

9.5.1.1. Herd-level disease states

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 9-3.

Table 9-3. Disease and disease-like transition states used in *NAADSM*.

Transition state	*Description*
Susceptible	Susceptible to infection, *i.e.*, lacking immunity
Latent	Infected, but not yet shedding the infectious agent, and not yet showing clinical signs of disease
Subclinical	Infected and shedding the infectious agent, but not yet showing clinical signs of disease
Clinical	Infected, shedding the infectious agent, and showing clinical signs of disease
Naturally immune	Immune from infection as a result of previous infection
Dead from disease	Dead due to disease-related mortality
Vaccine immune	Immune from infection as a result of vaccination
Destroyed	Removed from the population by destruction/depopulation

One of the peculiarities of working with *NAADSM* is that these states are applied to entire herds, rather than to individual animals. This approach is used primarily for two reasons:

- It is assumed that entire herds will be the unit of control: that is, control measures are applied equally to every animal in a herd. This would be the case for highly contagious animal diseases (for which *NAADSM* is designed) in which entire herds would be subjected to measures like quarantine, vaccination, or depopulation.
- This assumption substantially simplifies the program required for simulation, and makes it possible to run models more quickly.

There is no single rule to define exactly how users should interpret the meaning of a disease state at the herd level. One approach would be to assume that all animals within a herd have the same disease state. Every animal in a latent herd might be assumed to be latent (*i.e.*, infected but not yet capable of transmitting infection), every animal in a clinical herd would be

clinical (infected, capable of transmitting infection, and showing clinical signs of disease), *etc*. This approach has the advantage of simplicity, but it is not necessarily biologically plausible: an infected herd, for example, is likely to include animals in various states of infection, and may include some individual animals which never become infected.

A more refined way to define herd-level disease states is shown in Table 9-4. This approach accounts for the fact that an infected herd will have animals in different stages of infection, but still treats the entire herd as the unit of disease control.

Table 9-4. Herd-level disease state interpretations that account for differences in state among individual animals.

Transition state	***Description***
Susceptible	Units that contain no infected animals and lack effective herd immunity to disease.
Latent	Units that contain at least one latent animal (*i.e.*, at least one animal is infected but is not yet capable of transmitting infection and is not yet showing clinical signs of disease), but have no subclinical or clinical animals.
Subclinical	Units that contain at least one subclinical animal (*i.e.*, at least one animal is infected and capable of transmitting infection, but is not yet showing clinical signs), but have no clinical animals.
Clinical	Units contains at least one animal in the clinical stage (*i.e.*, at least one animal is infected, infectious, and showing clinical signs of disease).
Naturally immune	Units that contain no infected animals, and have effective herd immunity as a result of previous infection. (Note that not every individual must be immune for herd immunity to exist.)
Dead from disease	Units in which all or nearly all animals have died from disease, and which contain no infected (*i.e.*, latent, subclinical, or clinical) animals.
Vaccine immune	Units that contain no infected animals, and have effective herd immunity as a result of vaccination. (Note that not every individual must be immune for herd immunity to exist.)
Destroyed	Units in which every animal has been destroyed.

The way that you as a modeler choose to define disease states will influence subsequent decisions about model parameters. For example, the values or distributions chosen to represent the durations of individual disease states or the probabilities of spread of disease from an infected herd are at least partly dependent on the assumptions about herd-level disease states. As noted in Chapter 1, it is your responsibility to clearly document and justify the interpretations that you use, so that the validity of the models that you produce can be independently assessed.

9.5.1.2. Disease progression: Transitions between disease states

Without intervention, a typical newly infected herd will progress from its (herd-level) latent stage through the subclinical, clinical, and naturally immune stages before again becoming susceptible to infection. If susceptible units are vaccinated (see Section 9.14), they can become vaccine immune for a certain period of time. Herds in any stage may be subject to destruction (depopulation). Figure 9-15 shows all of the disease states in *NAADSM* and all possible transitions from one state to another.

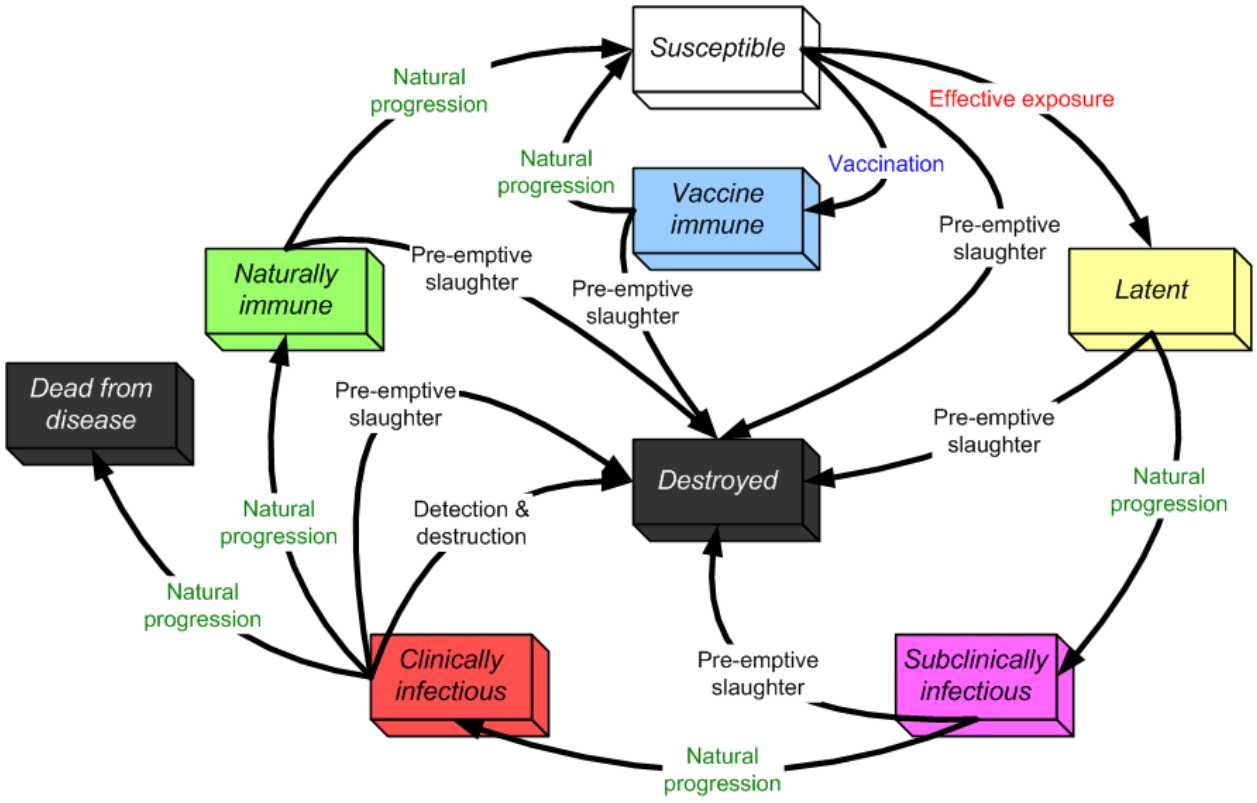


Figure 9-15. The disease cycle in *NAADSM* 4.0.

Labeled steps indicate possible mechanisms for the indicated transitions between states, adapted from Harvey *et al*. (2007).

For each production type, the user defines the length in days of all of these periods except the susceptible, destroyed, and dead-from-disease 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. Similarly, units that reach the dead-from-disease state remain in that state for the remainder of an iteration, and are not repopulated. The remaining five stages are associated with a time period determined from an appropriate, user-defined 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.

9.5.2. Setting disease parameters

Use the menu command **Scenario parameters → Disease** (Figure 9-16) to specify parameters for disease states. The first of the two wizard screens (Figure 9-17, accessed using the command **Scenario parameters → Disease → Disease options**) presents the user with the option whether to use within-unit prevalence. As noted above, it is not biologically plausible to assume that every animal in an infected herd is itself infected. Prevalence of disease within an infected herd will change over time. This option allows users to represent the change over time in within-herd prevalence of disease. Users may also choose a simpler (but potentially less realistic) approach, which does not account for dynamic prevalence of disease within herds.

Box 9-6. Simplicity versus reality in modeling

The choice whether or not to include a representation of within-unit prevalence of disease is the first of many examples we will see of having to choose between model simplicity and realism. There is no definitive answer as to which approach to use. In fact, this topic has been the subject of some debate in the scientific literature: see for example Carpenter *et al.* (2004), Kostova-Vassilevska (2004), Savill *et al.* (2007), and Reeves *et al.* (submitted for publication) for a discussion of different approaches.

What is important is that you as the modeler can justify and defend your choices regarding simplicity or realism, which will depend largely on the nature of the questions that you are attempting to address. A simple approach may be sufficient in some cases but not in others.

The within-unit prevalence option selected on the window shown in Figure 9-17 affects how the simulation of disease spread occurs. For this reason, a full discussion of the within-unit prevalence option is deferred until Section 9.7.

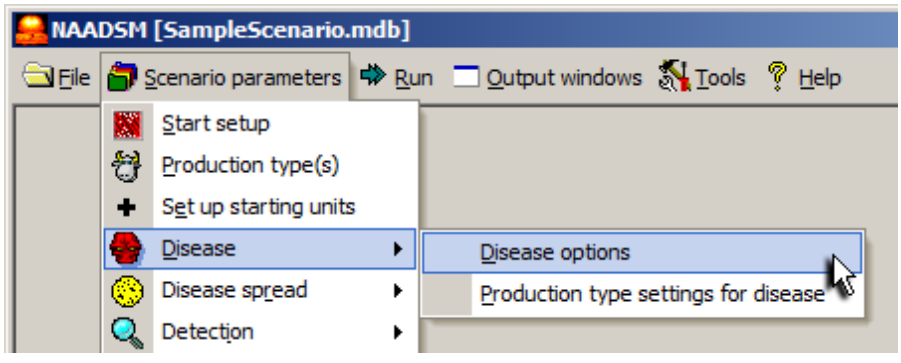


Figure 9-16. Commands of the **Scenario parameters → Disease** menu.

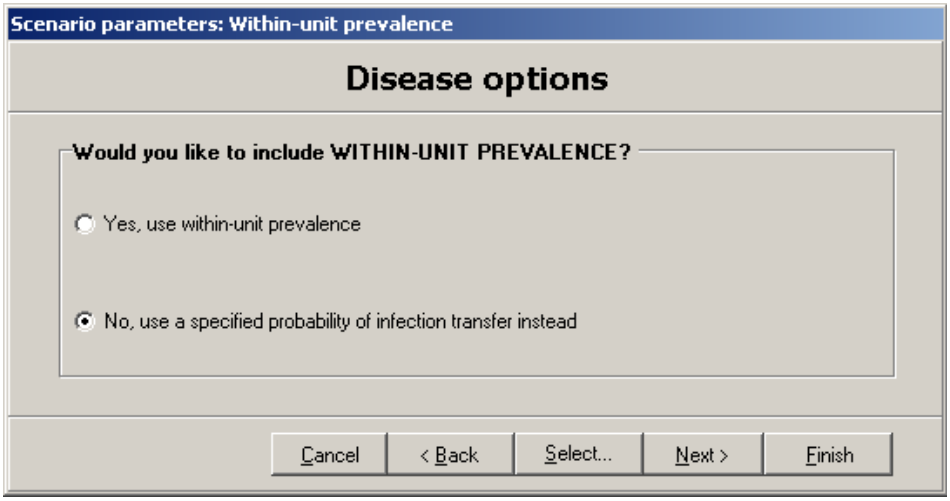


Figure 9-17. The **Disease options** window.

The within-unit prevalence option is discussed in Section 9.7.

9.5.2.1. Setting disease state durations

The **Disease** window (Figure 9-18) is used to set parameters for the duration of each disease stage (latent, subclinical, clinical, and immune) for each production type (see Sections 7.2.1 and 7.2.2). 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.

Simulate disease progression 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 parameters will appear.

Box 9-7. Using the **Simulate disease progression in units of this production type** *check box*

In almost every conceivable situation, herds of a particular production type will be included in a model only if you plan to simulate their involvement in the disease process. It is very unlikely that this option will not be checked.

If you encounter a situation in which you would not select this option, carefully consider whether the production type needs to be included at all. The simulation will proceed more efficiently without unused herds and production types.

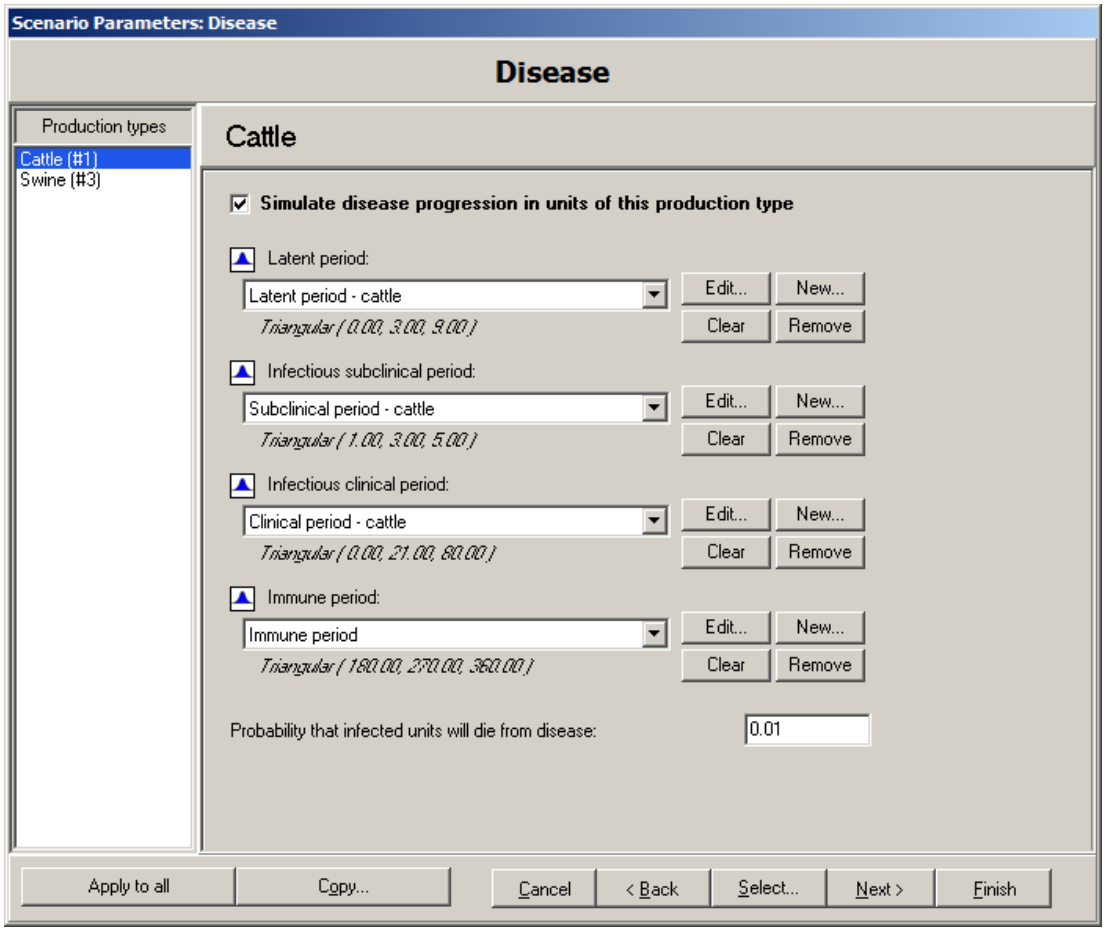


Figure 9-18. The **Disease** window.

Select a production type from the list on the left to display the disease parameters for that type. Note: if you have selected to use within-unit prevalence (see Sections 9.5.2 and 9.7), additional parameters for within-unit prevalence are displayed in this window.

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 7.1.5).

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 showing clinical signs of disease. Select or create a function with the *pdf* selection box (see Section 7.1.5).

Infectious clinical period: this variable is a probability density function (*pdf*) defining the duration of the period when herds of this production type are showing clinical signs of disease. Select or create a function with the *pdf* selection box (see Section 7.1.5).

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 7.1.5).

Probability that infected units will die from disease: this parameter is used to set the likelihood that an infected unit of the indicated production type will die from disease, *i.e.*, make the transition to the dead-from-disease state. Units that do not die from disease will become naturally immune. Specify a probability from 0 (indicating no disease mortality) to 1 (indicating that every infected unit will ultimately die from the disease).

Within-unit Prevalence of infection and **Within-unit prevalence of infectiousness**: if the option to use within-unit prevalence is selected from the **Disease options** window (Figure 9-17), then these parameters appear in the **Disease** window. See Section 9.7 for a description of these optional parameters.

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 9-2).

Box 9-8. Additional wizard screen buttons: **Apply to all** *and* **Copy**

Figure 9-18 shows two additional buttons along the bottom of the wizard screen, labeled **Apply to all** and **Copy...**. These buttons appear on all wizard screens that are used to set parameters for individual production types or for individual combinations of production types (Section 7.2.3).

In some cases, you may wish to use the same parameter values for several or even all production types or combinations included in your model. Click the **Apply to all** button to apply the currently displayed parameter values to every production type or combination in your model. A confirmation dialog will be displayed before the new parameters are applied to every production type or combination.

If you want to apply the current set of parameters to only some production types or combinations, click the **Copy...** button. You will then be presented with a list of production types or combinations to choose those to which the current parameter values should be applied.

9.6. Modeling disease spread

9.6.1. Understanding the basics of disease spread

9.6.1.1. Exposures, sources, and recipients

In *NAADSM*, disease may be spread from an infected herd to a susceptible herd. This transmission may occur by several mechanisms, each of which is described below. Every event that might result in spread of disease (*i.e.*, every *exposure*) involves the interaction of exactly two herds. There can be many such exposures, and individual herds might be involved in many exposures, but each individual exposure involves only two herds.

The *source* of an exposure is the infected herd from which a contact or exposure originates. The *recipient* or *destination* is the herd that is exposed. This relationship between sources and recipients of exposure is illustrated in Figure 9-19.

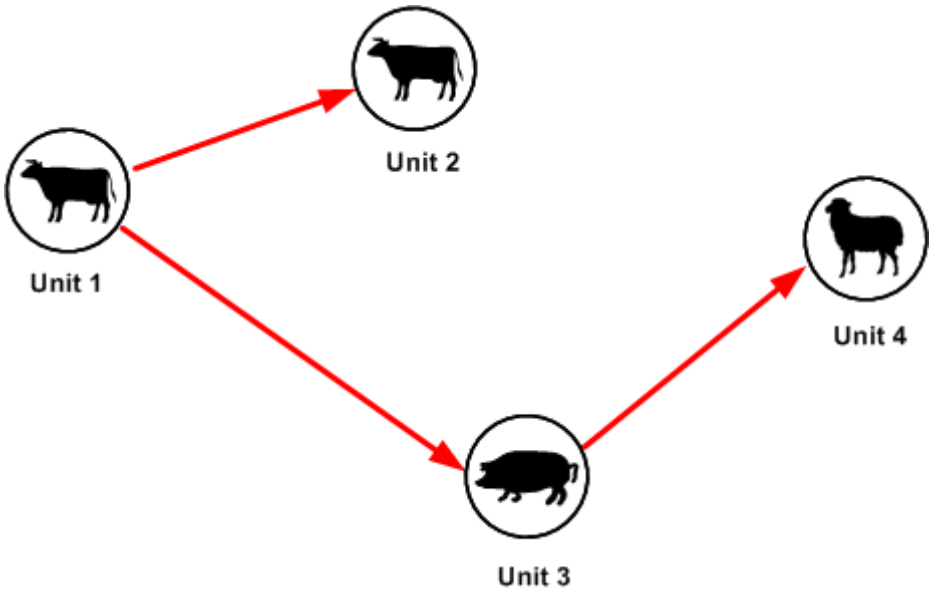


Figure 9-19. Sources and recipients of exposures that might result in disease spread.

Units (herds) are shown in black. Units of three different production types are indicated: units 1 and 2 have the production type “Cattle”, unit 3 has production type “Swine”, and unit 4 has production type “Sheep”. Exposures and their directions are shown as red arrows. Unit 1 is the *source* of two separate exposures. Unit 2 is the *recipient* of one exposure. Unit 3 is the *recipient* of one exposure (which has unit 1 as its *source*), and is the *source* of a different exposure (which has unit 4 as its *recipient*). These exposures could result in transmission of disease if the conditions described in the text are met.

When an exposure occurs, it may or may not result in the transmission of disease, depending on its *adequacy* and on its *effectiveness*. An *adequate* exposure is an exposure that would have resulted in disease spread *if it had occurred between an infectious source and a susceptible recipient*. If and only if these conditions exist, an adequate exposure is guaranteed to result in disease transmission. For each form of disease spread, the modeler specifies a probability that each exposure will be adequate. *NAADSM* applies these probabilities in a stochastic way to simulate whether an exposure is adequate or not. Adequate exposures can occur between herds regardless of their disease states, but such exposures will result in disease spread only when the source is infectious and the recipient is susceptible.

Any exposure that actually does result in disease transmission is said to be *effective*. Thus, an adequate exposure is effective if and only if it occurs between an infectious source herd and a susceptible recipient herd. Similarly, an exposure that is not adequate cannot be effective. The examples in Table 9-5 further illustrate the concepts of adequate and effective exposures.

Table 9-5. Examples of adequate and effective exposures.

Disease state: source herd	***Disease state: recipient herd***	***Adequate exposure[1]***	***Effective exposure[2]***	***Comments***
Infectious	Susceptible	Yes	Yes	The conditions are right for disease transmission.
Infectious	Susceptible	No	No	Although the disease states are right for disease transmission, an inadequate exposure can never result in disease spread.
Infectious	Infectious	Yes	No	Although this contact is adequate, it is not possible to transmit disease to a unit that is already infected.
Susceptible	Infectious	Yes	No	Although this contact is adequate there is no disease to transmit from the susceptible source.
Infectious	Vaccine immune	Yes	No	Although this contact is adequate, immune recipients cannot be infected.

[1] The probability that an exposure will be adequate is specified by the user as an input into the simulation.
[2] Whether an exposure is effective (*i.e.*, whether an exposure is results in disease transmission) is an outcome of the simulation.

9.6.1.2. Forms of disease spread

NAADSM 4.0 simulates four methods by which disease can be spread. Any combination of these methods may be used in each *NAADSM* simulation. These mechanisms are as follows:

Direct contact is contact that directly involves animals in a source herd coming in contact with animals in a recipient herd. Direct contact spread is described in more detail in Section 9.6.3.

Indirect contact typically involves the movement of people, materials, vehicles, equipment, animal products, *etc.*, among herds. Indirect contact spread is described in more detail in Section 9.6.3.

Local-area spread is the non-directional transmission of disease that cannot be well-characterized or traced, such as spread by insects, pests, lapses in biosecurity, and localized aerosol transmission. Local-area spread is discussed in Section 9.6.4.

Airborne spread involves the directional transmission of disease from a source to a recipient herd by dispersal of virus particles or other causal agents via the atmosphere. Airborne spread is described in more detail in Section 9.6.4.1.

9.6.2. Initiating *NAADSM* settings for 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 9-20). Depending on your selections in the **Spread options** window, one or more of the other menu commands may be disabled.

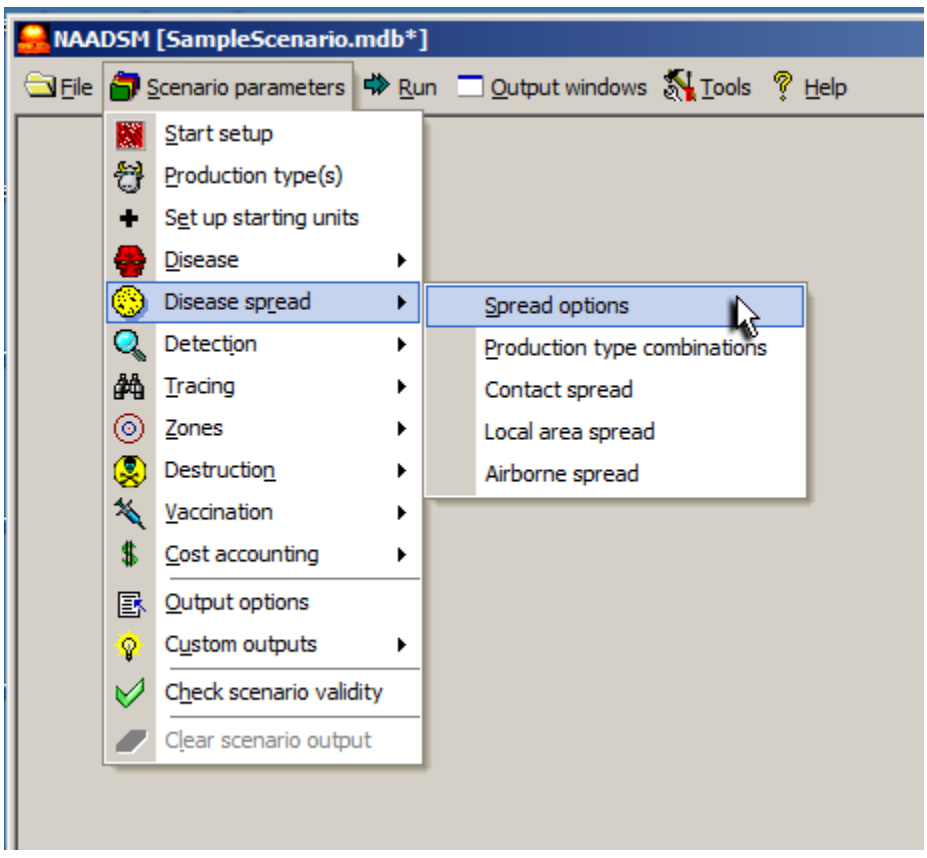


Figure 9-20. Commands of the **Scenario parameters → Disease spread** menu.

9.6.2.1. The Spread options window

The **Spread options** window (accessed via the menu command **Scenario parameters → Disease spread → Spread options**) asks the question **Which type(s) of disease spread would you like to model during simulation runs?** Use the check boxes to select whether you want *NAADSM* to simulate spread via **Contact** with an infected herd (described in Section 9.6.3), **Local-area** spread (Section 9.6.4), **Airborne** spread (Section 9.6.4.1), or any combination of these options. Then use one of the wizard buttons (Figure 9-2) to move to the next screen.

If you choose to simulate airborne spread, then an additional set of parameters, labeled **For airborne spread**, will be presented as shown in Figure 9-21. These parameters describe the area at risk of exposure by an infectious unit. If the infectious unit is located in the center of the circle, then the region shown in blue is the region that could be exposed to disease, based on directional airborne spread. Directionality is described by degrees (range: 0 to 360) The circle gives a visual representation of the range that you specify. A range from 0 to 360 will color the entire circle blue, giving the non-directional spread (see Section 9.6.4 below regarding local-area spread). Smaller ranges (*e.g.*, 45 to 135, as shown in Figure 9-21) will restrict the region at risk based on direction, thus leaving part of the circle gray.

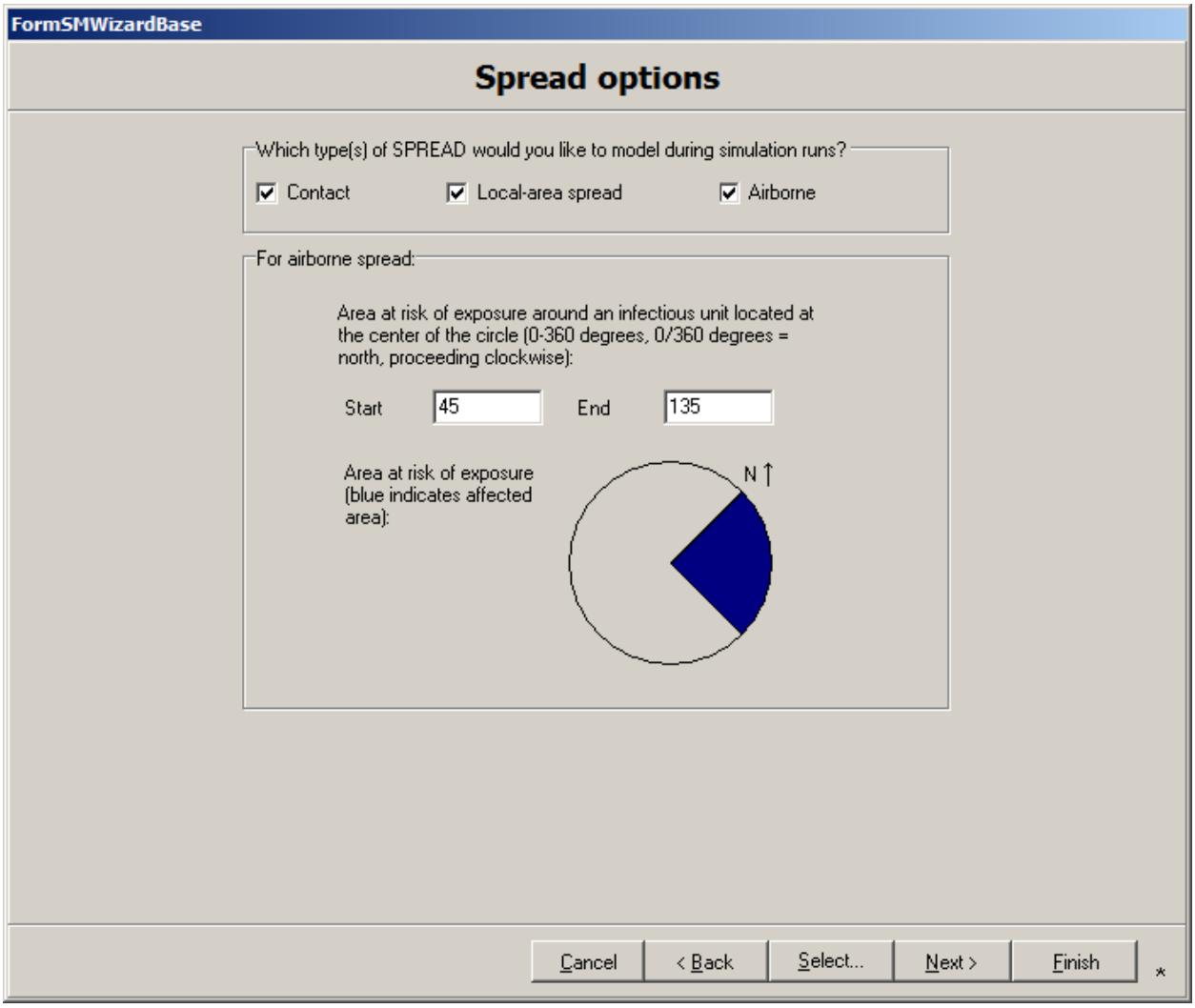


Figure 9-21. The **Spread options** window, showing parameters for specifying directionality of airborne spread.

9.6.2.2. The Production type combinations window

Recall from Section 7.2.3 that disease spread parameters are set for pairs of production types. If any kind of disease spread is selected in the **Spread options** window (Section 9.6.2.1), the **Production type combinations** window will be displayed (Figure 9-22). 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.

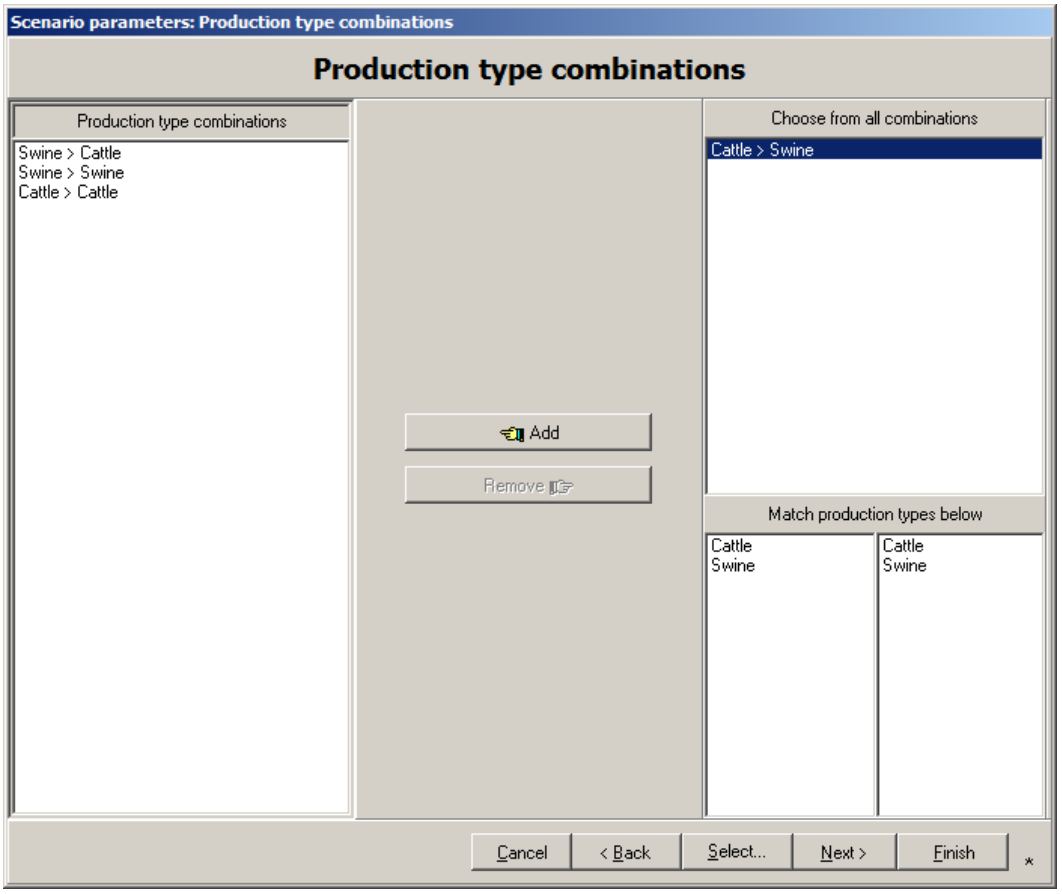


Figure 9-22. The **Production type combinations** window.

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 **Swine > Cattle** shown in Figure 9-22, the source of disease is **Swine** and the destination is **Cattle**. In other words, disease can be spread from units (herds) of the production type "Swine" to units of the production type "Cattle". 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.

9.6.3. Direct and indirect contact spread

9.6.3.1. Understanding contact spread

As mentioned above, ***direct contact*** involves animals in a source herd coming into contact with animals in a recipient herd. Most *NAADSM* users restrict the definition of direct contact slightly to include only the physical introduction of one or more animals from a source herd into a recipient herd. Other forms of contact that might technically be considered "direct" (for example, "across the fence" contact between animals on different farms) are most often incorporated by a different mechanism. This approach can make it easier to develop values for the parameters required for direct contact discussed below.

> *Box 9-9. Direct contact and the transfer of animals between herds*
>
> Although direct contact involves the *conceptual* transfer of animals between herds, *NAADSM* does not explicitly simulate the movement of animals from one herd to another. The number of animals in a herd does not change during a simulation (except when a herd is depopulated or dies out from disease).

By contrast, ***indirect contact*** typically involves the movement among herds of people, materials, vehicles, equipment, animal products, and any other items that might be capable of carrying the infectious agent from herd to herd. The parameters for direct and indirect contact are similar, but these two mechanisms operate independently from one another, and with some different assumptions, as discussed below.

As discussed in Section 9.6.1, every contact has a source and a destination. Contacts in *NAADSM* are *not* simulated as a sequence of visits to herds. There is currently no way in *NAADSM* to simulate contact that might occur along a "route" that goes from herd 1, then to herd 2, next to herd 3, next to herd 4, and so on.

9.6.3.1.1. 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 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 A). 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. Separate contact rates can be specified for every combination of source and recipient production types included in a *NAADSM* scenario.

Box 9-10. The Poisson distribution

Poisson distributions are commonly used to model the number of events that will occur within a particular period of time based on an average rate of occurrence. The number of contacts that will occur per day is an example of this kind of phenomenon, so the developers of *NAADSM* have chosen to use the Poisson distribution.

The choice of the Poisson distribution assumes that contacts between premises follow a *Poisson process*. A discussion of stochastic processes is beyond the scope of this guide, but *NAADSM* users may wish to consult other sources on simulation modeling to develop an understanding of the assumptions of the Poisson and other stochastic processes. The book *Quantitative Risk Analysis: A Guide to Monte Carlo Simulation Modeling* (Vose, 2000) is an indispensable reference which includes a discussion of this material, and is highly recommended background reading for all *NAADSM* users.

Box 9-11. Fixed baseline contact rates

NAADSM also offers the option of specifying a fixed number of daily contacts. Some production types might have tightly controlled shipment schedules. In cases like this, it might be helpful to specify a fixed number of daily contacts. If you want to generate exactly five contacts per day for each source herd, you can provide a fixed contact rate of five recipient units per source unit per day.

It is also possible to generate a shipment that occurs every *x* days. For example, to specify that a shipment will occur exactly every two days, use the fixed baseline contact rate of 0.5 recipient herds per source herd per day. To specify that a shipment will occur exactly once every 30 days, use a fixed baseline contact rate of 1/30 = 0.03333 recipients per source per day.

Like the mean initial contact rate, a fixed baseline contact rate is affected by movement restrictions (see Section 9.6.3.1.4).

9.6.3.1.2. Probabilities of infection transfer

Not every contact that takes place will necessarily result in the transmission of disease. For every combination of source and recipient production types, the user establishes a probability that a contact, if it occurs from an infected herd to a susceptible herd, will transmit infection. This probability is then applied stochastically to every contact that occurs in a simulation.

Like all other parameters related to disease spread, the probability of infection transfer can be set separately for direct and indirect contacts for each combination of source and recipient production type.

For indirect contacts, a probability of infection transfer is always specified. For direct contacts, a probability may be specified, or this probability can be determined based on prevalence of disease within a herd: see Section 9.7 for a discussion of the effects of the within-herd prevalence option.

9.6.3.1.3. Contact distances

The distances over which contacts occur in *NAADSM* are determined in part by a probability density function (see Section 7.1.5). For every contact that originates from a particular source herd, a value is drawn from an appropriate user-specified *pdf* that represents a straight-line distance to a potential recipient herd.

It is highly unlikely that a recipient herd will be identified that is exactly the desired distance the source herd, so *NAADSM* identifies the recipient of the contact by finding the herd that is the closest possible match to the distance drawn from the *pdf*. (The choice of recipient is also determined by its production type and a few other characteristics discussed in later sections.)

It is important to note that the *best possible* match based on distance is not necessarily a *good* match. Suppose, for example, that a distance of 10 km is drawn from the *pdf*, but that the nearest possible recipient herd is 150 km from the source. This herd will be selected as the recipient of contact, in spite of the fact that it is substantially farther away from the source herd than 10 km. It is best to think of distances drawn from contact distance *pdfs* as general guidelines, rather than strictly applied parameters.

> *Box 9-12. Model parameters and model validity*
>
> The way that *NAADSM* simulates contact distances offers a reminder that *NAADSM* parameters should be considered interrelated: contact distance parameters should be developed with a consideration of the region encompassed by the population of interest in mind to ensure that the approach used in the simulation is logical and valid. If the distance distribution used to represent contacts differs substantially in scale from the size of the region of interest, the results obtained may not be what are expected.

9.6.3.1.4. Movement restrictions

After the detection of disease (see Section 9.8), a common control measure would be the implementation of movement restrictions to reduce the number of contacts between herds, thereby reducing the possibility for disease spread. The contact rate required by *NAADSM* as described above is the initial, or unrestricted, baseline contact rate. Movement restrictions are modeled in *NAADSM* by the reduction over time of this baseline contact rate.

The effects of movement restrictions on contact rates are specified by relational functions (see Section 7.1.6). Figure 9-23 shows one such *rel* in the *rel* editor window. The *x* axis shows the number of days since disease was first detected in the population. The *y* axis shows the percent of baseline contacts that can occur at that time: 100% of the baseline indicates no reduction in contact rate, while 0% of baseline would indicate complete stoppage of contacts from units of the designated source production type to units of the designated recipient production type.

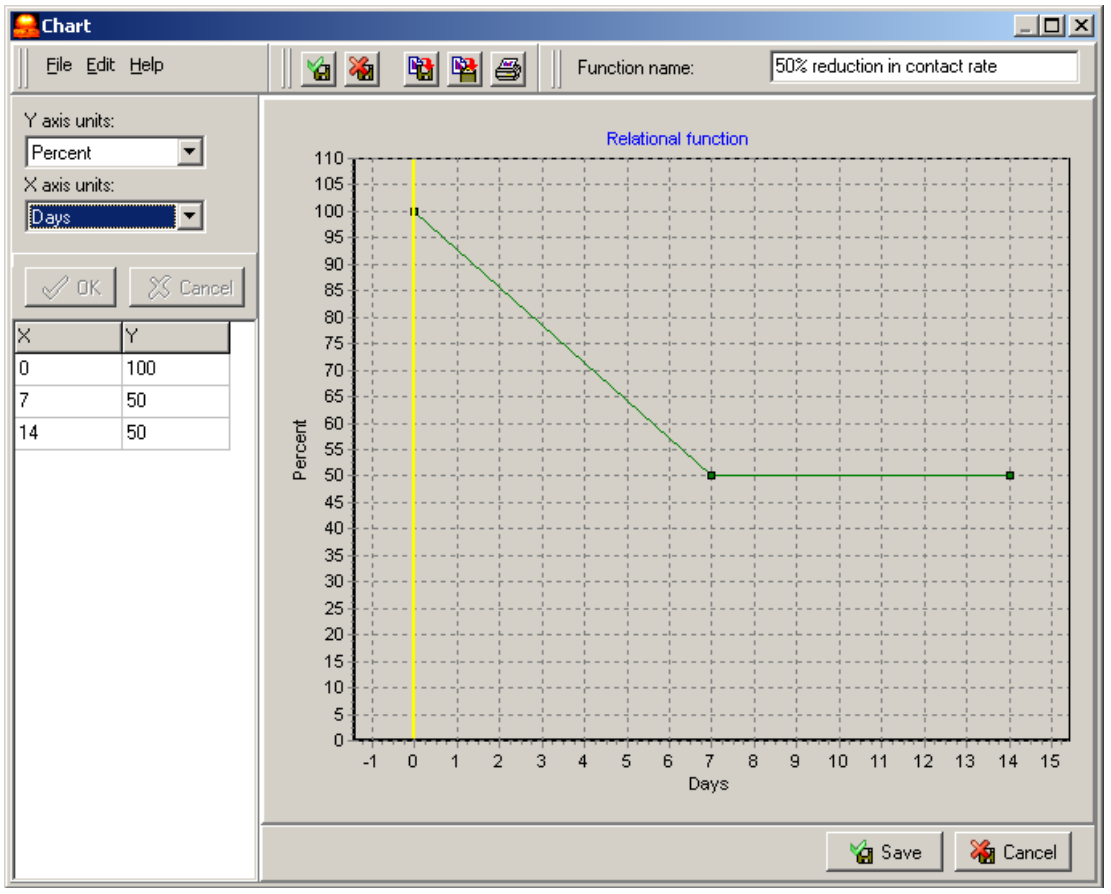


Figure 9-23. A relational function showing the reduction in contact rate over time as a result of movement restrictions.

The *x* axis shows the number of days since detection of disease in the population, with day 0 indicating the day on which disease was detected. The *y* axis shows the percentage of contacts that are still allowed to occur.

Suppose that, for contacts between units of two particular production types, the baseline contact rate is six units per day. Prior to disease detection, an average of six contacts would originate at each source herd of the indicated source production type. On the day that disease is detected (day 0 on the x axis), this baseline rate still applies.

Over time, however, the *rel* in Figure 9-23 shows that the average number of contacts will gradually decrease until the seventh day after disease detection, when the contact rate is 50% of the baseline rate. By the seventh day after disease detection, then, each source herd would generate on average 6 × 50% = 3 contacts per day. In the case illustrated here, there is no further reduction in contact rate: it remains steady at 50% for the duration of the simulation. Users may, of course, specify other functions with further reductions over time.

9.6.3.2. *NAADSM* parameters for contact spread

9.6.3.2.1. The Contact spread window

If contact spread was selected in the **Spread options** window (Section 9.6.2), the **Contact spread** window will be displayed (Figure 9-24). Use this window to modify parameters describing direct and indirect contact spread of disease for each production type combination.

Direct and indirect contact use the same kinds of parameters, but the values used may be different, and the two mechanisms operate independently from one another. The upper portion of the **Contact spread** window shows parameters for direct contact. Scrolling down in this window will reveal parameters for indirect contact.

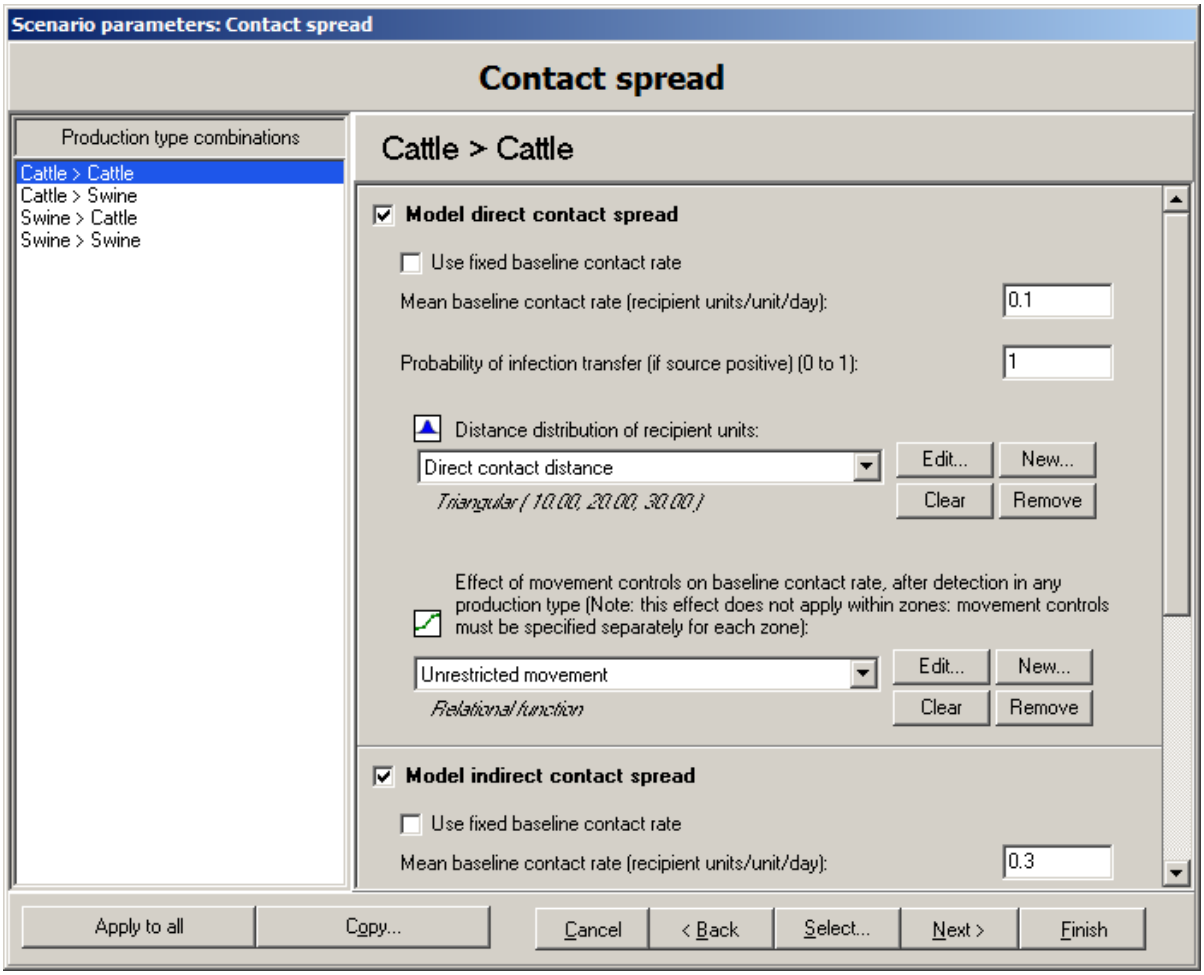


Figure 9-24. The **Contact spread** window, showing all parameters for direct contact.

Note that the probability of infection transfer parameter for direct contact will not be present if the option to use within-unit spread of disease is included: see Section 9.7.

Select a production type combination from the list on the left side of the **Contact spread** window (Figure 9-24), and then adjust parameters as described below.

9.6.3.2.2. 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.

Use fixed baseline contact rate: check this box if you wish to specify a fixed initial contact rate, as described in Section 9.6.3.1.1.

Mean baseline contact rate (recipient units/unit/day): enter the unrestricted contact rate, as described in Section 9.6.3.1. (If **Use fixed baseline contact rate** is checked, this option will be called **Fixed baseline 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 7.1.5).

Proportion of the unit included in a shipment: this parameter appears only when within-unit prevalence is used. See sections 9.5.2 and 9.7 for details.

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. As described in Section 9.6.3.2, this variable is a relational function (see Section 7.1.6) of time (in days) since disease was detected (see Section 8.7). Select or create a function with the *rel* selection box (see Section 7.1.6).

Box 9-13. Movement controls in zones

The movement restrictions specified here apply throughout the entire population of interest, and the entire region being modeled with *NAADSM*. For this reason, these restrictions are occasionally referred to as "global". For more localized control of movement, zones may be used (Section 9.12). As described in Section 9.12.1.1, movement restrictions used for zones completely override the global movement restrictions specified here. It is possible in *NAADSM* to allow more movement locally within zones than is allowed globally. Pay special attention to any movement restriction parameters that might be used to ensure that they are doing what you expect them to do.

9.6.3.2.3. 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 9.6.3.2). 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 9.6.3.2.1), with two exceptions. Latent units cannot spread disease by indirect contact, so this check box is not displayed. Also, there is no parameter for indirect contact that is analogous to the proportion of units included in a shipment.

9.6.4. Local-area spread

Local-area spread represents the non-directional spread of disease in the area surrounding an infected premises by any of a variety of mechanisms that might be difficult to directly characterize or trace. Such spread might be due to insects, pests, lapses in biosecurity, or localized aerosol transmission. Local-area spread might also encompass certain forms of contact that might technically be considered "direct" (for example, "across the fence" contact between animals on different farms), but which do not involve the physical introduction of animals from one unit into another (see Section 9.6.3).

9.6.4.1. Understanding local-area spread in *NAADSM*

Local-area spread can occur from an infectious source herd to susceptible recipients based on a particular probability of spread. The user specifies several parameters to establish a "baseline" probability of disease spread between two herds. Then, on each simulation day, the probability of spread is calculated for every pair of source and recipient herds, and *NAADSM* stochastically determines whether potential recipients will become infected or not. Mathematical formulas in the model then adjust this "baseline" probability to account for the different factors discussed below and in the *NAADSM* model specification (*NAADSM* Development Team, 2012).

The *disease states of the source unit and potential recipient units* determine whether disease spread by local-area transmission is even possible. Only infected source units that are subclinical or clinical can spread disease by this route: latent herds, by definition, are not shedding the disease agent, so it cannot be transmitted. As in all cases, potential recipient herds must be susceptible in order to be infected by local-area spread.

The *production types of the source and potential recipient units* can influence disease spread. Consider, for example, the case of foot-and-mouth disease (FMD): cattle are generally thought to be more prone to infection by localized aerosol spread (a mechanism that would be represented in *NAADSM* by local-area spread) of FMD virus than swine. Swine, however, produce and secrete higher amounts of FMD virus, and are thought to be more infectious. These species differences in susceptibility and infectiousness can be accounted for in *NAADSM*.

The *distance between a source and a recipient herd* affects the probability of spread by local-area transmission. If all other factors are equal, a susceptible potential recipient herd that is relatively close to an infectious source herd will have a higher probability of becoming infected than a potential recipient herd that is farther away. Probability of disease transmission continues to decrease with greater distance between source and potential recipient units.

The *sizes of the source and potential recipient herds* might also influence the probability of spread. Conceptually, larger herds can have more animals producing and expelling virus particles or other agents, and might be more infectious than smaller herds. Similarly, because larger herds have more animals, the likelihood of at least one animal becoming infected upon local-area exposure is higher.

Finally, *within-unit prevalence of disease* can affect the probability of spread. The argument is similar to that for herd size: herds with high prevalence of disease are presumed to

be more infectious than herds with low prevalence. The use of within-unit prevalence is discussed further in Section 9.7.

In order to parameterize local-area spread, the user is asked (conceptually, rather than literally) to identify and describe two arbitrary herds in the population. These two herds can be actual herds taken from the population file, or can be completely hypothetical, and are characterized by the user as follows:

- The arbitrarily selected *source* herd is described based on its production type and the number of subclinical and clinical animals in the herd.
- The *recipient* herd is described based on its production type, and the number of susceptible animals that it includes.
- The user is then asked for the distance (in kilometers) between these two arbitrarily selected units.
- Finally, the user is asked to define the probability that, on any particular day, the designated source herd will infect the designated recipient herd.

Based on the specified parameters for these two arbitrary or hypothetical herds, *NAADSM* will then extrapolate the probability of local-area disease transmission from an actual infected herd in the model and each surrounding potential recipient herd in a running simulation.

Figure 9-25 illustrates the application of some sample parameters to make calculations of the daily probabilities of disease spread between units at various distances. For a more thorough description of the mathematics behind this extrapolation, readers are referred to the *NAADSM* model description (*NAADSM* Development Team, 2012).

Box 9-14. How are parameters for local-area transmission obtained?

Although the parameters and mathematics that define local-area spread are conceptually straight-forward, finding values for these parameters is not an easy task. In situations where historic outbreak data exists, it may be possible to produce estimates of these parameters from existing information.

Most often, however, *NAADSM* users will be limited to the use of expert opinion for producing values for local-area spread parameters. In cases like this, it may be advisable to subject local-area spread parameters to sensitivity analysis, to determine the robustness of model-produced conclusions to changes in these parameters.

Members of the *NAADSM* Development Team have developed a *Microsoft Excel* spreadsheet to help users visualize the calculations performed by the modeling framework. Please see the technical paper by Reeves and Hupalo (2012), available at <*http://www.naadsm.org/techpapers*>, for a description of this spreadsheet.

The development of parameters for local-area spread is an ongoing area of research. Interested model users are invited to contact members of the *NAADSM* Development Team to share ideas and possible solutions.

Box 9-15. The number of infectious animals in the source herd for local-area spread

Although the value selected for the number of infectious animals in the source herd is somewhat arbitrary, it should also be somewhat realistic. In a simple model, this number will be the average number of infectious animals found in infectious herds of a particular production type, regardless of the length of time that herds have been infectious. Such a single, average value may be difficult to produce, and its use might be even more difficult to justify: the within-herd prevalence of disease is a dynamic property, and it would be more realistic to represent it as such. Further discussion of this point is presented in Section 9.7.

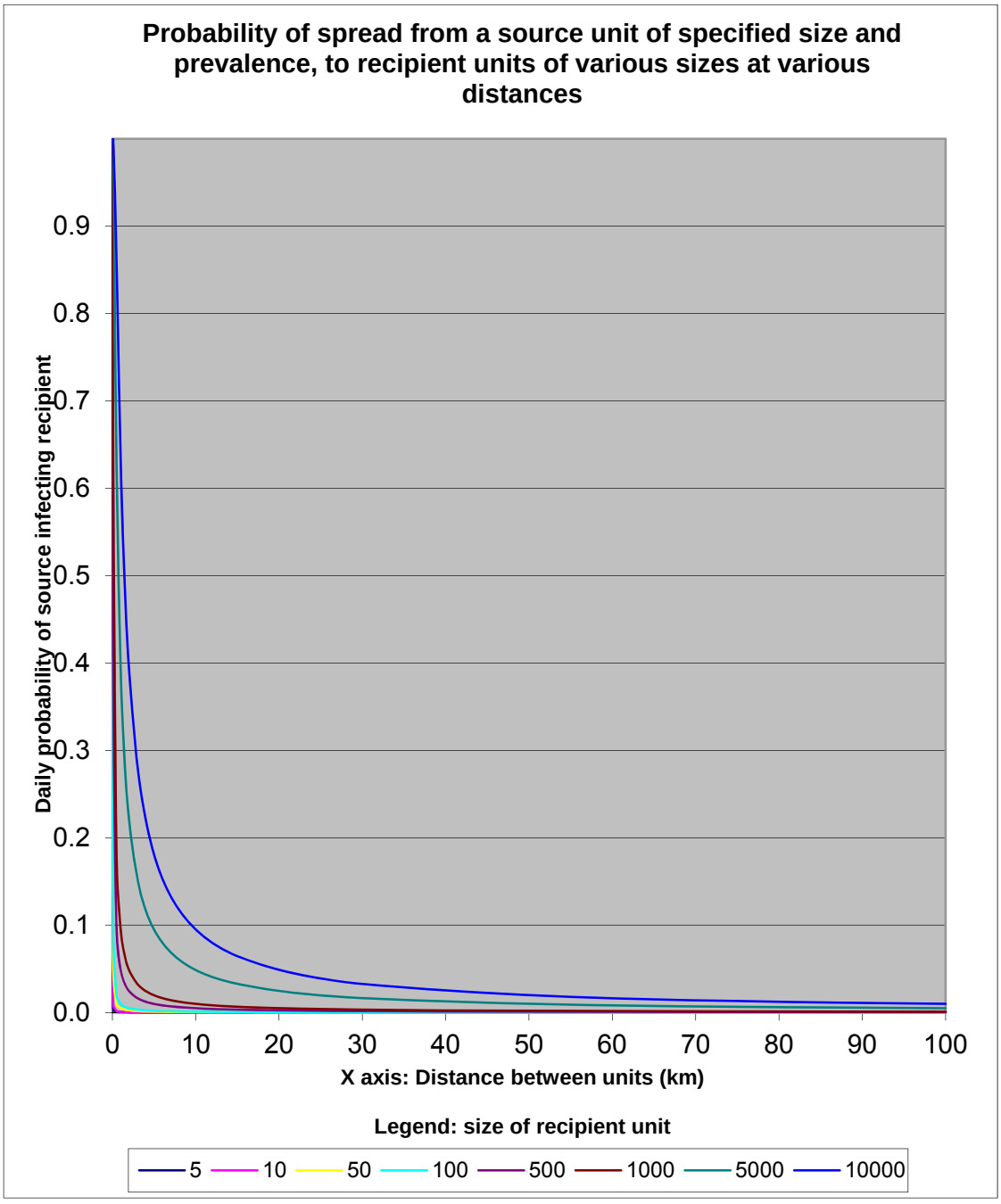


Figure 9-25. Daily probabilities of local-area spread, based on sample parameters.

The parameter values used to generate this plot were as follows: number of infectious animals in the source unit: 10; number of susceptible animals in the recipient unit: 100; distance between units: 1 km; daily probability of spread given these conditions: 0.0002 (*i.e.*, a 0.02% chance per day of disease transmission from the source to the recipient unit)

Based on these parameters, *NAADSM* extrapolated additional probabilities of disease spread to units at distances apart as shown on the x axis, and to recipient units of various sizes as shown in the legend. This plot was generated using the spreadsheet included with Reeves and Hupalo (2012).

9.6.4.2. Setting local-area spread parameters

In order to set local-area spread parameters, the **Local-area spread** option must first be selected from the **Spread options** window, as described in Section 9.6.2.1 (Figure 9-21).

Parameters for local-area spread are set for each combination of source and recipient production types, as shown in Figure 9-26. To simulate local area spread for a particular pair of source and recipient production types, check the box labeled **Model local area spread between these production types**.

The remaining input boxes on the screen are used to characterize two arbitrary herds, as described in the preceding section. Enter the number of infectious animals in the source unit, the number of susceptible animals in the recipient unit, and the distance between the two arbitrarily selected source units as shown. Finally, enter the daily probability of local area spread between the two units.

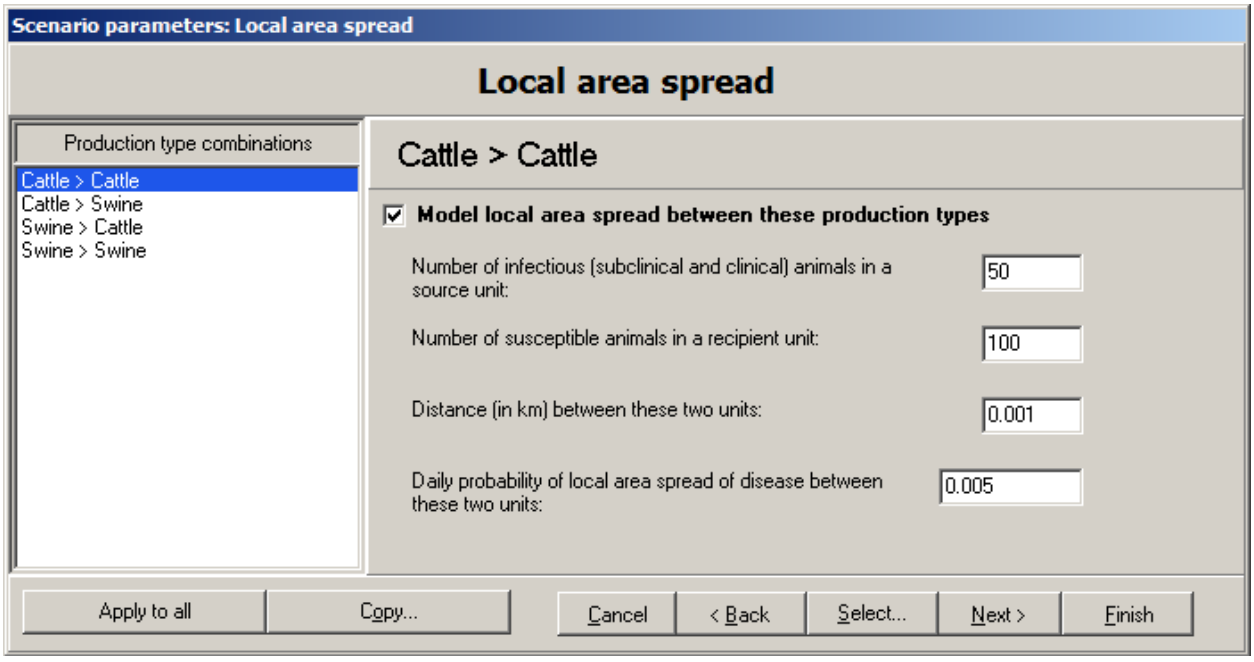


Figure 9-26. The **Local area spread** window.

9.6.5. Airborne spread

NAADSM offers the option of simulating airborne transmission of disease, *e.g.*, by long-distance dispersal of virus particles or other causal agents via the atmosphere. This transmission can be directional, to reflect the effect of predominant wind current directions. Recall that the area at risk of airborne transmission is set using the **Spread options** window, discussed in Section 9.6.2.1 (Figure 9-21).

The parameters and mathematics used to represent airborne spread in *NAADSM* 4 are the same as those used to represent local-area spread (Section 9.6.4). Parameter values may be different in order to reflect the differences among these two mechanisms, but the approach to setting airborne spread parameters is identical. The **Airborne spread** window (Figure 9-27) is likewise identical to the **Local-area spread** window (Figure 9-26), and is used the same way (Section 9.6.4.2).

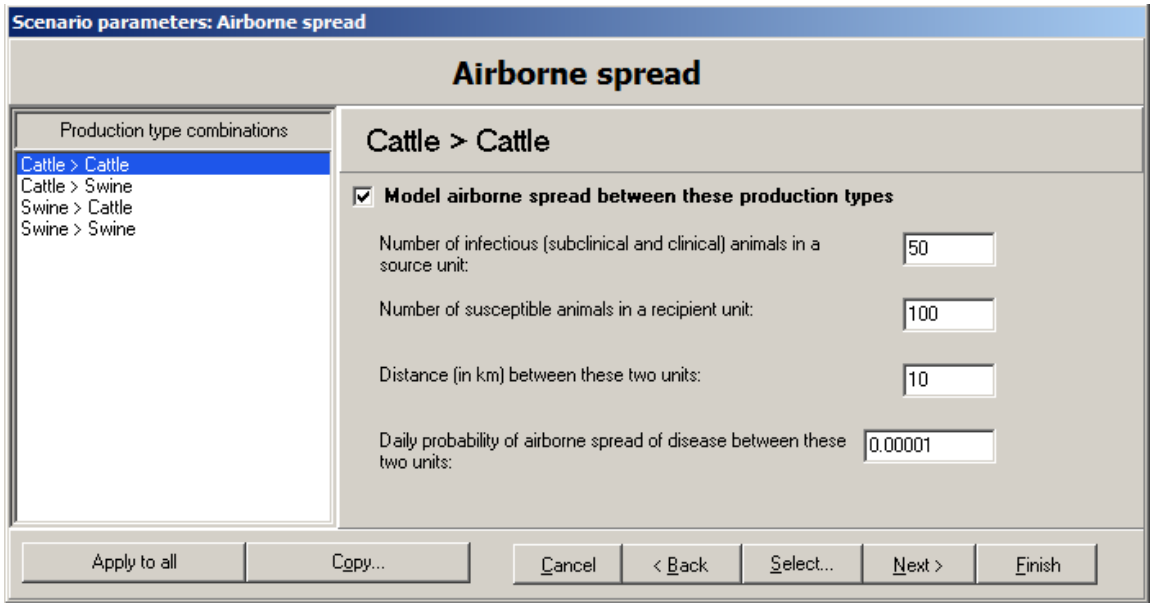


Figure 9-27. The **Airborne spread** window.

9.7. Within-unit prevalence of disease

9.7.1. Understanding within-unit prevalence in *NAADSM*

NAADSM 4.0 is a herd-based model: it does not attempt to simulate events that occur within herds, such as the transmission of disease from individual infected animals to susceptible animals within a herd. Several authors have suggested, however, that changes in within-herd prevalence over time should be considered when modeling spread of disease among herds, as changes in prevalence will affect the infectiousness of herds (*e.g.*, Carpenter *et al.*, 2004; Kostova-Vassilevska, 2004; Reeves *et al.*, submitted for publication). This concept was introduced in Section 9.5.2.

Although *NAADSM* does not simulate within-unit spread of disease, it does offer a mechanism by which the effects of changes in within-unit prevalence over time can be incorporated in a simulation, at least in a simplified way.

Within-unit prevalence in *NAADSM* currently affects three components: disease spread by direct contact, by local-area transmission, and by airborne transmission. As within-unit prevalence of disease increases, an infected herd will become increasingly infectious: that is, the probability of disease transmission from that infected herd will increase. Similarly, the probability of disease transmission from an infected source herd will decrease as within-unit prevalence decreases.

9.7.1.1. Within-unit prevalence and disease spread by direct contact

Recall from Section 9.6.3.1.2 and Figure 9-24 that one of the parameters for direct contact is the probability of infection transfer: this is the probability that a contact, if it occurs between an infectious source herd and a susceptible recipient herd, will result in infection of the recipient herd. If the within-unit prevalence option is not used in *NAADSM*, the user specifies a single value for this probability of infection transfer.

If within-unit prevalence is used, then *NAADSM* will automatically calculate a probability of disease spread by direct contact. This calculation is based on two factors:

- The prevalence of infected animals within the source herd
- The number of animals included in a shipment from the source herd to the recipient herd

Recall that the presence of only one infected animal is required for an entire herd to be treated as though it were infected (Section 9.5.1.1 and Table 9-4). By similar reasoning, transferring even one infected animal from a source to a recipient herd is sufficient to result in disease spread between the two herds. The higher the prevalence of infected animals in the source herd, the higher the probability that at least one infected animal will be shipped. Similarly, larger shipment sizes are associated with an increased probability that at least one infected animal will be included in a shipment.

Box 9-16. Shipment sizes and direct contact

Recall that *NAADSM* does not explicitly simulate the movement of animals from one herd to another (Box 9-9). Shipment size is used to provide a more realistic estimate of the probability of disease spread by direct contact, but does not affect the number of animals in either the source or the recipient units (*i.e.*, the number of animals in each herd does not change).

9.7.1.2. Within-unit prevalence, local-area spread, and airborne spread

Section 9.6.4.1 mentions that the number of infectious animals in a source herd influences the probability of disease spread by local-area and airborne transmission. Recall that, when local-area and airborne spread parameters are set, one of the characteristics of the arbitrarily chosen source herd that must be specified is the number of infectious animals in the herd. *NAADSM* then uses this number to extrapolate the probability of disease transmission by local-area or airborne spread between any two actual herds in the population. When within-unit prevalence is used, *NAADSM* will adjust the probability of disease transmission based on the actual number of infectious animals in the source herd. Although one arbitrarily chosen value is still required to establish the "baseline" situation, *NAADSM* can more dynamically and more realistically adjust this baseline to account for differences in within-unit prevalence.

Box 9-17. The number of infectious animals in the source herd, revisited

Box 9-15 notes that, if within-herd prevalence is not used, the number of infectious animals in the source herd should reflect an overall average prevalence of infectious animals expected throughout the subclinical and clinical states for the source production type, so that the daily probability of transmission reasonably reflects the average situation.

When within-herd prevalence is used, it might be easier to specify the expected daily probability of disease transmission by local-area or airborne spread, *assuming that the source herd is at its peak level of infectiousness*. This probability then will be automatically adjusted downward to reflect the actual prevalence of infectiousness in the source unit.

9.7.1.3. Within-herd prevalence for direct contact versus within-herd prevalence for local-area and airborne spread

Based on the preceding discussion (Sections 9.5.1, 9.6.3, 9.7.1.1, and 9.7.1.2), a subtle but potentially important observation can be made: the probability of disease spread by direct contact is influenced by the within-herd prevalence of all infected animals (*i.e.*, animals that are latent, subclinical, or clinical), but the probability of disease spread by local-area or airborne transmission is influenced by the within-herd prevalence of only animals that are shedding the infectious agent (*i.e.*, animals that are subclinical or clinical). If a latent animal from a source herd is introduced into a previously uninfected recipient herd, then that recipient herd is infected, according to the definitions presented in Table 9-4. By contrast, latent animals by definition

(Table 9-3) cannot contribute to local-area or airborne spread, because they are not shedding the agent.

For this reason, *NAADSM* 4 makes use of two related but separate within-herd prevalences: the within-herd prevalence of all infected animals (which influences direct contact), and the within-herd prevalence of subclinical and clinical (*i.e.*, shedding) animals (which influences local-area and airborne spread).

9.7.2. Setting within-unit prevalence

Within-unit prevalence of all infected animals as well as of shedding (subclinical and clinical) animals are specified as relational functions (Section 7.1.6). A sample within-unit prevalence curve for all infected animals is shown in Figure 9-28. The *y* axis of this relational function shows the percent of the herd that is infected, and the *x* axis shows time. The time scale used for within-unit prevalence curves is a bit peculiar, and is unique in *NAADSM* for reasons that will discussed below.

Within-unit prevalence is not a stochastic parameter in *NAADSM*: each production type can have only one within-unit prevalence curve that represents the "typical" or the "average" situation. The curve shown in Figure 9-28, for example, shows that a "typical" outbreak in a "typical" cattle herd (which would include that herd's latent, subclinical, and clinical stages) will last about 45 days, and that peak prevalence of infection of nearly 95% is reached on day 19.

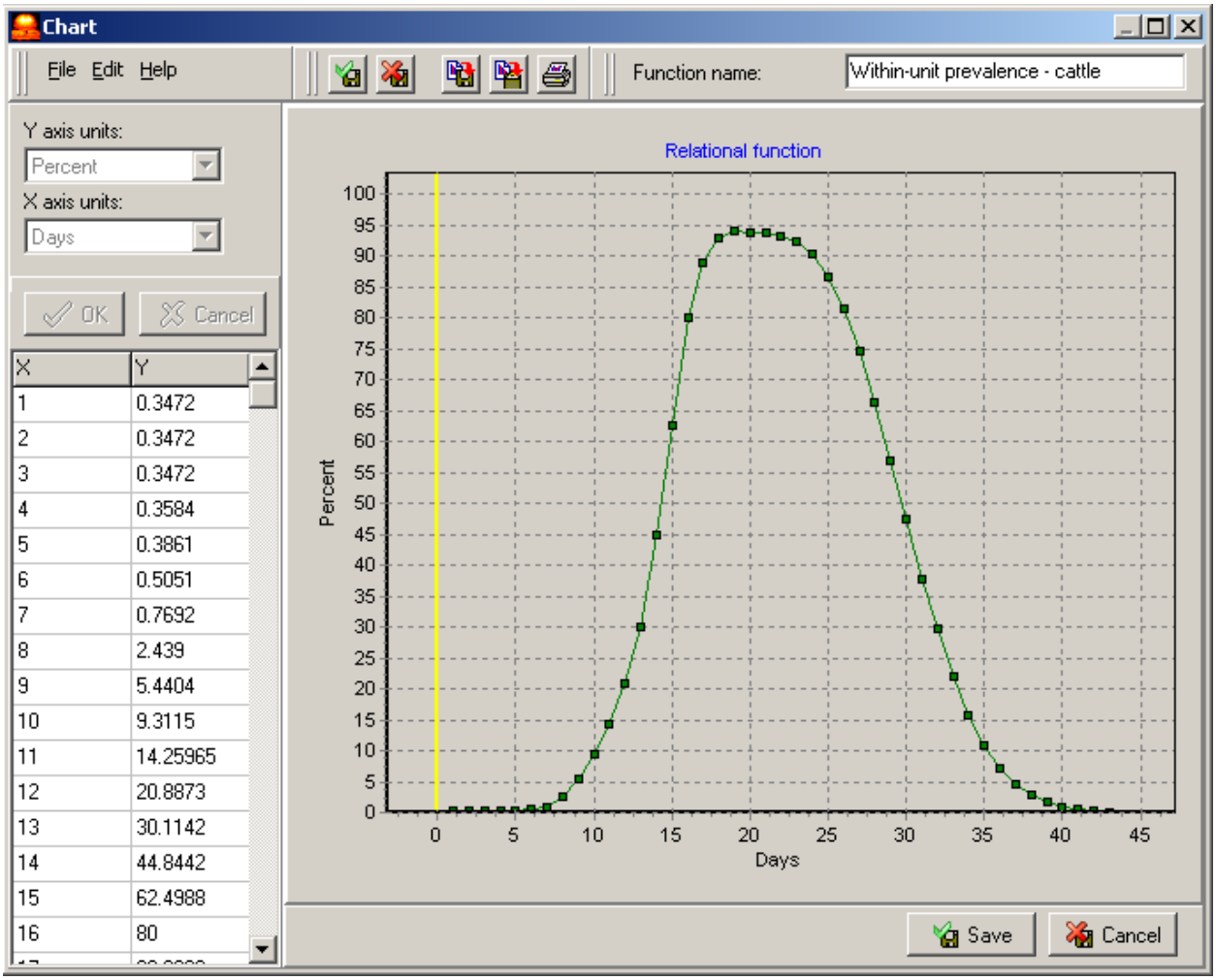


Figure 9-28. A relational function showing a sample within-unit prevalence curve.

Note that prevalence is expressed as a percentage (0 to 100%) of the herd that is infected on each day. Note also that the time scale of within-unit prevalence curves is used differently than other relational functions in *NAADSM*: see section 7.1.6.

Recall from Section 9.5, however, that the durations of each herd's individual disease states are determined stochastically: because the values for an individual herd's latent, subclinical, and clinical state durations are drawn from distributions, it is unlikely that the total duration of a herd's infected period will be exactly 45 days.

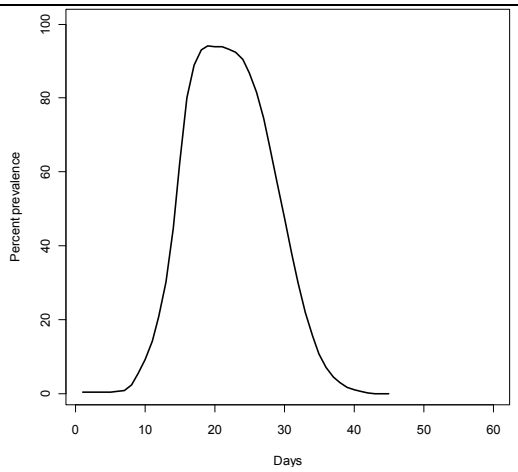
NAADSM deals with this situation by treating the time scale for within-unit prevalence curves as a *relative* scale: regardless of the number of days specified for a within-unit prevalence curve, *NAADSM* will dynamically adjust this duration to properly reflect the actual duration of each herd's infected period, as determined by the values drawn from the appropriate latent, subclinical, and clinical state duration distributions. Figure 9-29 illustrates how this adjustment is made. Note that *only* the duration of the within-unit prevalence curve is adjusted in *NAADSM*: the peak prevalence is not changed. Also note that within-unit prevalence curves are the *only*

relational functions used in *NAADSM* which treat time on a relative scale. In all other cases, the days represented in a relational function are used exactly as specified.

A) A herd with a 45-day infected period

Latent period: 5 days
Subclinical period: 8 days
Clinical period: 32 days

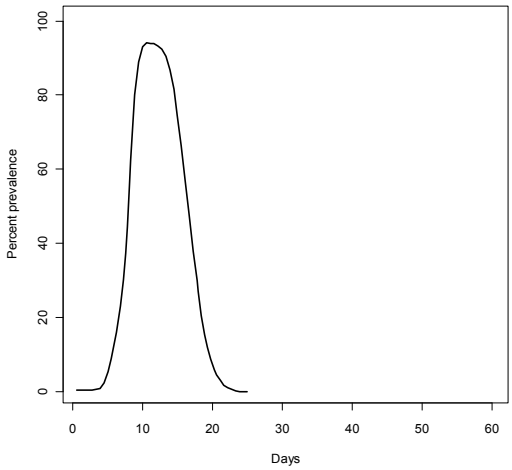
Peak prevalence of about 95% is reached on day 19, not quite at the half-way point for the infected period



B) A herd with a 25-day infected period

Latent period: 2 days
Subclinical period: 5 days
Clinical period: 18 days

Peak prevalence of about 95% is reached on day 11, not quite at the half-way point for the infected period.



C) A herd with a 60-day infected period

Latent period: 5 days
Subclinical period: 10 days
Clinical period: 45 days

Peak prevalence of about 95% is reached on day 25, not quite at the half-way point for the infected period.

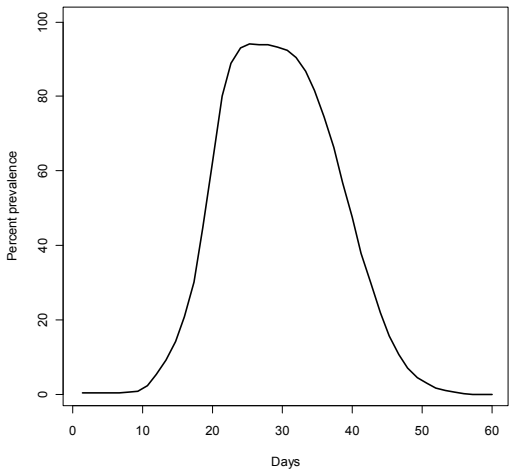


Figure 9-29. Application a within-unit prevalence curve to three herds with infected periods of different durations.

Note that peak prevalence is not changed, but time is treated on a relative scale.

Figure 9-17 above shows the **Disease options window**. This window is accessed by using the **Scenario parameters → Disease → Disease options** menu command. If you wish to include representations of within-unit prevalence in your models, choose the option labeled **Yes, use within-unit prevalence** from this screen.

As a result of this selection, the appearance of the **Disease** window will change. Compare Figure 9-18 above with Figure 9-30 below, and note the appearance of two new parameters near the bottom of the window. Specify the relational functions described above for the parameters labeled **Within-unit prevalence of infection (latent, subclinical, and clinical)** and **Within-unit prevalence of infectiousness (subclinical and clinical)**.

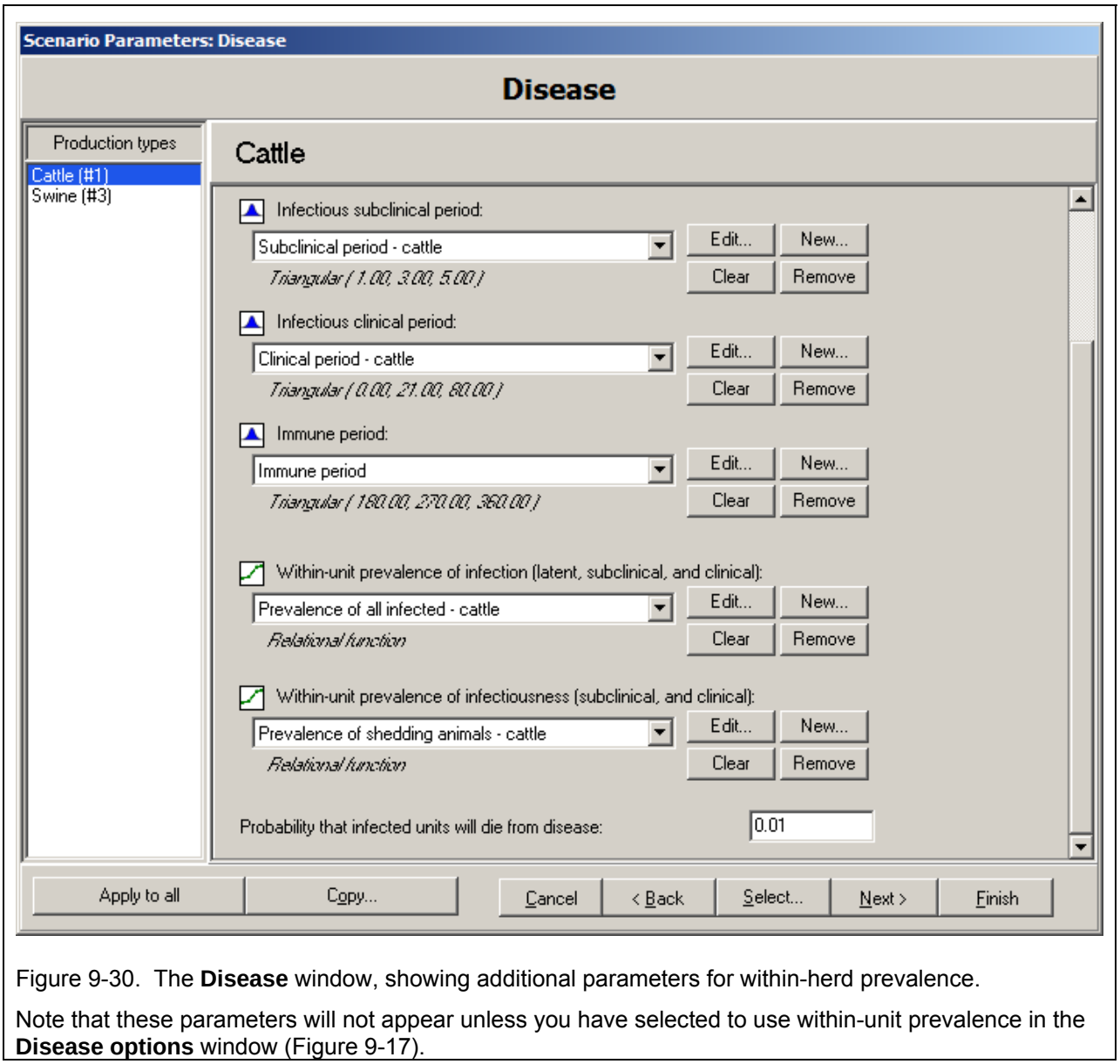


Figure 9-30. The **Disease** window, showing additional parameters for within-herd prevalence.

Note that these parameters will not appear unless you have selected to use within-unit prevalence in the **Disease options** window (Figure 9-17).

Box 9-18. How are parameters for within-herd prevalence obtained?

One method that has been used for *NAADSM*-based investigations to produce within-herd prevalence parameters involves the use of a separate model of within-herd disease spread to generate the required daily within-unit prevalence curves (see Patyk *et al.*, submitted for publication; and Reeves *et al.*, submitted for publication). A *NAADSM* technical paper, entitled *Developing unit-level disease state and within-unit prevalence parameters for NAADSM* and available at <*http://www.naadsm.org/techpapers*> describes this technique (Reeves, 2012).

9.7.2.1. Additional information for within-unit prevalence for direct contact

Section 9.7.1.1 mentions that the number of animals in a shipment also affects the probability of disease spread by direct contact. When the **Yes, use within-unit prevalence** option is selected from the **Disease options** window (Figure 9-17), a new parameter, labeled **Proportion of the unit included in a shipment**, is displayed in the **Contact spread** window (Figure 9-31). The number of animals included in a shipment is calculated based on the proportion of animals included in a shipment and the unit size.

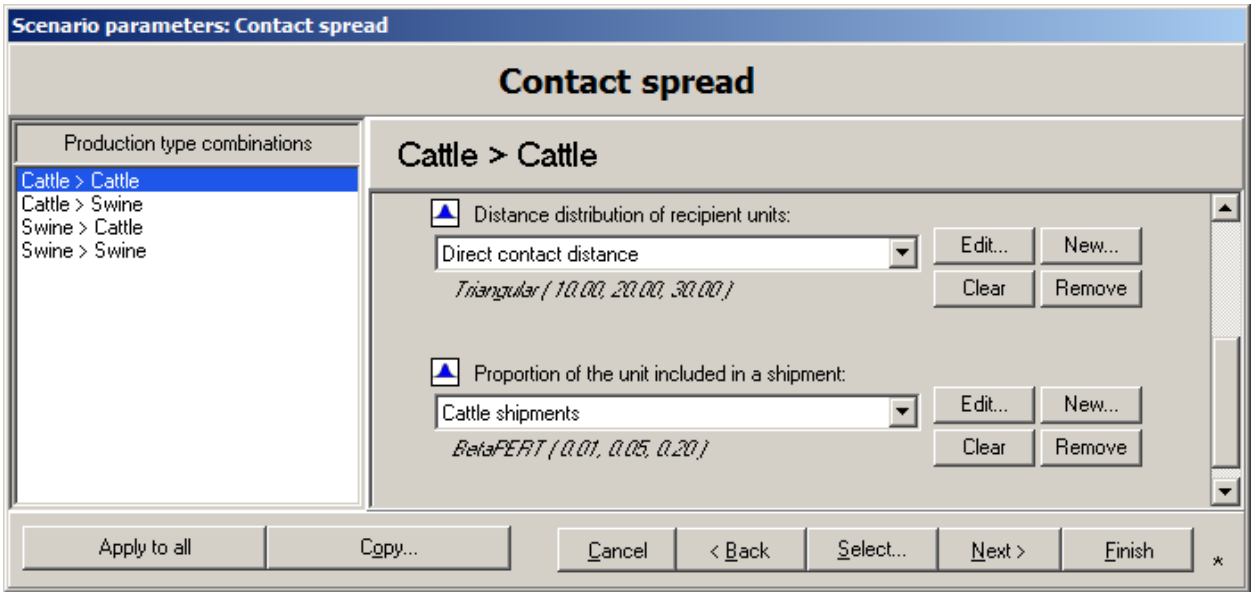


Figure 9-31. The **Contact spread** window, showing the **Proportion of the unit included in a shipment** option.

Note that this parameter will not appear unless you have selected to use within-unit prevalence in the **Disease options** window (Figure 9-17).

9.8. Modeling disease detection

9.8.1. Understanding disease detection in *NAADSM*

Disease detection in *NAADSM* refers to the identification of infected units based on the appearance of clinical signs or disease mortality, followed by reporting of such units so that further action can be taken. Once disease has been detected, various control measures, such as movement restriction (Section 9.6.3.1.4), destruction (Section 9.13) or vaccination (Section 9.14) 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 that clinical signs (or disease mortality) will be observed in an infected herd, and the probability that a herd will be reported once clinical signs (or disease mortality) have been observed. The overall chance of detection is the product of these two probabilities.

Both probabilities are given as relational functions (see Section 7.1.6), to account for changes to the probability of observing signs and the probability of reporting over time during the course of an outbreak. For example, the probability of observing clinical signs in an infected herd 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 9-32 demonstrates the interaction of these two relational functions.

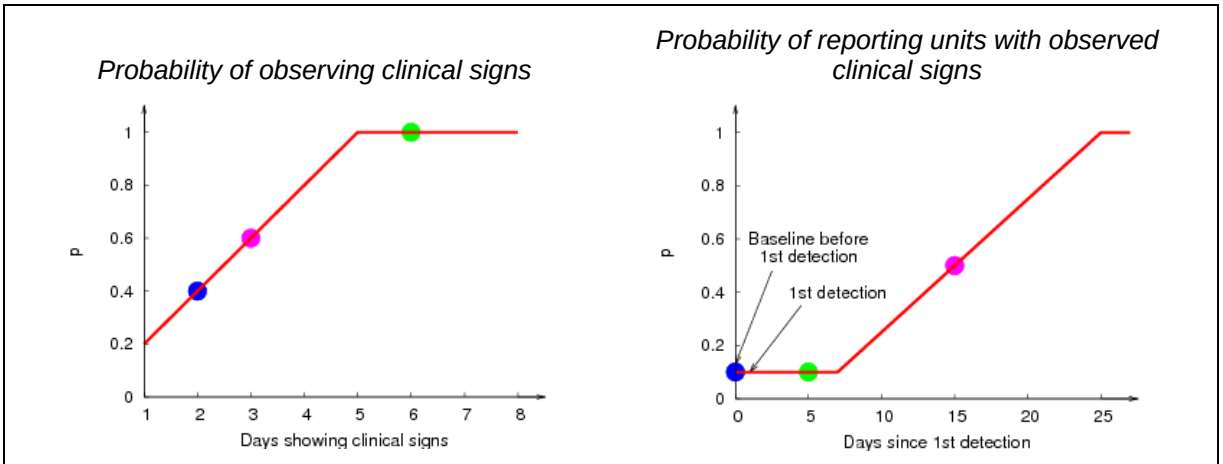


Figure 9-32. Calculating the overall probability of detection based on clinical signs.

This is based on the probability of observing clinical signs and the probability of reporting a detected unit.

- Before 1st detection, 2nd day of clinical signs, overall probability = 0.4 × 0.1 = 0.04
- 5 days since 1st detection, 6th day of clinical signs, overall probability = 1 × 0.1 = 0.1
- 15 days since 1st detection, 3rd day of clinical signs, overall probability = 0.6 × 0.5 = 0.3

A similar calculation is made for detection based on disease mortality.

Note that these two relational functions use different time scales. The probability of observing clinical signs is based on an individual herd: this function begins on the first day of a herd's clinical period. This function is applied independently to each infected herd. The probability of reporting a unit with observed signs is based on what is known about disease in the population. This function begins on the first day that disease has been detected in any herd in the population. Figure 9-32 shows a baseline probability of reporting disease, which will apply until at least one unit has been successfully detected. After that initial event occurs in the population, the probability of reporting can change.

Box 9-19. Baseline probability of reporting disease

The probability of reporting units with observed clinical signs is based on the number of days that have passed since the first detection of an infected unit in the population: for example, Figure 9-32 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%.

Box 9-20. Within-unit prevalence of disease and disease detection

In reality, the prevalence of disease within a herd influences disease detection: the greater the prevalence of disease, the greater the probability of detecting it. Although *NAADSM* does not use the within-unit prevalence parameter described in Section 9.7 to influence disease detection, the curve that represents the probability of observing clinical signs could be established to reflect the effect of within-unit prevalence on disease detection.

A more detailed model of disease detection, which accounts not only for within-unit prevalence but also different sampling strategies and diagnostic techniques, is currently under development for a future version of *NAADSM*.

Because the probabilities of observing and reporting disease mortality may differ from the probabilities of observing and reporting clinical signs of disease, these parameters are specified independently. Thus, *NAADSM* uses two pairs of relational functions for simulating disease detection: one pair for observing and reporting clinical signs, and a second pair for observing and reporting disease mortality. It is the responsibility of the user to specify how these function might differ, and to provide appropriate justification for any differences.

9.8.2. 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 9-33). (Depending on your selection in the **Detection options** window, the **Production type settings for detection** menu command may be disabled.)

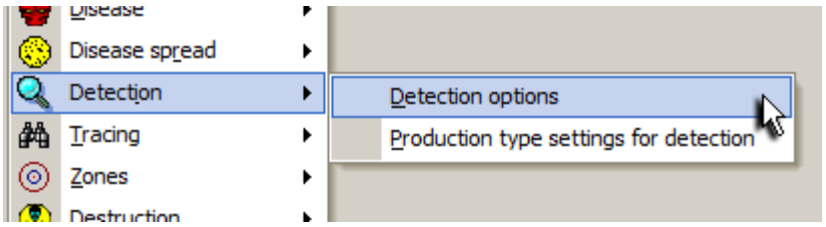


Figure 9-33. Commands of the **Scenario parameters→Detection** menu.

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 9-34).

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

Probability of observing clinical signs, given the number of days that a unit has been infectious: see Section 9.8.1. The variable is a relational function (see Section 7.1.6) of time (in days) that a unit has been infectious. Select or create a function with the *rel* selection box (see Section 7.1.6).

Probability of observing death from disease, given the number of days that a unit has been dead from disease: this variable is a relational function (see Section 7.1.6) of time (in days) that a unit has been in the dead from disease state. Select or create a function with the *rel* selection box.

Probability of reporting a unit with observed clinical signs, given the number of days since disease was first detected in any unit: see Section 9.8.2. The variable is a relational function (see Section 7.1.6) of time (in days) since disease was detected. Select or create a function with the *rel* selection box (see Section 7.1.6).

Probability of reporting a unit that is dead from disease, given the number of days since disease was first detected in any unit: this variable is a relational function (see Section 7.1.6) of time (in days) since disease was detected. Select or create a function with the *rel* selection box.

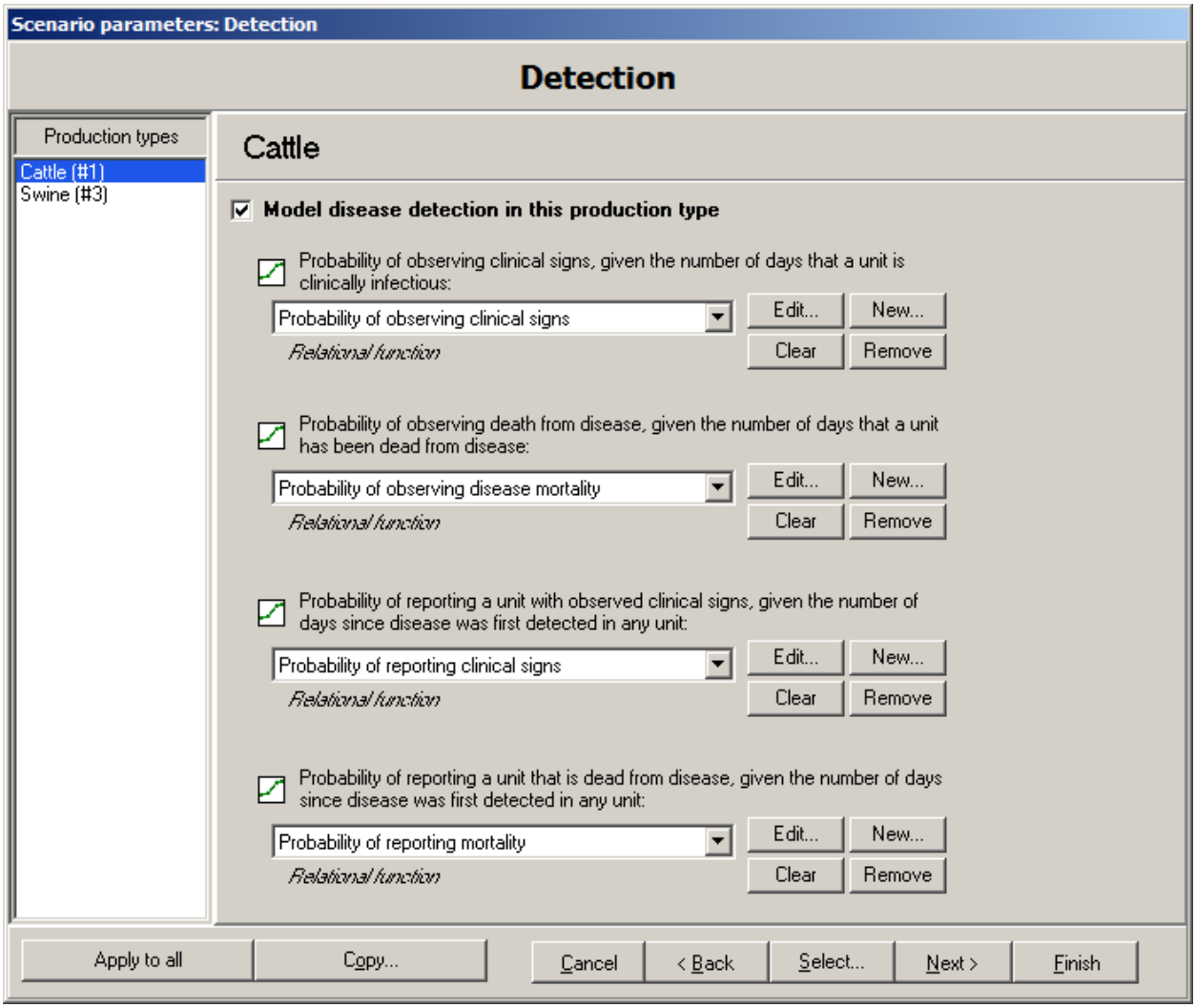


Figure 9-34. The **Detection** window.

9.9. Modeling tracing

9.9.1. Understanding tracing in *NAADSM*

Tracing refers to the process of identifying units (herds) at high risk for disease based on contact with detected units. Direct contacts and indirect contacts (section 9.6.3) to and from detected units may be traced. *Tracing forward* (also referred to as *tracing out*) is the process of determining which units received a contact from, and might have been infected by, a detected unit. *Tracing back* (also called *tracing in*) is used to determine which units were sources of contact (and potential sources of infection) for the detected unit.

Each attempt to trace a contact in *NAADSM* involves two units: the *origin* of a trace is the detected herd from which a trace investigation starts. The unit *identified by* a trace is the unit to which the trace leads. In the case of a tracing forward, the origin of a trace was the *source* of the contact. In the case of tracing back, the origin of a trace was the *recipient* of a contact. Figure 9-35 and Table 9-6 illustrate each of these concepts. In this figure, unit 1 has been detected based on clinical signs. Prior to detection, this unit had been the recipient of two contacts (from units 2 and 3, which are sources of contact with unit 1) and was also the source of two additional contacts (to units 4 and 5, which are recipients of contact with unit 1). Upon detection of unit 1, these contacts can be traced.

In *NAADSM*, each trace proceeds in a single step forward or back: for example, detection of unit 1 in the situation shown in Figure 9-35 does not automatically lead to further tracing from unit 3 (which was previously infected) or from units 4 or 5 (which are currently infected). Further tracing can occur from these units only if they are detected, as described below.

9.9.1.1. Consequences of tracing in *NAADSM*

In *NAADSM*, units identified by tracing will be automatically quarantined and thus can no longer spread disease by direct contact. Quarantined units, however, can still contribute to dissemination of disease by indirect contact and airborne spread. Optionally, units identified by tracing may be queued for destruction (see Section 9.13).

Tracing can contribute to detection of disease. This can occur in two ways. After a successful trace, the unit identified by tracing can be examined for clinical signs of disease. If the identified unit is clinical, it can be detected by this mechanism. Second, if the unit identified by tracing is infected (*i.e.*, it is in the latent, subclinical, or clinical disease states) or is naturally immune, then it can be detected by diagnostic testing. If a unit is detected by either a herd exam or a diagnostic test, then additional tracing can occur from the newly detected unit. Herd exams and diagnostic testing are described in sections 9.10 and 9.11, respectively.

Box 9-21. Tracing and disease detection

Users of versions of *NAADSM* prior to version 3.2 will recognize that the relationship between tracing and disease detection is a new capability, originally introduced in *NAADSM* 3.2 and carried over into *NAADSM* 4.0. Prior to this version, tracing and detection were completely independent of one another.

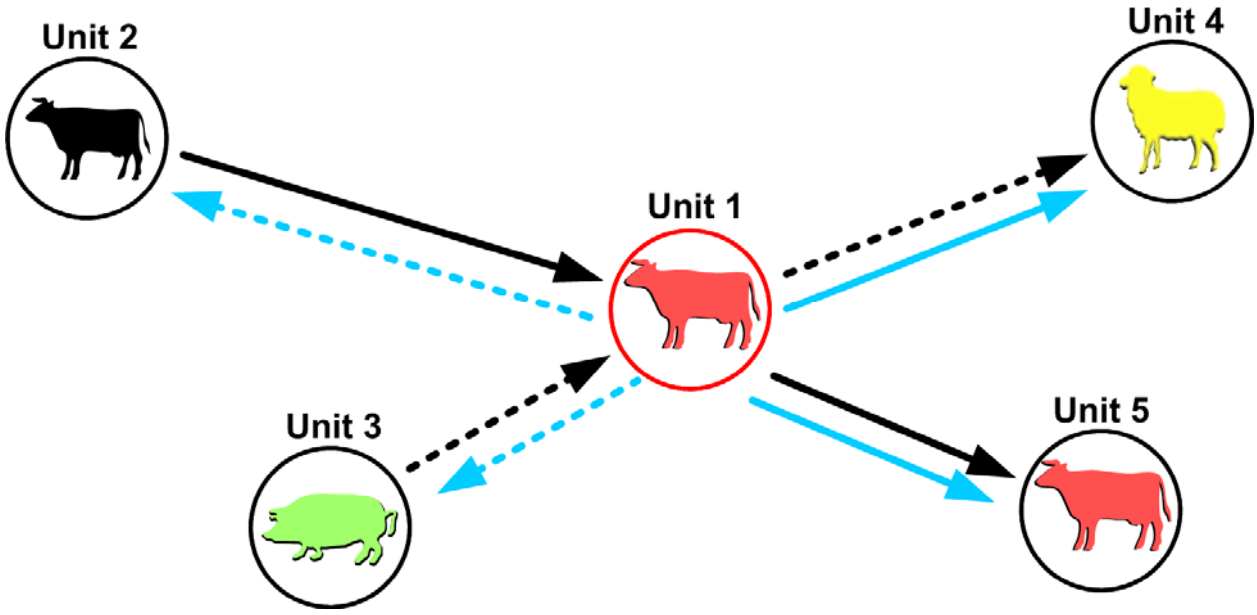


Figure 9-35. Contacts and tracing.

See text and Table 9-6 for a description of the depicted events.
Black lines indicate contacts (solid = direct contact, dashed = indirect contact). Blue lines indicate trace investigations (solid = trace-forward, dashed = trace-back). Color indicates disease state (black = susceptible, yellow = latent, red = clinical, green = naturally immune).

Table 9-6. Summary of events depicted in Figure 9-35.

Unit ID	***Disease state***	***Contacts***	***Tracing***
1	Clinically infectious Detected based on clinical signs	*Recipient* of direct contact from unit 2 *Recipient* of indirect contact from unit 3 *Source* of indirect contact with unit 4 *Source* of direct contact with unit 5	*Origin* of all tracing events
2	Uninfected, and susceptible to disease	*Source* of direct contact with unit 1	Could be *identified by tracing back* of direct contact from unit 1
3	Naturally immune Not detected	*Source* of indirect contact with unit 1	Could be *identified by tracing back* of indirect contact from unit 1
4	Latent	*Recipient* of indirect contact from unit 1	Could be *identified by tracing forward* of indirect contact from unit 1
5	Clinically infectious Not yet detected	*Recipient* of direct contact from unit 1	Could be *identified by tracing forward* of direct contact from unit 1

9.9.2. Setting global tracing options

Parameters for tracing are closely related to those for herd exams and diagnostic tests (described in sections 9.10 and 9.11, respectively). The commands of the **Scenario parameters → Tracing** menu (Figure 9-36) are used to set these parameters.

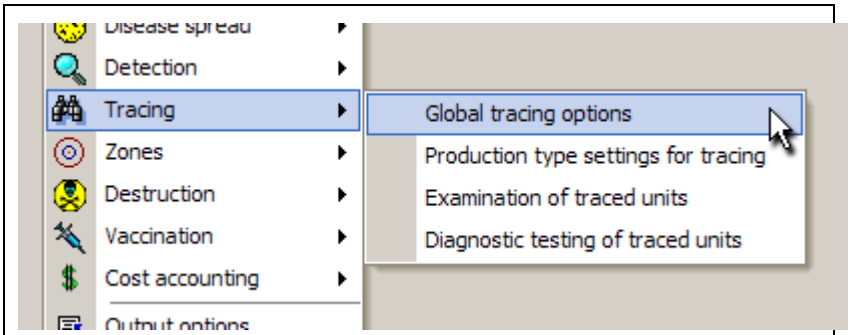


Figure 9-36. Commands of the **Scenario parameters→ Tracing** menu.

The **Global tracing options** window (Figure 9-37) is used to select whether to use tracing for any production types included in a model. If tracing is used, then herd exams can be selected. If tracing and herd exams are conducted, then diagnostic testing can also be selected.

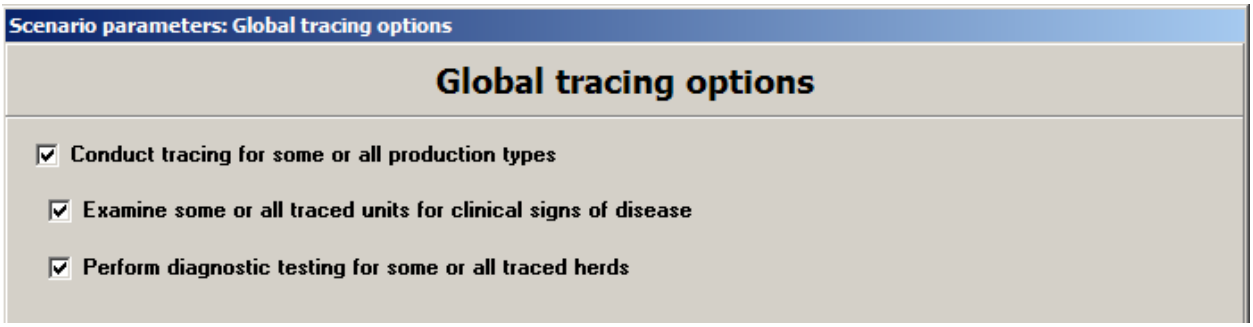


Figure 9-37. Options of the **Global tracing options** window.

If **Conduct tracing for some or all production types** is selected, then the option to **Examine some or all traced units for clinical signs of disease** is enabled, and can be selected. Similarly, if this option is selected, then **Perform diagnostic testing for some or all traced herds** is enabled, and can be selected.

9.9.3. Setting tracing parameters

Options for tracing are specified in the **Tracing** window (Figure 9-38). Users can select to carry out tracing forward and/or tracing back for direct and/or indirect contacts.

The so-called "critical period" is the period of time (given in days) prior to detection of the origin unit of the trace, for which contacts should be investigated. Consider, for example, a critical period of 28 days. If a unit is detected on simulation day 48, then every contact that involved that unit from day 20 to day 47 (*i.e.*, during the 28 days prior to detection) can be traced.

The probability of trace success indicates the likelihood that an attempt to conduct a trace will be successful. For example, if 10 contacts occur within the critical period and the probability of trace success is 0.9, then an average of 9 of those contacts will be successfully traced.

The tracing delay is the number of days that it takes to complete a trace investigation. This parameter can be given as a probability density function (section 7.1.5), to account for the variability in the time it takes to carry out tracing.

Tracing parameters are production-type specific. Note that the probability of a trace investigation succeeding depends on the production type of the unit where the trace originates. Suppose a unit of production type A ships to a unit of production type B. If the A unit is then detected, the probability that trace out will find the B unit depends on the probability set for production type A. If the B unit is detected instead, the probability that trace in will find the A unit depends on the probability set for production type B, which may be different.

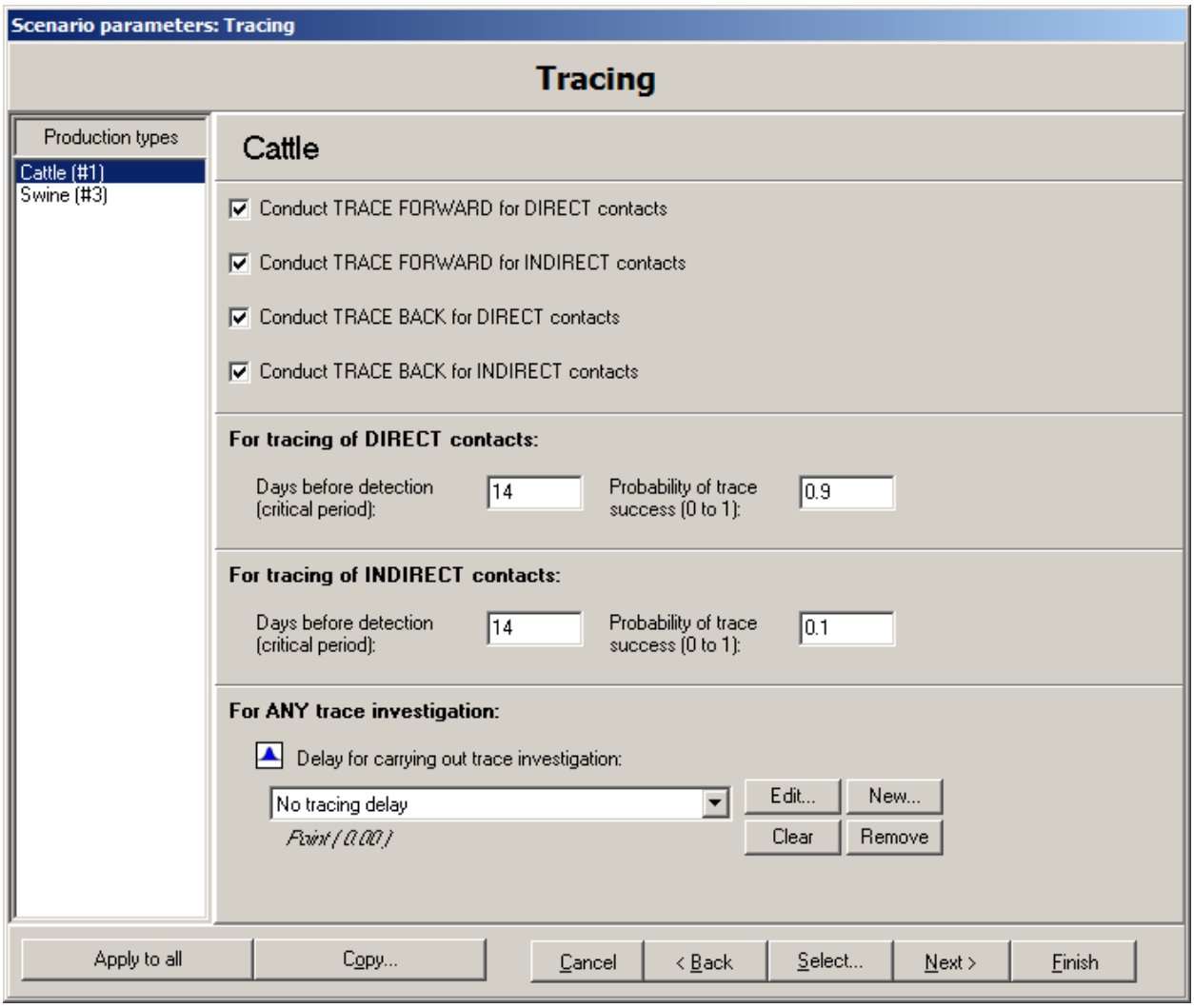


Figure 9-38. The **Tracing** production type parameter window.

9.10. Modeling herd exams

9.10.1. Understanding herd exams

Recall from section 9.8 that detection typically involves the observation of clinical signs of disease or disease mortality, as well as the reporting of those signs to appropriate authorities so that further action can be taken. The process of detection described in section 9.8 can be thought of as a somewhat passive process, which can become more active, as awareness of disease in a region increases.

Herd exams, by contrast, are active processes. A herd exam can be thought of as an inspection by trained experts of animals on a premises that has been identified by a trace examination. If animals in the herd are showing clinical signs of disease or disease mortality, those signs can be detected. Trained personnel are more likely to observe signs than passive observers. The key parameter for herd exams is a multiplier that describes how much more likely a trained observer is to detect signs compared to more passive observers. Suppose, for example, that the value of this multiplier is 1.5. If a passive observer has a 40% chance of observing signs of disease in a herd on a given day, then a trained investigator conducting a herd exam will have a 60% chance of noticing those signs.

Because herd exams are carried out by official personnel, *NAADSM* assumes that reporting of observed clinical signs or disease mortality is guaranteed: that is, if clinical signs or disease mortality are detected by a herd exam, then the probability that they will be reported is assumed to be 1. This term, then, does not need to be considered in calculating the overall probability of observation and reporting after a herd exam.

9.10.2. Setting herd exam parameters

Herd exam parameters are specified in the **Unit examination for clinical signs** window (Figure 9-39), which is accessed using the menu command **Scenario parameters → Tracing -> Examination of traced units** (Figure 9-36 above). (Recall that the **Examine some or all traced units for clinical signs of disease** box discussed in section 9.9.2 must be selected for this option to be available.) This window is used to select the types of tracing investigations to be followed up with herd exams, and the multiplier for disease detection, described in section 9.10.1.

Parameters for herd exams are production-type-specific: users can independently select which production types to conduct herd exams for, and can specify separate multipliers to account for species differences in the appearance of clinical signs and mortality.

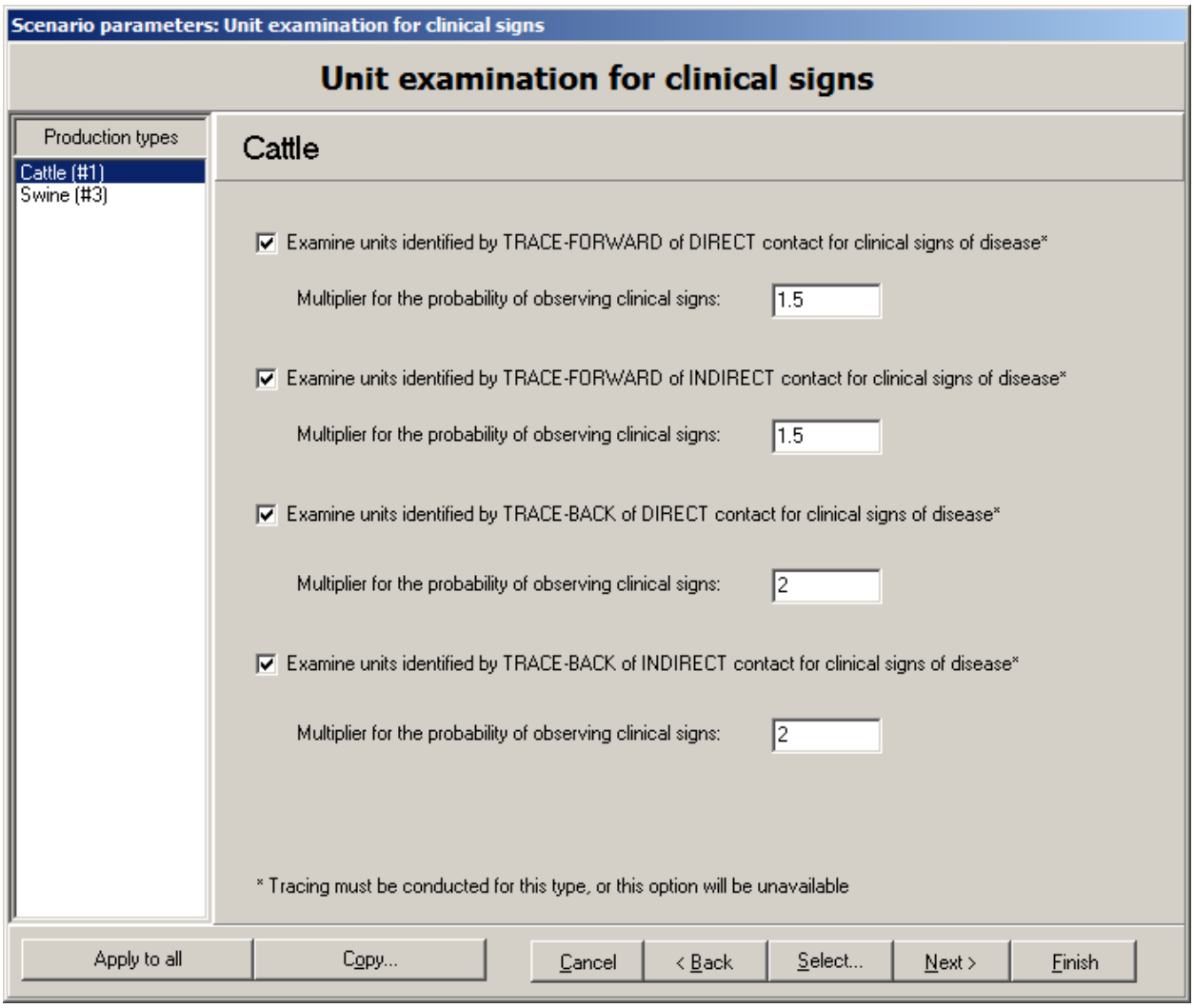


Figure 9-39. The **Unit examination for clinical signs** window.

Box 9-22. Detecting units in the dead from disease state after a herd exam

Although the captions in the *NAADSM* application discussed in this section mention observation of clinical signs, they also apply to units that are in the dead from disease state. It is appropriate to think of disease-induced mortality as an observable clinical sign of disease in a unit.

9.11. Modeling diagnostic tests

9.11.1. Understanding diagnostic testing in *NAADSM*

In most cases in *NAADSM*, herds can be detected only while they are showing clinical signs or disease mortality (sections 9.8 and 9.10). Diagnostic testing after a trace investigation provides a way to detect units in other infected states, or even units that were previously infected and have become naturally immune.

After a trace investigation, a herd exam (section 9.10) offers the first opportunity to detect disease. If the herd exam does not detect disease (either because the unit examined was not showing signs, or because, simply by chance, signs were not detected), then a diagnostic test can be administered.

A diagnostic test is characterized by its *herd-level* sensitivity and specificity. The sensitivity and specificity of the diagnostic test determine the likelihood that the outcome is true positive, true negative, false positive, or false negative. A perfect test (that is, a test that always correctly identifies the disease status of a herd) will have sensitivity and specificity of 1 (100%).

The higher a test's sensitivity specificity are, the more likely the test is to accurately identify both infected and uninfected herds. A test with high sensitivity and low specificity will accurately identify most infected herds (*i.e.*, will produce true positive test results), but will also tend to produce many false positive test results (*i.e.*, herds which are not infected will test positive for disease). A test with low sensitivity and high specificity will produce many false negative results (because the test is not sensitive enough to correctly identify all truly infected herds), but will not produce many false positive results (the true disease status of most uninfected herds will be correctly identified). Finally, a test with low sensitivity and low specificity will produce the most incorrect results (both false positive and false negatives).

In *NAADSM*, herd-level sensitivity and specificity are not influenced by a unit's disease state or by within-unit prevalence of disease: a latent unit with low prevalence of infection will be treated in exactly the same way as a clinical unit with a very high prevalence of shedding animals. Table 9-7 shows the possible results of diagnostic testing, depending on the actual disease state of the unit tested.

A unit that tests positive, regardless of its actual disease state, will be treated as a detected unit. Detection can result in further tracing (section 9.9), or can trigger zones, destruction, or vaccination (sections 9.12, 9.13, and 9.14). If a unit tests negative, regardless of its actual disease state, no further action will be taken.

Table 9-7. Possible diagnostic test results and consequences for units in different disease states

	Diagnostic test result	
Actual disease state	*Test-positive*	*Test-negative*
Susceptible	False positive test resulting in detection	True negative test
Latent	True positive test resulting in detection	False negative test
Subclinical	True positive test resulting in detection	False negative test
Clinical	True positive test resulting in detection	False negative test
Naturally immune	True positive test resulting in detection	False negative test
Dead from disease	True positive test resulting in detection	False negative test
Vaccine immune	False positive test resulting in detection	True negative test
Destroyed	*Units that have been destroyed are not tested.*	

9.11.2. Setting diagnostic testing parameters

If the **Perform diagnostic testing for some or all traced herds** box is checked in the **Global tracing options** window (section 9.9.2), then the **Diagnostic testing of traced units** menu command and **Diagnostic testing** wizard screen will be enabled (Figure 9-40).

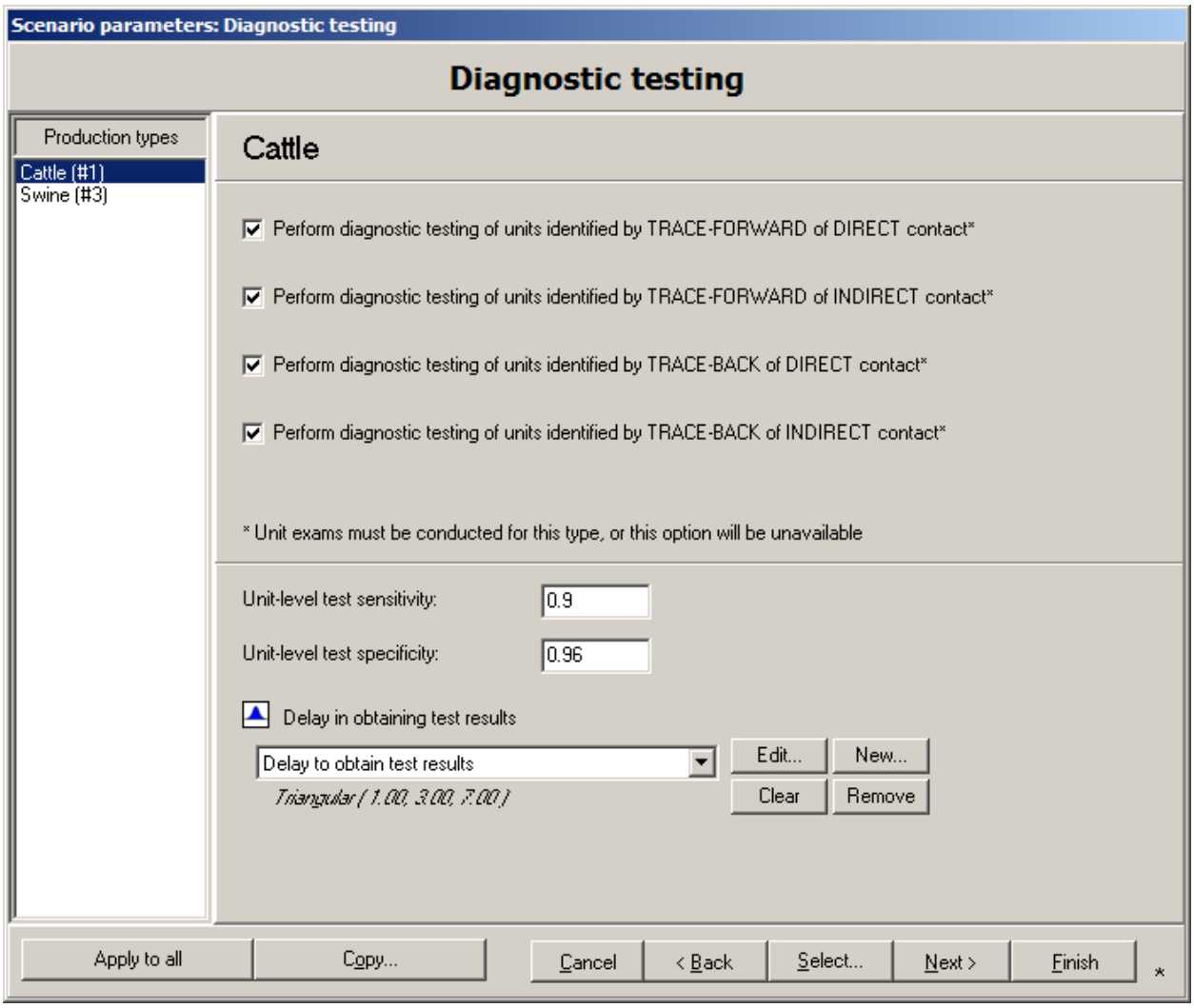


Figure 9-40. The **Diagnostic Testing** window.

On this screen, each type of tracing has a checkbox to enable performing diagnostic testing for the disease in the selected production type after different types of tracing.

Test sensitivity is the probability that the test will correctly identify an infected or naturally immune unit. Enter a value between 0 and 1.

Test specificity is the probability that the test will correctly identify an uninfected unit. Enter a value between 0 and 1.

The **Delay in obtaining test results** determines how much time it will take (in days) to obtain the results of a diagnostic test. This parameter is a probability density function (section 7.1.5). When test results are delayed, the results will be based on the disease state of the unit on the day the test occurred, not the day the results become available.

9.12. Modeling Zones

9.12.1. Understanding zones

Zones are used in *NAADSM* to simulate greater movement control and enhanced surveillance in areas around detected units and/or in areas around units that have been identified by tracing. A basic zone consists of a circle, formed around a detected unit, which will generally include other units in the vicinity. Figure 9-41 shows a unit map from a *NAADSM* simulation. Note that several units are surrounded by dark red circles. These circles represent the area encompassed by zones. Near the center of the figure is an example of a basic zone (A): it was formed around a clinical unit when that unit was detected, and encompasses several other units. (Zones can also be formed around units that have been identified by tracing.) Below and to the left of this are three more overlapping circles (B). These indicate areas where zones were formed, again centered around detected units. Because these circular areas overlapped, they were merged to give a single continuous zone region.

Note that, although zones are circular and hence superficially similar to destruction and vaccination rings (described in sections 9.13 and 9.14, respectively), zones, destruction rings, and vaccination rings are all implemented independently in *NAADSM*.

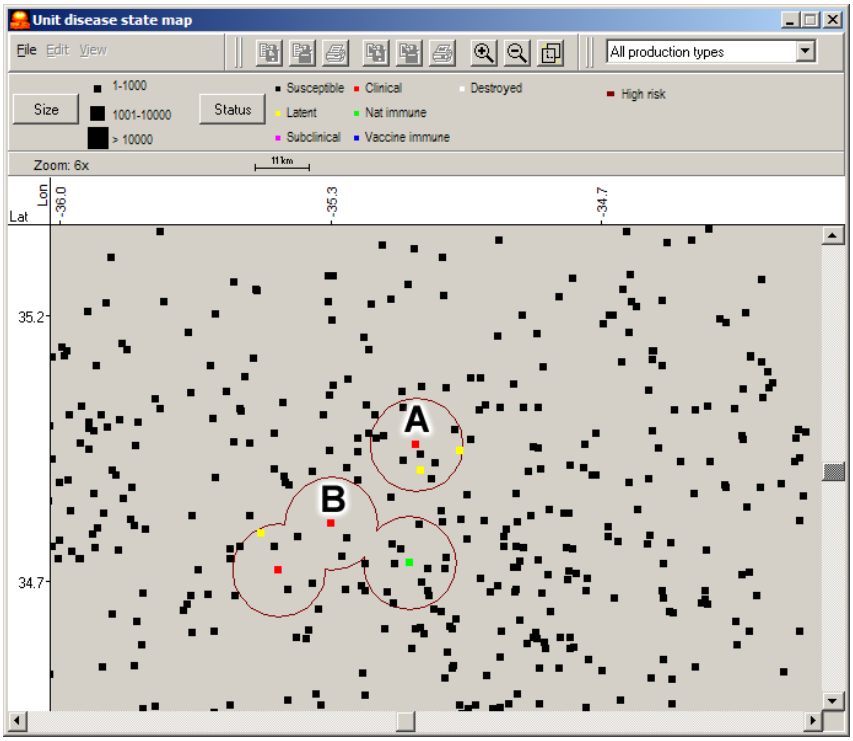


Figure 9-41. Basic zones in *NAADSM*.

Four zone foci have been created in this situation, as a result of detection of clinical disease. At this point in the simulation, three units that triggered zones are clinically infectious (red). One has advanced to the naturally immune state (green). Region A represents a single zone. Region B was formed by three overlapping zones. All of region B is now treated as a single zone.

9.12.1.1. Movement control in zones

Zones are used to restrict disease spread by direct and indirect contact among units in two ways. First, no contacts are allowed to occur from units in a zone to units outside of that zone: that is, contacts are not allowed to leave a zone. (By contrast, contacts may enter zones: a contact can occur if it has its source outside of a zone and its destination inside a zone.)

Second, the frequency of contacts that occur within a zone can altered. Recall from section 9.6.3 that a "baseline" contact rate is established for each combination of production types included in a *NAADSM* model, and that this contact rate can be modified using a relational function to reflect the reduction in the baseline contact rate after detection of disease in the population. The movement restrictions described in section 9.6.3.1.4 are applied "globally": that is, these movement restrictions affect all contacts that occur in the population. Local movement restrictions can also be implemented within zones in areas around detected units. Zone-based local movement restrictions operate independently of global movement restrictions, so it is possible to simulate different changes in the frequencies of contact at different spatial scales.

The rules applied to zones for movement control are summarized in the following examples, which refer to Figure 9-41 above:

- Units in zone A cannot be sources of contact with units that are outside of the zoned areas. Similarly, units in zone B cannot be sources of contact with units that are outside of the zoned areas.

- Units in zone A cannot be sources of contact with units in zone B, nor can units in zone B be sources of contact with units in zone A.

- Units outside of the zoned areas can be sources of contact with units in zone A and zone B.

- Units in zone A can be sources of contact to other units in zone A. Similarly, units in zone B can be sources of contact with other units in zone B. The rate at which these contacts occur, however, can be altered or reduced from the baseline rate of contact.

9.12.1.2. Disease detection in zones

Disease detection in zones is very similar to disease detection by herd exams (section 9.10.1). Each zone can be associated with a multiplier, that increases the likelihood that clinical signs will be observed in a clinically infectious herd that is inside a zone. Also like disease detection by herd exams, it is assumed that reporting of observed clinical signs in a unit inside a zone is guaranteed.

Note that *NAADSM* currently does not simulate diagnostic testing or any other specific activity that might be carried out in zones that might lead to improved disease detection. The detection multiplier specified for each zone applies only to clinical units.

9.12.1.3. Multiple zone levels

Figure 9-41 shows only a single type of zone. Although there are two separate regions (A and B), the same parameters apply to both regions. It is possible, however, to specify several zones. Figure 9-42 shows a map from a simulation in which two types of zones are used. The regions designated A and B have the same zone type. Outside of these, region C represents a second, larger type of zone. Note that region C is composed of several overlapping circles of the same zone type. Different zone types are always concentric: the two types of zones shown in Figure 9-42 are formed around the same detected units.

In situations that use multiple zone types, more stringent movement controls and/or more effective disease detection could be modeled in the smaller (higher risk) zone type, with less stringent control and detection in the larger (lower risk) type.

NAADSM imposes movement restriction rules for contacts across multiple zones which are similar to those described in section 9.12.1.1. Readers are referred to the *NAADSM* model

specification (*NAADSM* Development Team, 2010a) for a complete description of how zones are implemented.

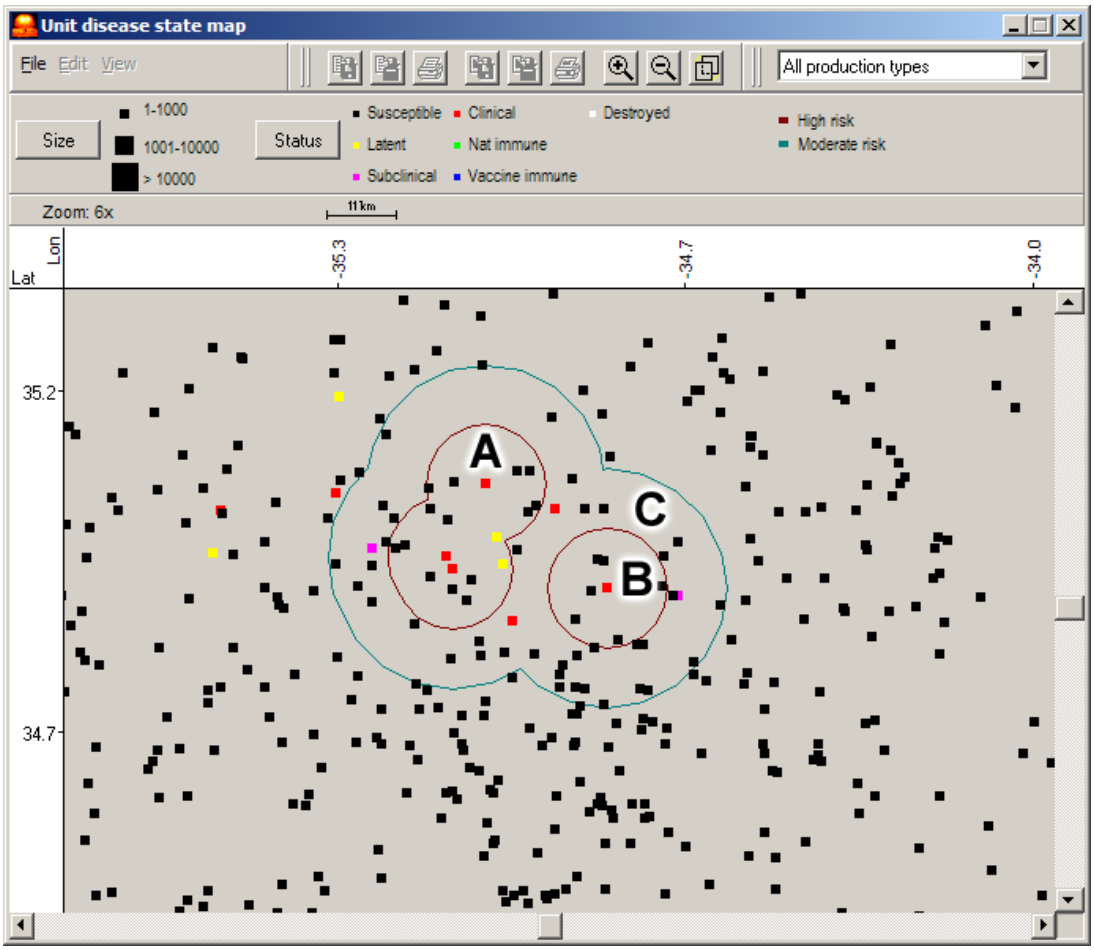


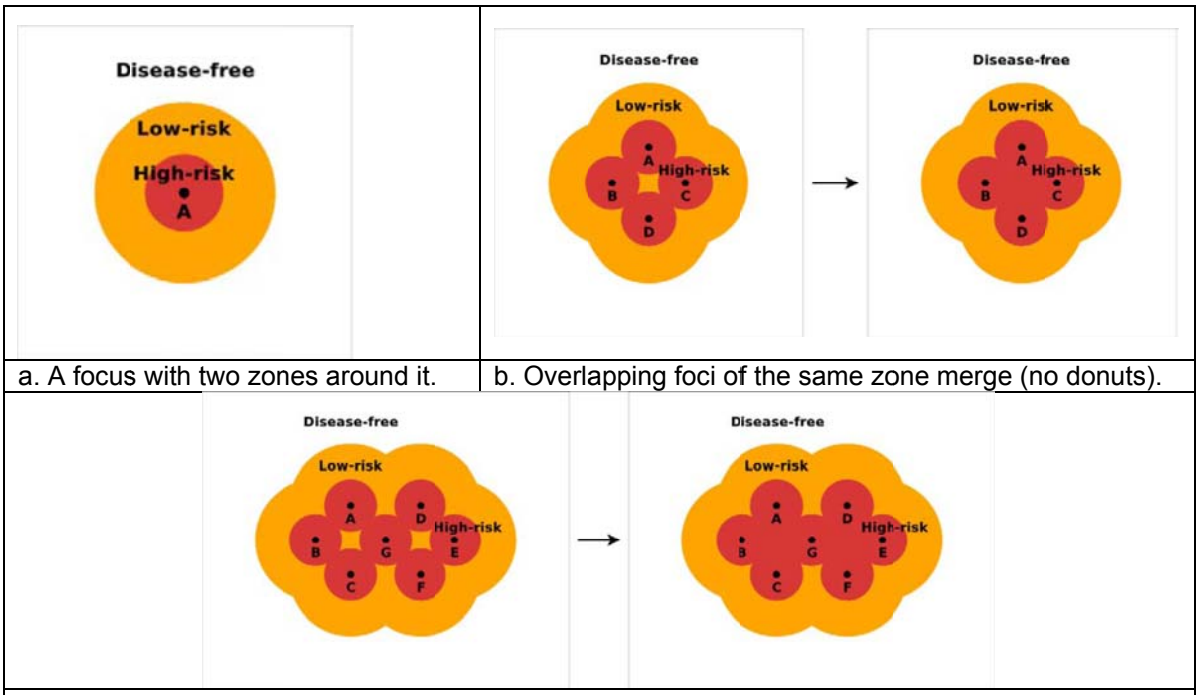
Figure 9-42. Multiple zone types in *NAADSM*.

Two types of zones have been parameterized in this simulation. These types were designated “high risk” and “moderate risk”, as shown in the caption above the map. Two “high risk” zones (A and B) are present, as well as one “moderate risk” zone (C). As described in the text, zone types are created concentrically in *NAADSM*. In the situation shown, “moderate risk” zones have a larger radius than and always surround “high risk” zones.

9.12.1.4. Additional characteristics of zones

Zones are dynamic: as illustrated, above (Figure 9-41 and Figure 9-42), the shape and size of zones can change over time during a simulation. When an infected unit is newly detected in an existing zone area, a new zone focus is created, and the zone expands. Overlapping regions of the same zone type merge (Figure 9-43b). Zones with lower surveillance levels are absorbed when enclosed by a zone of a higher surveillance level (Figure 9-43c).

Although zones can change shape and size, there is currently no way in *NAADSM* to remove them. Once in place, zone restrictions cannot be lifted. Readers are again referred to the *NAADSM* model specification (*NAADSM* Development Team, 2010a) for complete details.



a. A focus with two zones around it.

b. Overlapping foci of the same zone merge (no donuts).

c. Enclosed areas of a lower surveillance level are automatically absorbed.

Figure 9-43. Several surveillance zone concepts.

9.12.2. Setting zone parameters

A series of three windows is used to set zone parameters. These windows are displayed using the commands on the **Scenario parameters → Zone** menu (Figure 9-44). Depending on your selections in the **Zone options** window, the remaining two menu commands may be disabled.

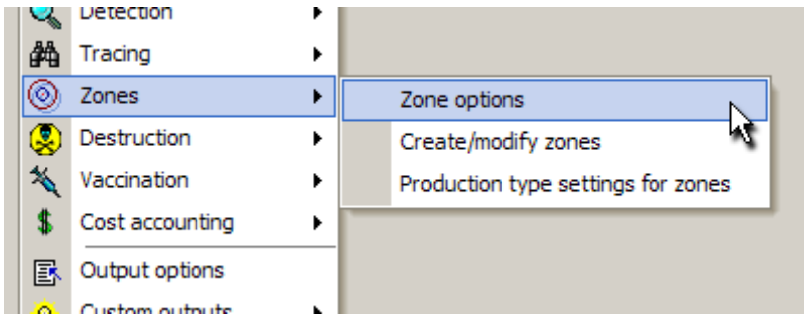


Figure 9-44. Commands of the **Scenario parameters→ Zones** menu.

9.12.2.1. Setting zone options

Select **Scenario parameters → Zones → Zone options** to turn on or turn off the use of zones in the simulation. The form only contains two radio buttons (Figure 9-45). Choosing **Yes, include zones** turns on using zones; and **No, do not include zones** turns off using zones and disables the remaining zone menu items.

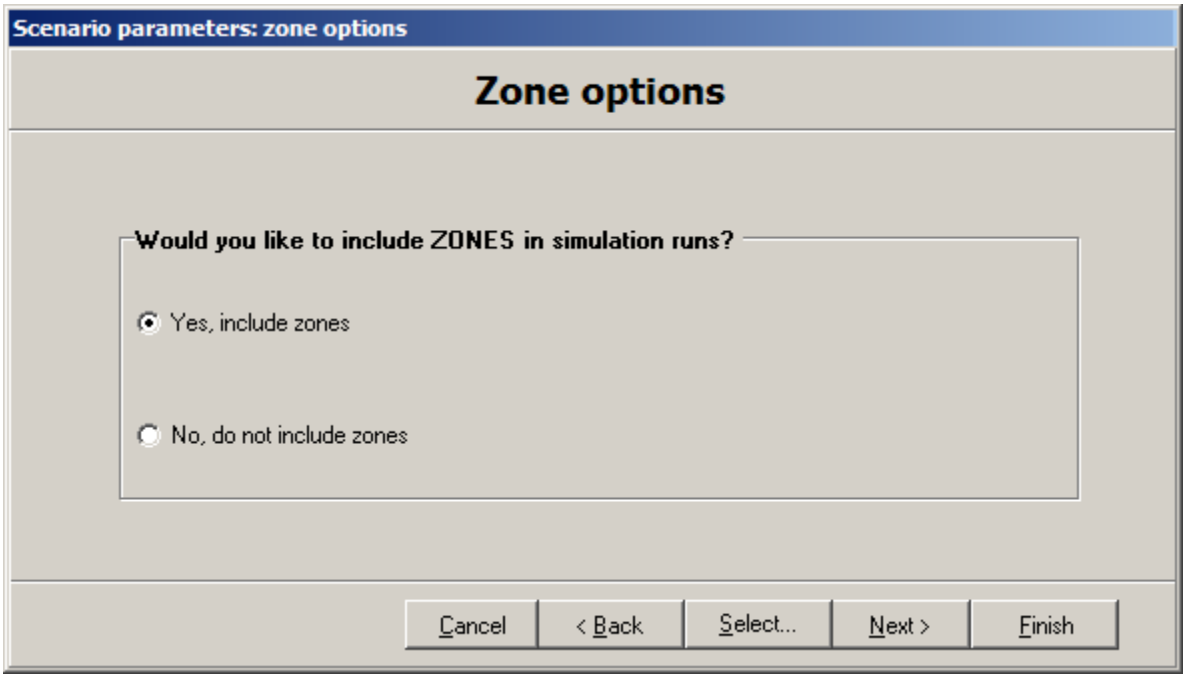


Figure 9-45. The **Zone options** window parameters.

9.12.2.2. Creating and modifying zones

If the **Yes, include zones** radio button is selected in the **Zones options** window, then the **Create/modify zones** menu command and **Zones** wizard screen will be enabled (Figure 9-46).

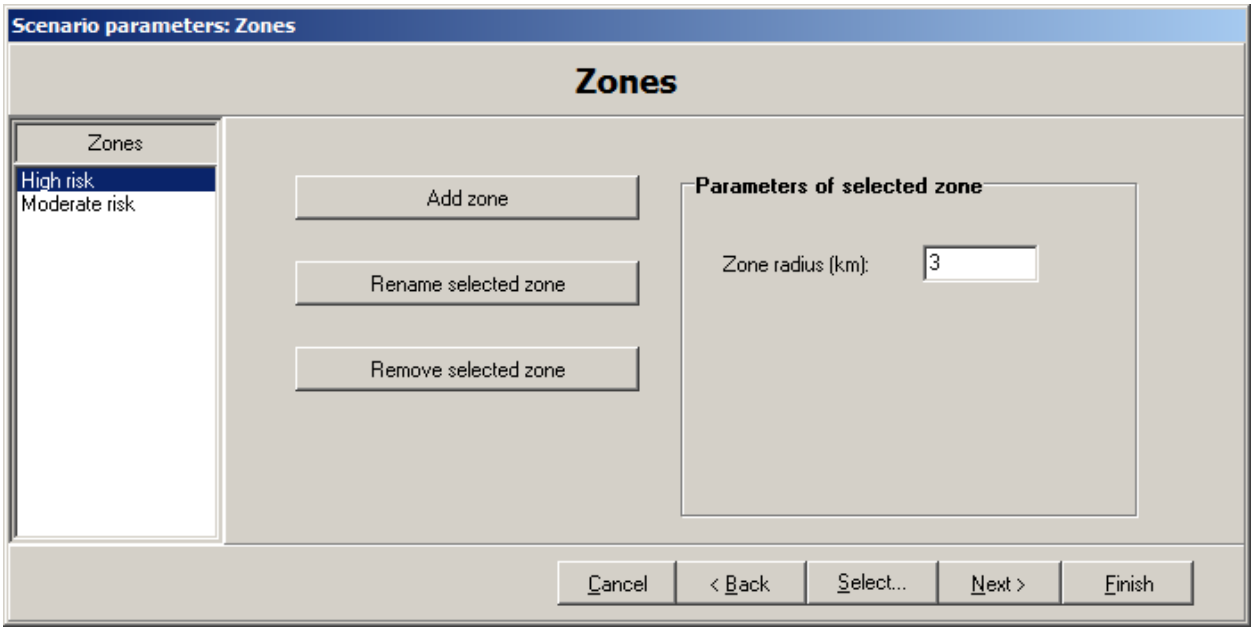


Figure 9-46. The **Zones** window for setting up zones.
The two zones included in the sample scenario (named "High risk" and "Moderate risk") are shown.

Zones are specified by a user-defined name and radius. Click the **Add zone** button to create and name a new zone and then set the **Zone radius** in units of kilometers. Specify a unique zone radius (in km) for the selected zone. *NAADSM* does not limit the number of zones which can be created, but no two zones may be the same size.

The **Rename selected zone** and **Remove selected zone** buttons operate on the zone currently selected on the left side of the window, and allow the user to rename or delete a specified zone, respectively.

9.12.2.3. Production type settings for zones

The **Zones parameters** window (Figure 9-47) is used to set production-type specific parameters for modeling the use of zones. This window is accessed using the **Scenario parameters → Zones → Production types settings** for zones menu command.

9.12.2.3.1. Creating (triggering) zones

There are three choices regarding the circumstances for creating a zone around a unit of the selected production type (Figure 9-47). Check the appropriate box to select the circumstances when zones will be created around a unit. Parameters for the trigger of a zone focus are set individually for each production type:

Detection of infected units of this type triggers a zone focus: detection of infected units from clinical signs or diagnostic testing will place a zone around the unit.

Direct tracing of units of this type triggers a zone focus: forward or backward tracing of a direct contact with an infected unit will place a zone around the unit identified by tracing.

Indirect tracing of units of this type triggers a zone focus: forward or backward tracing of an indirect contact with an infected unit will place a zone around the unit identified by tracing.

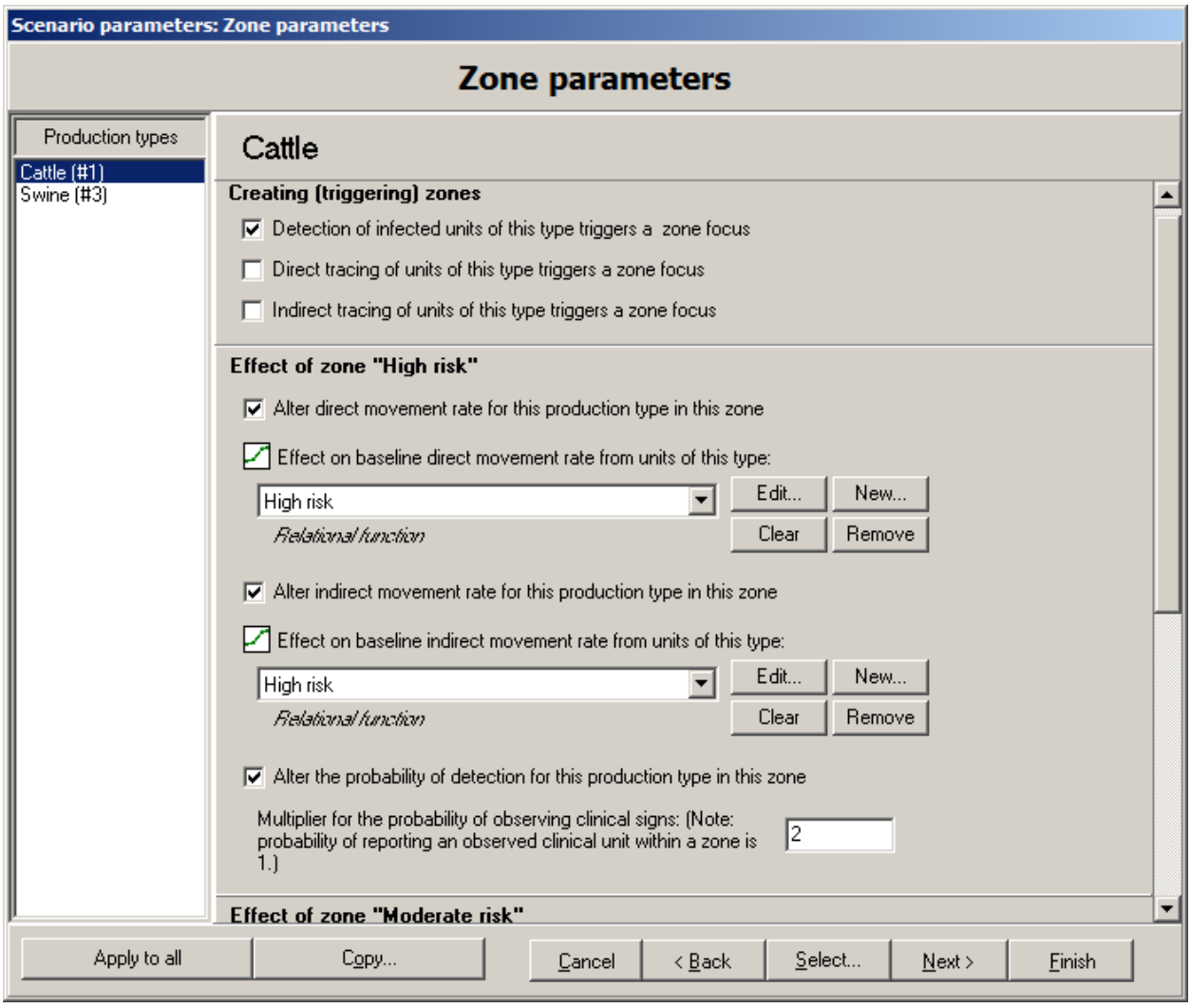


Figure 9-47. The **Zone parameters** window for managing zone attributes.

9.12.2.3.2. Setting the effects of each zone

Zone effects can be defined for each zone and may affect movement rates for direct and indirect contacts.

Alter direct movement rate for this production type in this zone: select this option to alter the direct movement rate. If this box is checked for the selected production type and zone, the **Effect of baseline direct movement rate from units of this type** will be enabled. This variable is a relational function of the percent change in the direct movement rate from units of the production type versus the number of days since the first detection. Select or create a function with the *rel* selection box (see Section 7.1.6).

Alter indirect movement rate for this production type in this zone: create a probability density function indication how indirect movement is controlled. If this box is checked for the selected production type and zone, the **Effect of baseline indirect movement rate from units of this type** will be enabled. This variable is a relational function of the percent change in the indirect movement rate from units of the production type versus the number of days since the first detection. Select or create a function with the *rel* selection box (see Section 7.1.6).

Alter the probability of detection for this production type in this zone: Enter a **Multiplier for the probability of observing clinical signs**, altering the baseline probability of observing clinical signs.

9.13. 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 9-48). Depending on your selections in the **Global destruction options** window, the remaining menu commands may be disabled.

Figure 9-48. Commands of the **Scenario parameters→ Destruction** menu.

9.13.1. Global destruction options

Some settings affect destruction regardless of the production types involved: these settings are made in the **Global destruction options** window (Figure 9-49), 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 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 a 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 7.1.6).

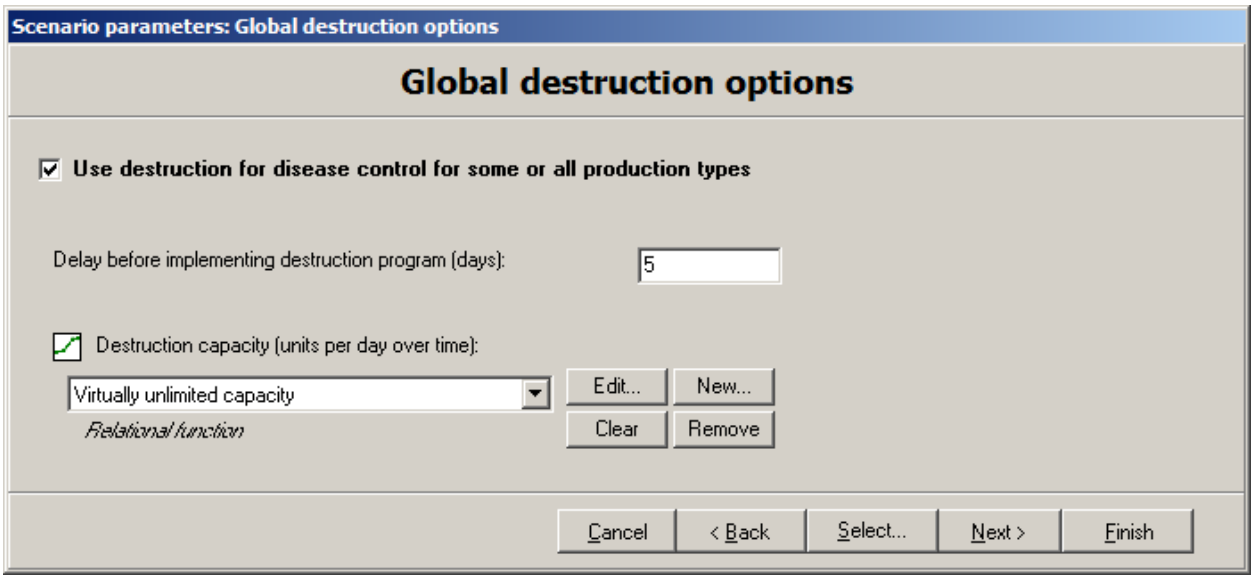


Figure 9-49. The **Global destruction options** window.

9.13.2. Production type settings for destruction

The **Destruction** window (Figure 9-50) is used to set production type-specific parameters for modeling destruction.

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.

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.

Pre-emptive destruction of units of this production type: choose the conditions under which units of this production type should be subjected to preemptive destruction (*i.e.*, units of this type will be targets of preemptive destruction). You may choose to preemptively destroy

units that have had either direct or indirect contact with a detected unit as determined by tracing (these options will be available only if tracing is being carried out: see Section 9.9). You may also choose to preemptively destroy units that are located within a destruction ring (see above).

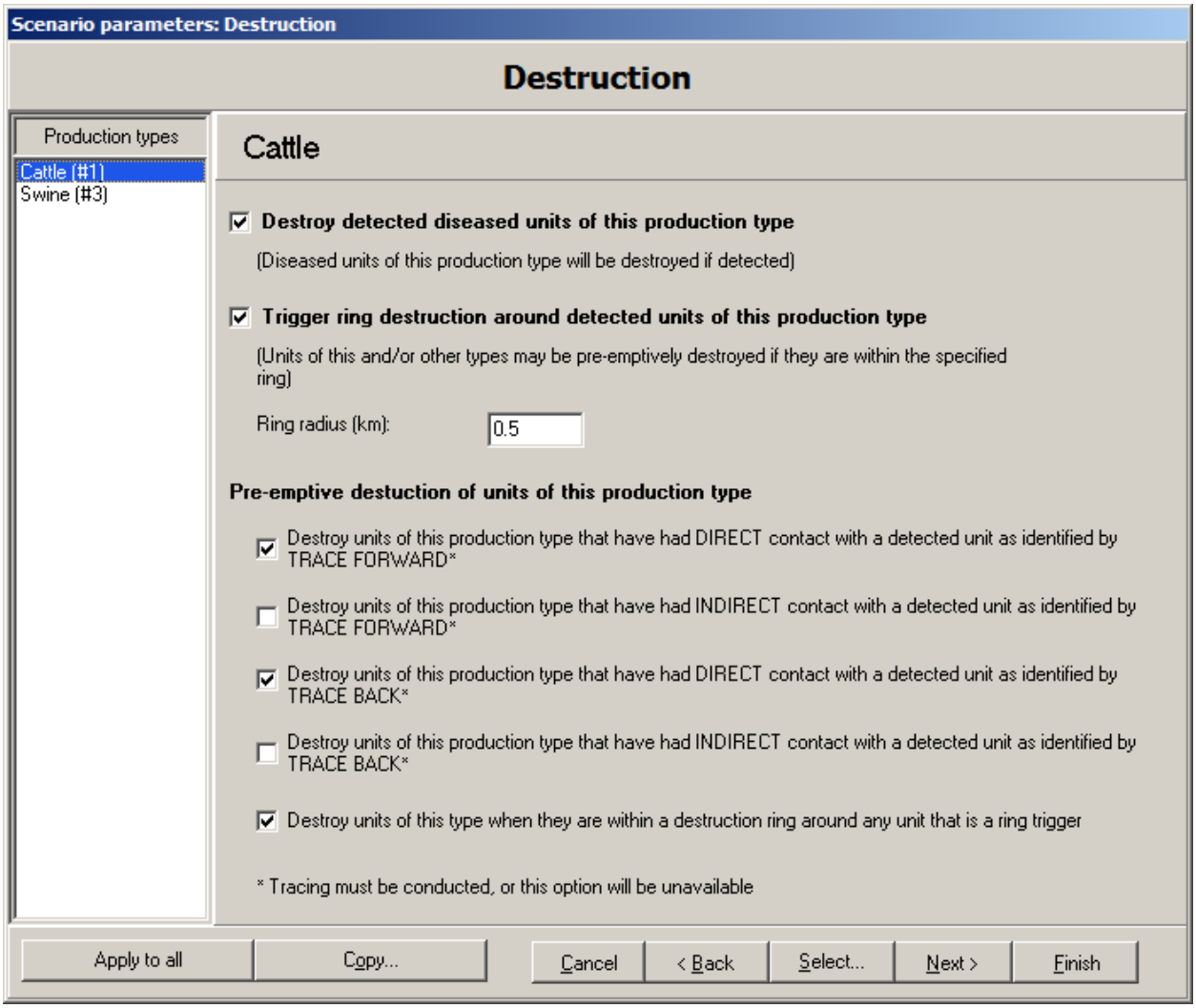


Figure 9-50. The **Destruction** window.

9.13.3. Destruction priorities

If destruction capacity is limited, the priority with which units are destroyed may 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 the queue for destruction.

9.13.3.1. Setting destruction priorities

The **Destruction priorities** window (Figure 9-51) allows you to specify the destruction priorities that you would like to model.

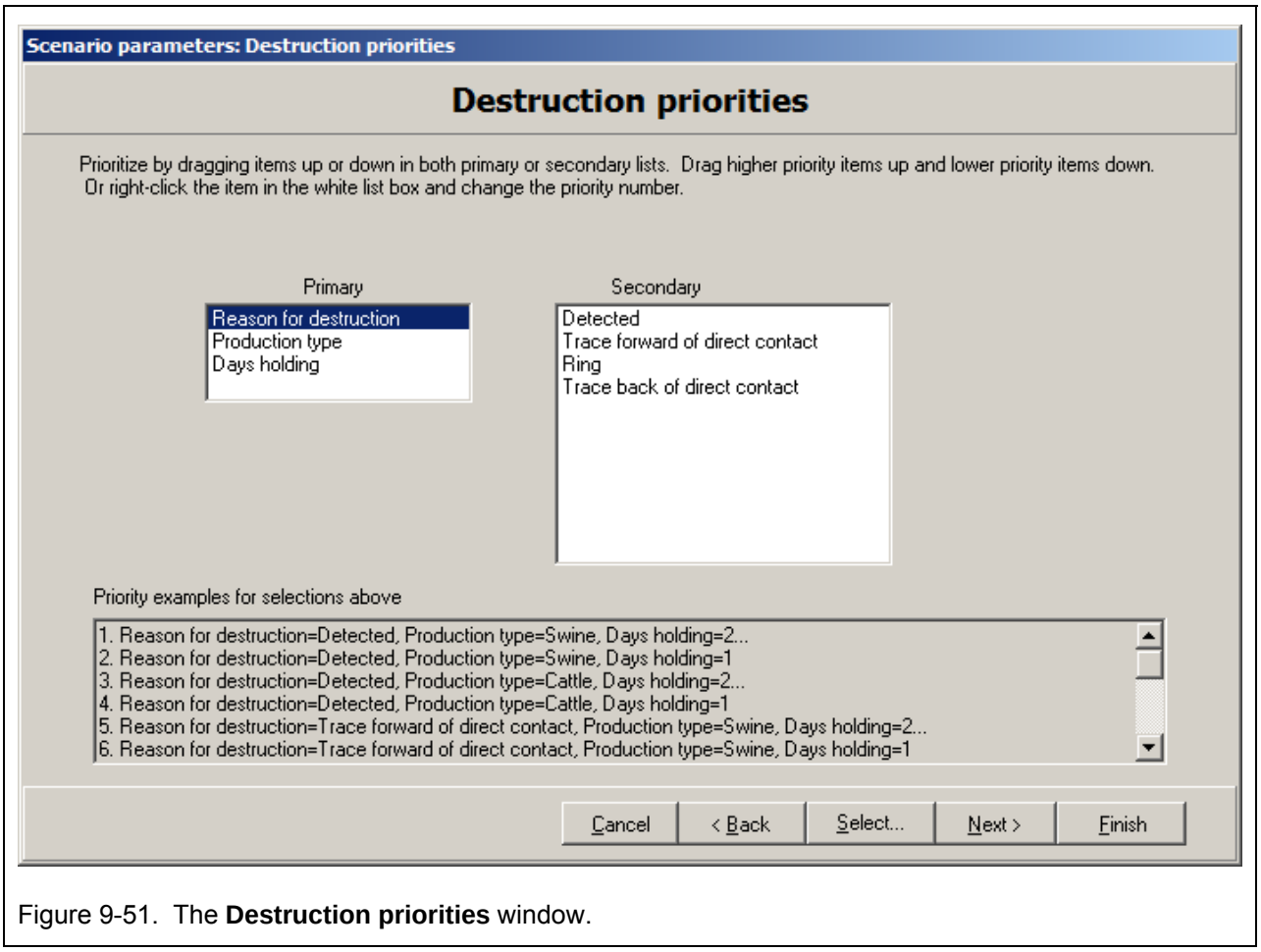


Figure 9-51. The **Destruction priorities** window.

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 in the queue for destruction. Reasons for destruction include all of the different reasons that a herd could have been targeted for destruction: detection of disease (Section 9.8); identification of a direct or an indirect contact with a detected unit by trace investigation (Section 9.9); or proximity to a detected unit within a specified radius (circle/ring destruction: see Section 9.13.2). 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 9.13.3.2.

Box 9-23. Detected units

Recall that units need not actually be infected to be detected: uninfected units that falsely test positive for disease (Section 9.11) will be treated just like infected, detected units for purposes of destruction.

9.13.3.2. 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

9.14. Modeling vaccination

A vaccination campaign may also be modeled as a method of disease control. 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 and which production types will be affected.

When a unit is vaccinated, it remains Susceptible for a time while immunity develops, then becomes Vaccine Immune. In the current version of *NAADSM*, vaccination is always 100% effective and conveys complete immunity to the vaccinated herd. 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.

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 9-52). Depending on your selections in the **Global vaccination options** window, the remaining menu commands may be disabled.

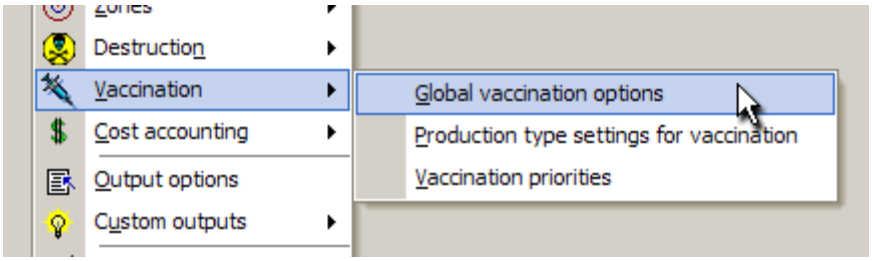


Figure 9-52. Commands of the **Scenario parameters→Vaccination** menu.

9.14.1. Global vaccination options

Some settings affect vaccination regardless of the production types involved: these settings are made in the **Global vaccination options** window (Figure 9-53), and are described below.

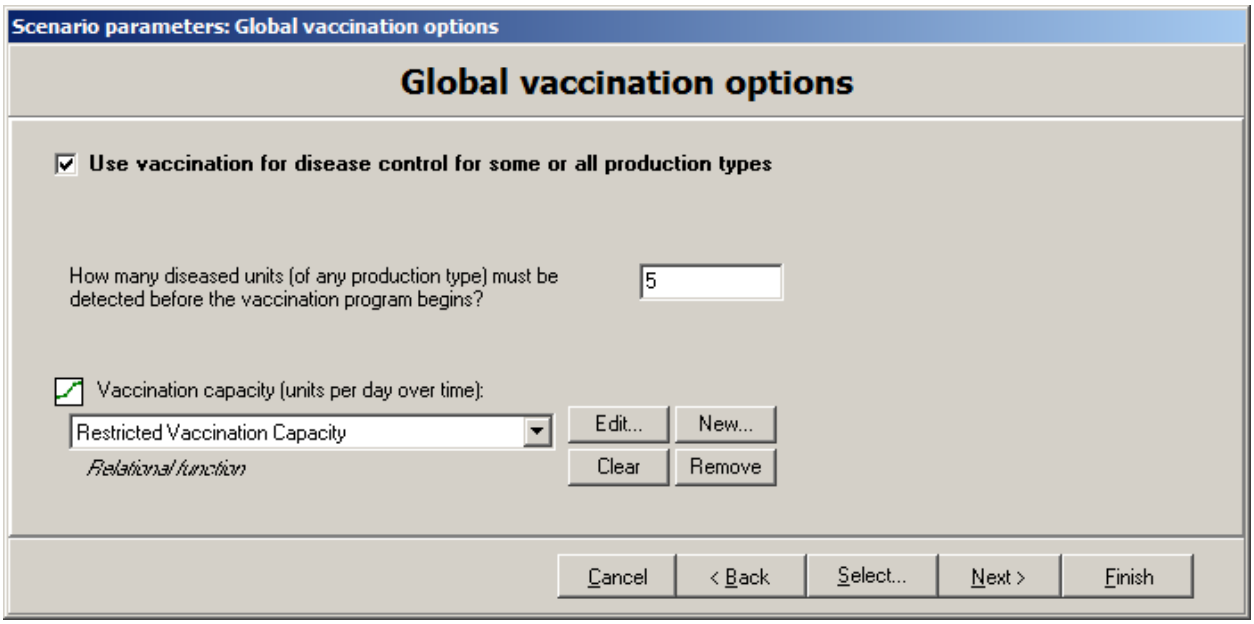


Figure 9-53. The **Global vaccination options** window.

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 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 9.13.1): 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 a relational function of the number of units (of any production type and of any size) that can be vaccinated per day versus the number of days since the first detection. Select or create a function with the *rel* selection box (see Section 7.1.6).

9.14.2. Production type settings for vaccination

The **Vaccination** window (Figure 9-54) 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 7.1.5).

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 9.5).

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 9.14.3.2).

Vaccinate detected units of this production type when they occur in vaccination rings: check this box if you want to vaccinate detected herds when the occur in vaccination rings.

Box 9-24. Vaccinating detected units

This parameter was introduced in a prior version of *NAADSM* to account for the fact that vaccination of an infected unit amounts to wasted effort: vaccinating a unit that is already infected has no effect on disease progression (Section 9.5). For some analyses, researchers wished not to expend simulated resources to vaccinate known infected units, as they expressed the opinion that known infected units would not be vaccinated in the event of an actual outbreak. *NAADSM* users now have the flexibility to choose which approach to simulate. It may be useful to recall that in *NAADSM* 4.0, however, units need not actually be infected to be detected: uninfected units that falsely test positive for disease (Section 9.11).

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.

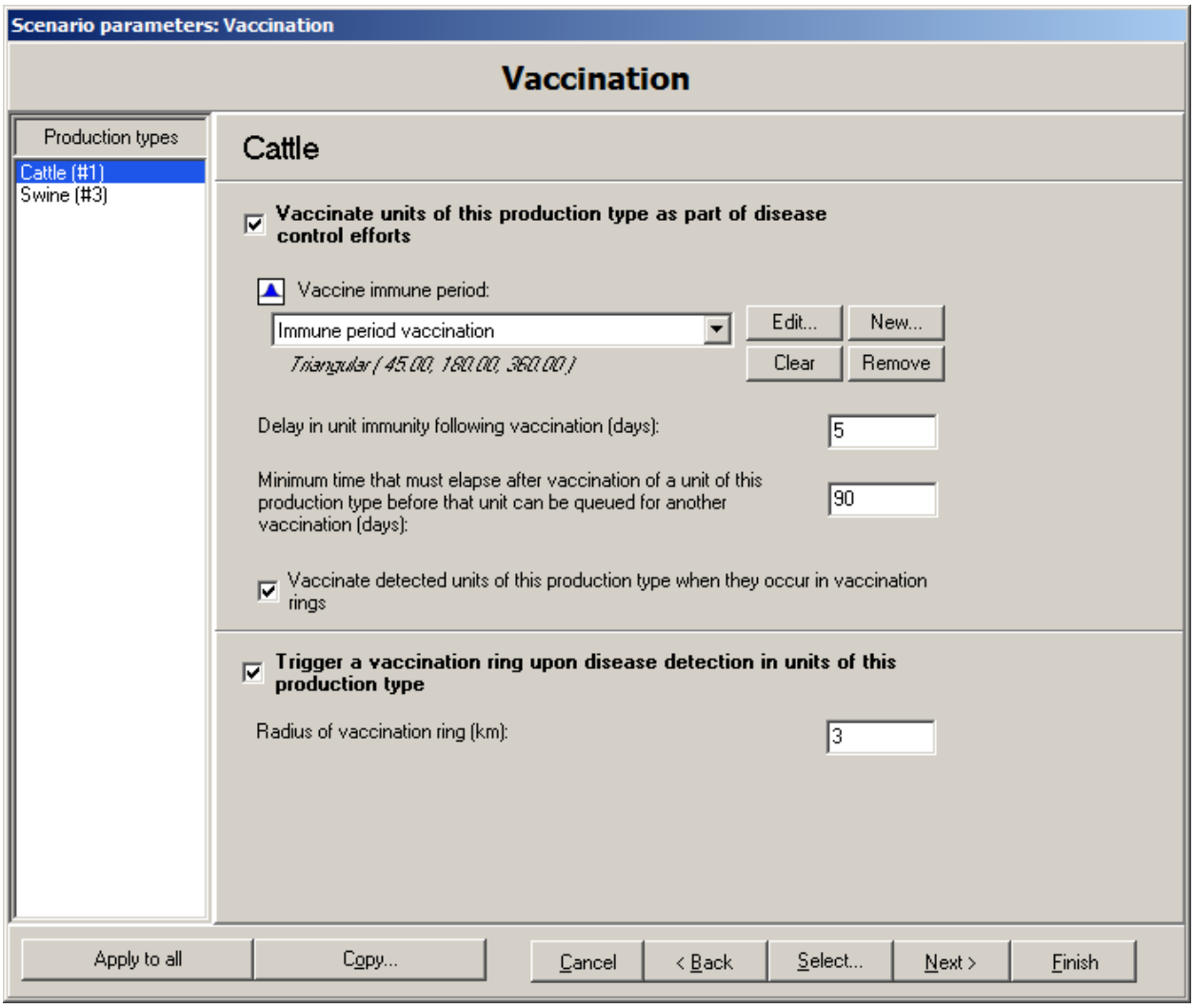


Figure 9-54. The **Vaccination** window.

9.14.3. 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 9.13.3).

9.14.3.1. Setting vaccination priorities

The **Vaccination priorities** window (Figure 9-55) allows you to specify the vaccination priorities that you would like to model.

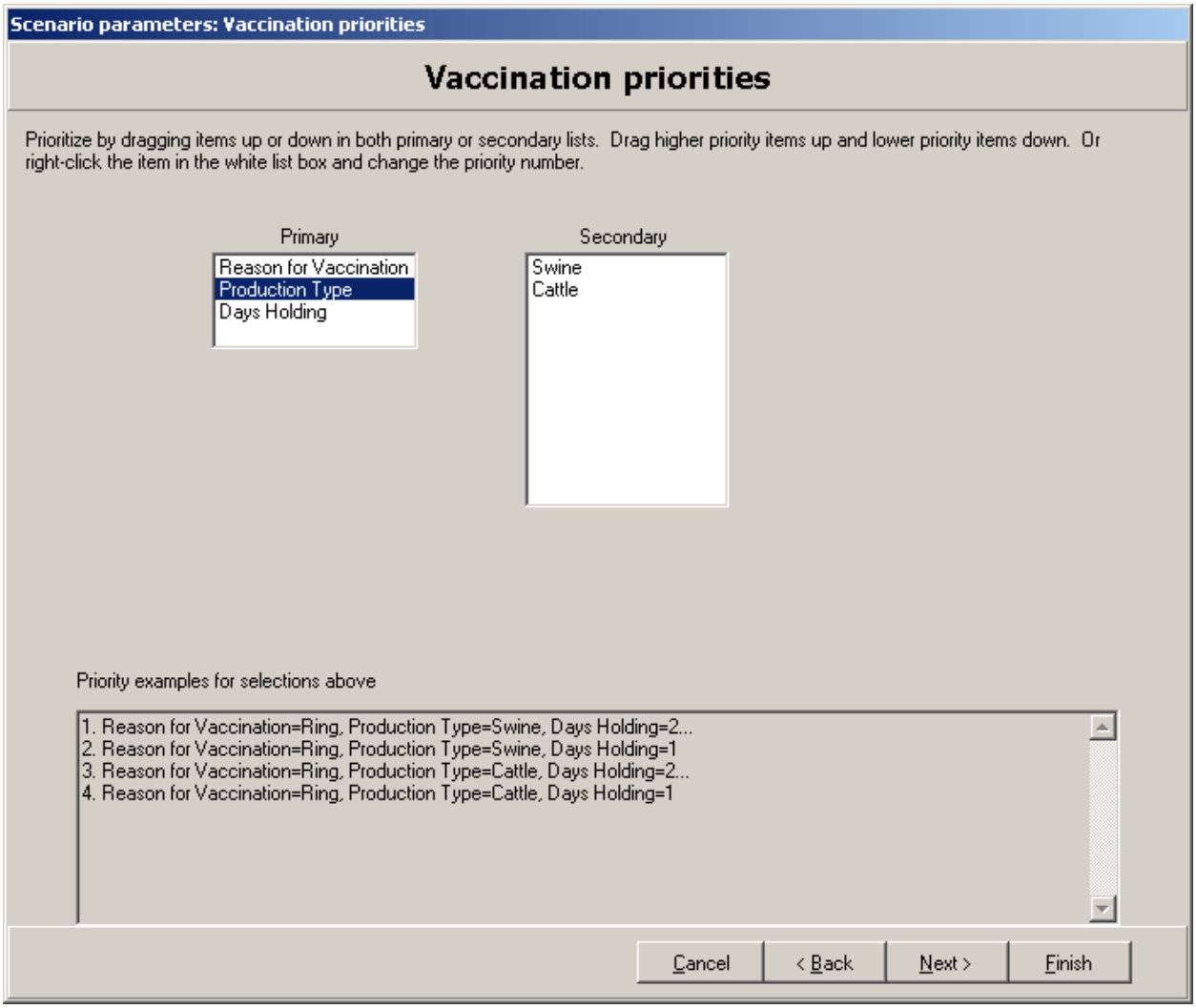


Figure 9-55. The **Vaccination priorities** window.

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.

Box 9-25. 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.

9.14.3.2. 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 9-56. Disease is detected in unit *A* ten days before disease is detected in unit *C*. Both detections trigger vaccination circles as shown.

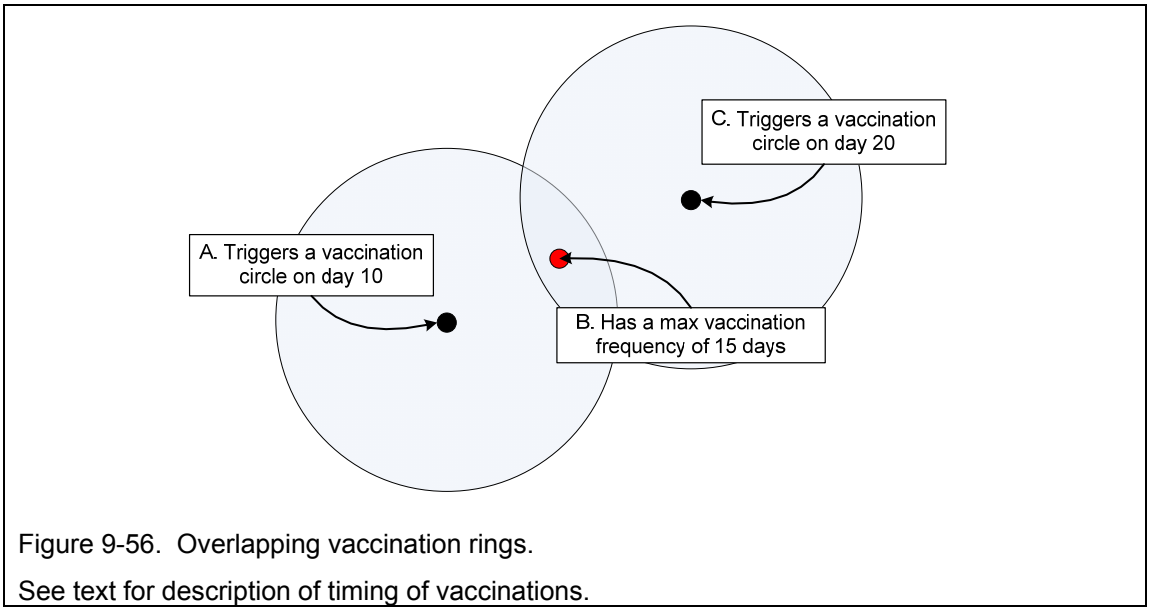


Figure 9-56. Overlapping vaccination rings.
See text for description of timing of vaccinations.

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.

9.15. 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 9-57). Depending on your selection in the **Cost accounting options** window, one or more of the **Production type settings** menu item costs series (surveillance, destruction, vaccination) may be disabled.

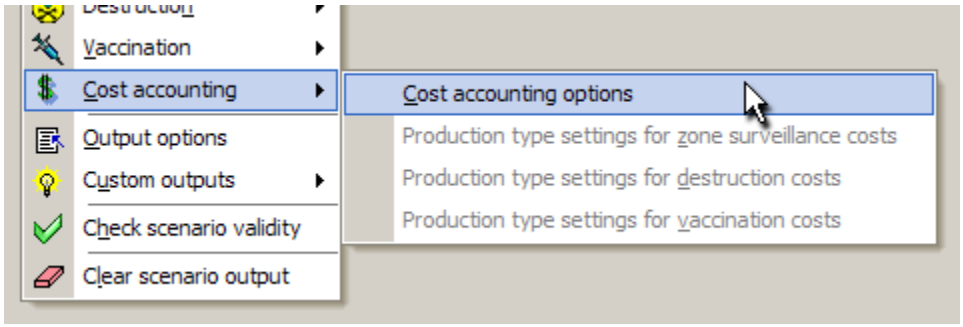


Figure 9-57. Commands of the **Scenario parameters → Cost accounting** menu.

The **Cost accounting options** window asks the question **For which control measures do you want to track DIRECT COSTS?** You can select **Track costs of surveillance zones, Track costs or destruction, Track costs of vaccination** using the check boxes (Figure 9-58). Your choices affect which of the production type settings for tracking costs menu items will be enabled.

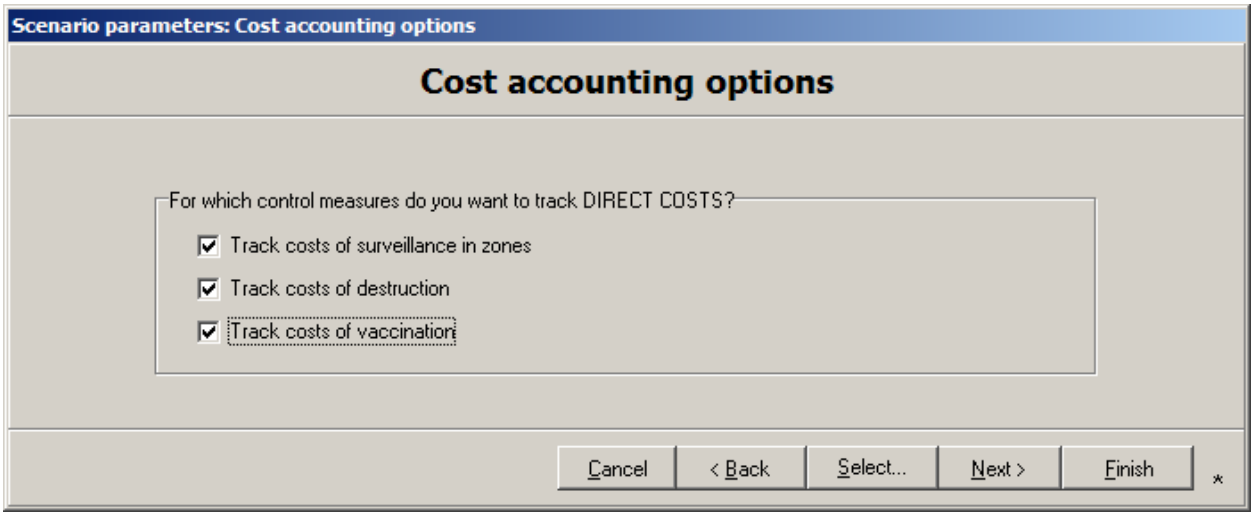


Figure 9-58. The **Cost accounting** window.

9.15.1. Costs associated with surveillance in zones

For each animal within a particular zone (see Section 9.12), there is a daily cost associated with enhanced surveillance, diagnostic testing, and any other special activity that might occur within that zone. The length of time that an individual unit – and animals within that unit – spend in a zone is the number of days from the time that the zone focus is established, until the time that at least one of the following three conditions is met: 1) The zone that the unit is in changes, *e.g.*, as a result of merging of zones or the creation of a new zone focus, 2) The unit is destroyed, or 3) The outbreak ends and all disease control measures are complete. Costs associated with zone surveillance are set here for each production type and zone combination (Figure 9-59). Enter the cost of surveillance per animal per day.

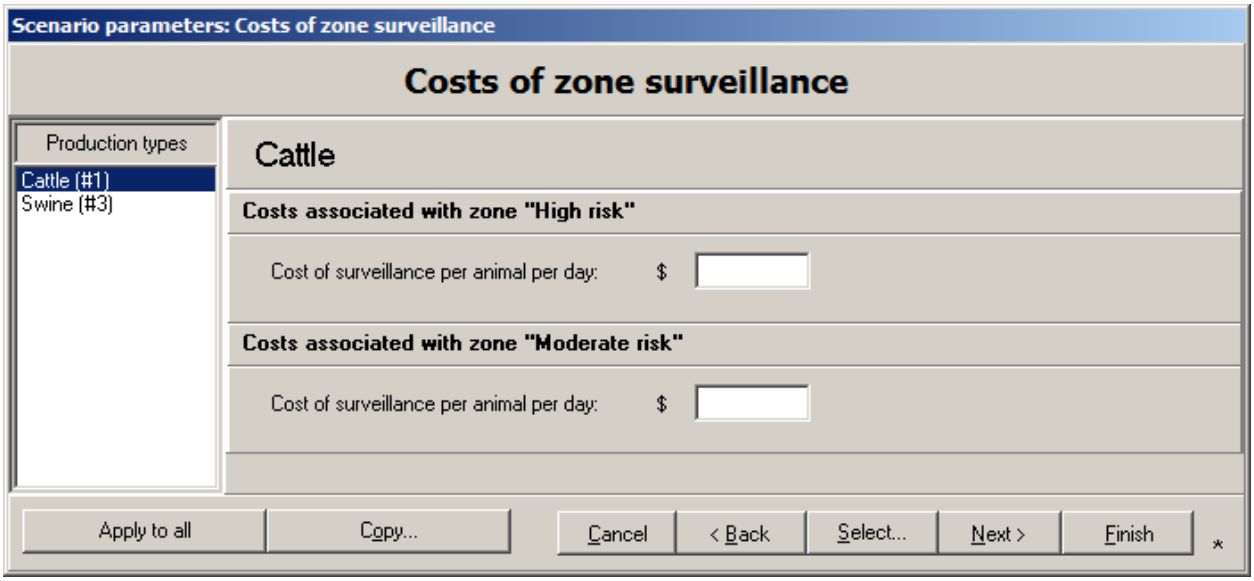


Figure 9-59. The **Costs of zone surveillance** window.

For each unit of a production type in a particular zone the number of "animal days" spent in the zone is calculated as: (Number of days that the unit is the zone) × (Number of animals in the unit). For the production type, the number of days spent in the zone is the sum of the number of animal days for each unit of that type in the zone. The total costs of zone surveillance for the production type in the zone is calculated as shown below:

(Animal days for a production type for a zone) ×
(Cost of surveillance in the zone, per animal of the production type, per day)

9.15.2. Costs associated with destruction

Costs associated with destruction are set for each production type (Figure 9-60). Some costs associated with destruction are per unit and others are per animal. 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** of 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 **indemnification**, **euthanasia**, and **carcass disposal** apply.

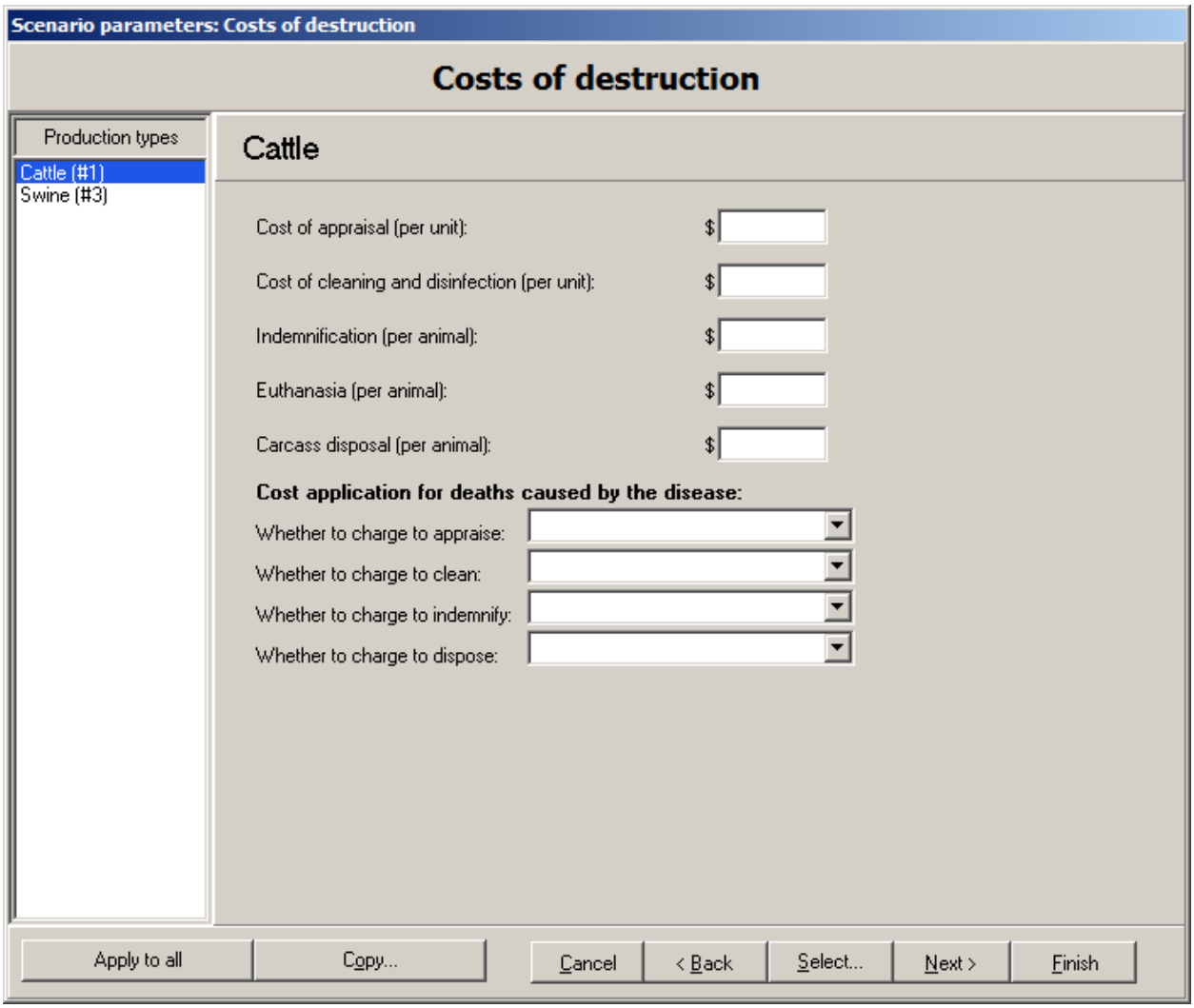


Figure 9-60. The **Costs of destruction** window.

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

(Appraisal cost + Cleaning and disinfection cost)
+ [(Number of animals in the unit)
× (Cost of euthanasia + Cost of indemnification + Cost of disposal)]

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

(Number of units destroyed) × (Appraisal cost + Cleaning and disinfection cost)
+ [(Total number of animals destroyed)
× (Cost of euthanasia + Cost of indemnification + Cost of disposal)]

9.15.2.1. Destruction costs for units in the "dead from disease" state

Accounting for costs of destruction of units that have progressed to the "dead from disease" state poses unique problems. It is clear that cost for euthanasia do not apply in such cases (at least if the typical definition for the unit-level dead from disease state is applied: see Section 3.4), but costs for cleaning and disinfection, disposal, appraisal, and indemnification may remain. Furthermore, the way in which such costs are accounted for may differ by jurisdiction. In some cases, for example, indemnification may not be paid for animals that have already died.

For units in the dead-from disease state, *NAADSM* provides three options for when to charge for appraisal, cleaning, indemnification, and disposal (Figure 9-61). These options are as follows:

- **Always applied**: Detected units in the dead-from-disease state are treated like any other unit for purposes of cost accounting. Direct costs apply as described above.
- **Never applied**: No payment is made for detected units in the dead-from-disease state. Such units are not considered in the estimation of costs associated with destruction.
- **Only applied after queued for destruction**: Costs are considered only for units that were placed in the queue to be destroyed, but which died out from disease before destruction could be carried out.

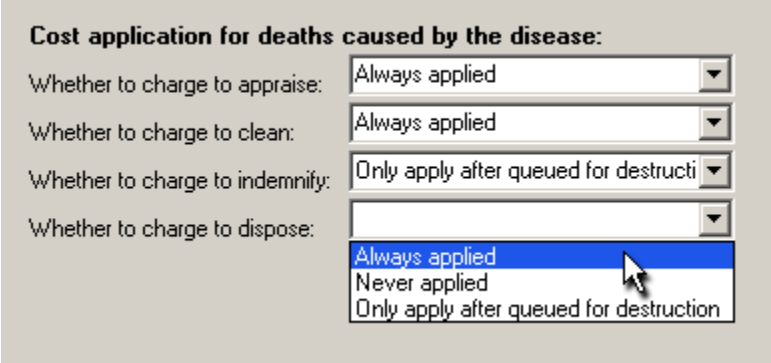


Figure 9-61. Options for accounting for costs when units have progressed to the "dead from disease" state.

These options are available in the lower section of the **Costs of destruction** window (Figure 9-60).

Box 9-26. Alternatives for cost accounting calculations

NAADSM is intended to provide general-purpose options for cost accounting, but as seen in this section, there may be many special cases that users might wish to consider, which can become quite complex. Although *NAADSM* itself cannot account for all of the schemes for cost accounting that users might wish to use, it is not difficult to export the epidemiologic outcomes generated by *NAADSM* for use in other applications where more detailed approaches to cost accounting could be implemented. All of the cost calculations shown above, for example, are simply multiplications based on the numbers of destroyed units or animals, which could be readily carried out in a spreadsheet application like *Microsoft Excel*. Box 8-1 and Section 11.4 describes how raw data regarding these epidemiological outcomes can be extracted for use outside of *NAADSM*.

Although it is tempting to try to account for specific details like those described in this section, users should also consider whether the overall accuracy of a simulation model like those produced by *NAADSM* (or by any other modeling platform) is sufficient to justify the attempt to achieve such a high level of precision regarding the quantitative outcomes of a disease outbreak.

9.15.3. 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 (Figure 9-62).

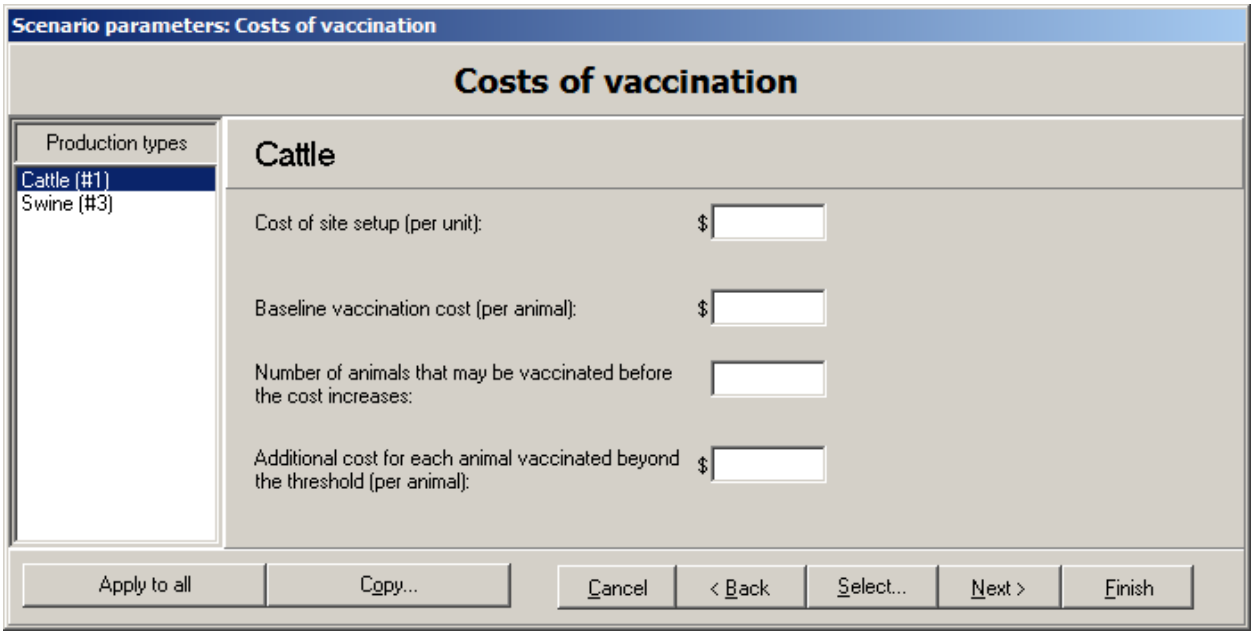


Figure 9-62. The **Costs of vaccination** window.

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

If the threshold is not reached:

[(Number of units vaccinated) × (Cost of site setup)]
+ [(Total number of animals vaccinated) × (Baseline cost per animal)]

If the threshold is reached:

[(Number of units vaccinated) × (Cost of site setup)]
+ [(Threshold level) × (Baseline cost per animal)]
+ [(Total number of animals vaccinated– Threshold level)
× (Baseline cost per animal + Additional cost per animal)]

9.16. 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. The options available on the **Output options** window (Figure 9-63) will allow you to track the daily events that occur in the model. This window is accessed from the **Scenario parameters → output options** menu item.

Recall from Chapter 4 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 a *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.

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 9-64 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 becomes 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.

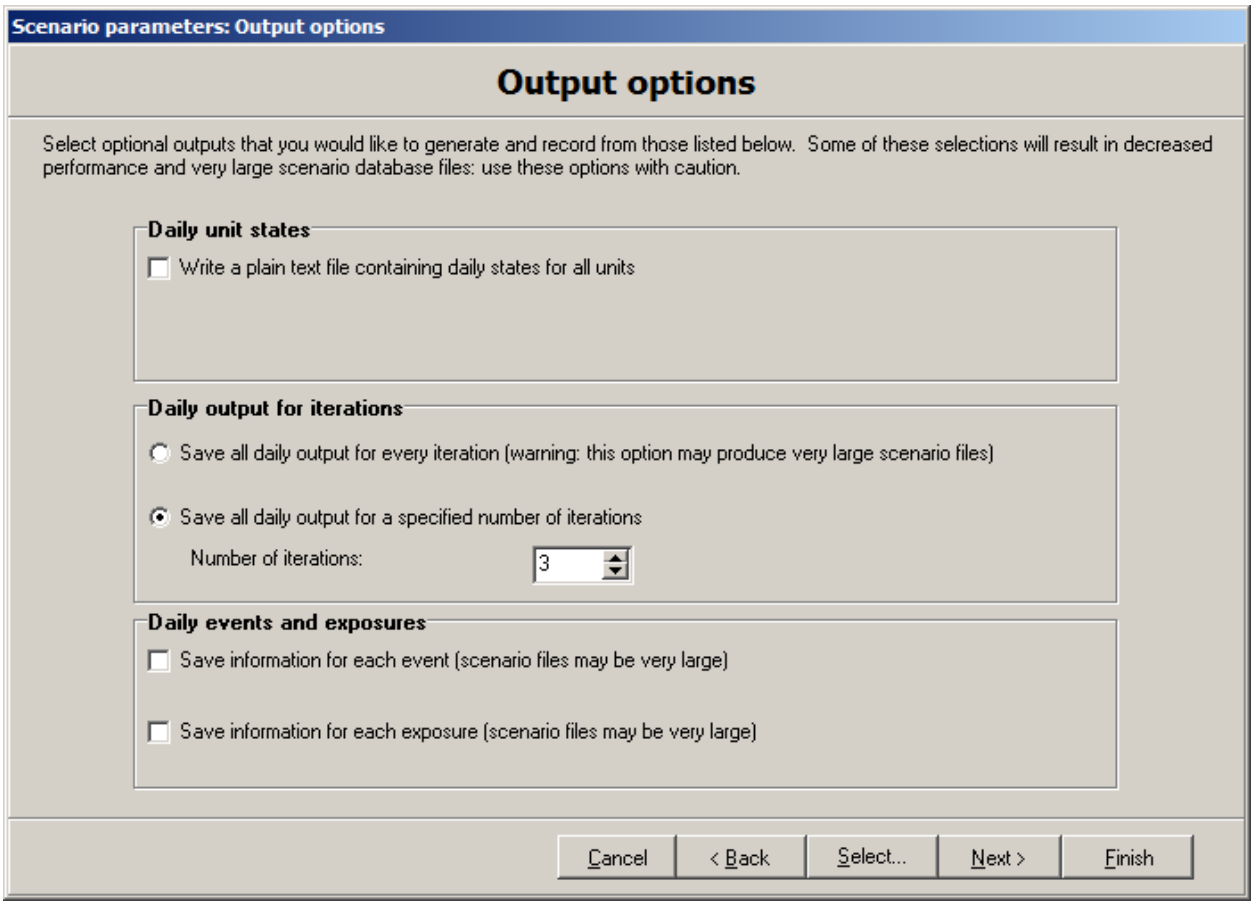


Figure 9-63. The **Output options** window.

```
Iteration 1
N N N N S S S S S L S S S N N N
N N N N S S S S S B S S S N N N
N N N N S S S S S B S S S N N N
N N N N S S S S S C S S S N N N
N N N N S S S S S C S S L N N N
N N N N S S S S S C S S B N N N
N N N D S S S S S D S S C N N N
N N N D S S S S S D 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 9-64. A sample plain text file showing daily herd (unit) states.

Disease states are indicated with the single character code shown in Table 9-2.

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.

Box 9-27. ***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 11.3.

9.17. Custom outputs

Access a SQL custom statement generator from **Scenario parameters → Custom outputs**. Check with the programming team for more information before attempting to use this feature. It is still an experimental feature that is not well documented and relies on the user knowing the data naming conventions of *NAADSM* and the syntax of Structured Query Language (SQL).

9.18. 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 Chapter 10).

Box 9-28. Attempting to run an invalid scenario

If you attempt to run a scenario (with one of the commands discussed in Chapter 10) 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.

9.19. Using the Clear scenario output command

Recall from Chapter 8 that a *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.

*Box 9-29. **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!).

10. 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 9.4) 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 11.4) will be available.

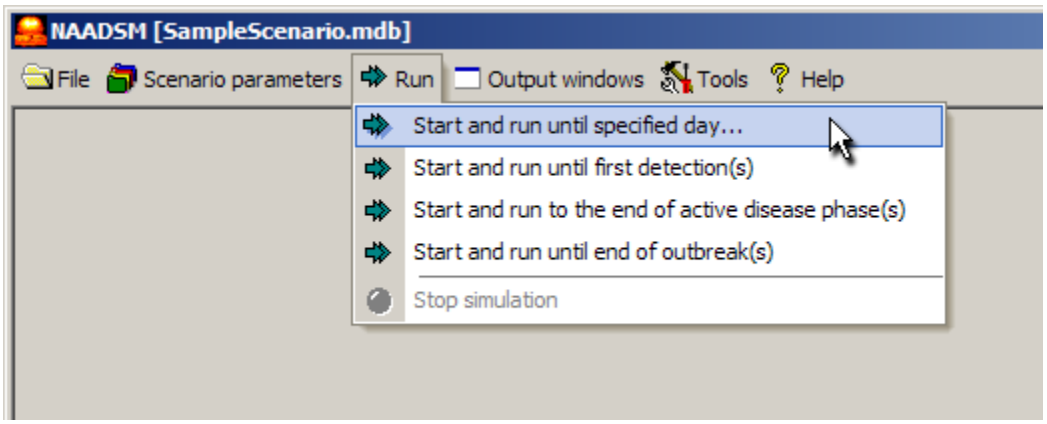


Figure 10-1. Commands of the **Run** menu.

Start and run until specified day...: Selecting this option will cause each iteration to end after exactly the specified number of simulation days. Enter a value greater than 0, then click OK.

Start and run until first detection(s): Selecting this option will stop cause each iteration to end on the first simulation day on which disease is first detected in a herd of any production type.

Start and run until end of outbreak(s): 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. Selecting this option will cause all iterations of the simulation to run until disease can longer spread in the population *and* until all control measures are complete. In other words, iterations will end when no herds remain infected, and when no herds remain wait-listed for vaccination or destruction.

Start and run until end of the active disease phase(s): Selecting this option will cause all iterations of the simulation to run until disease can no longer spread in the population, *i.e.*, until there are no longer infected herds in the population. Unlike the option discussed above, any vaccination or destruction activities that remain to be carried out *will not* occur as part of the simulation.

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.

11. 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 11-1) 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 11.2 and 11.3 10.2), as well as statistics based on many iterations of a scenario (Section 11.4). 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 11.5 for a description of these options.

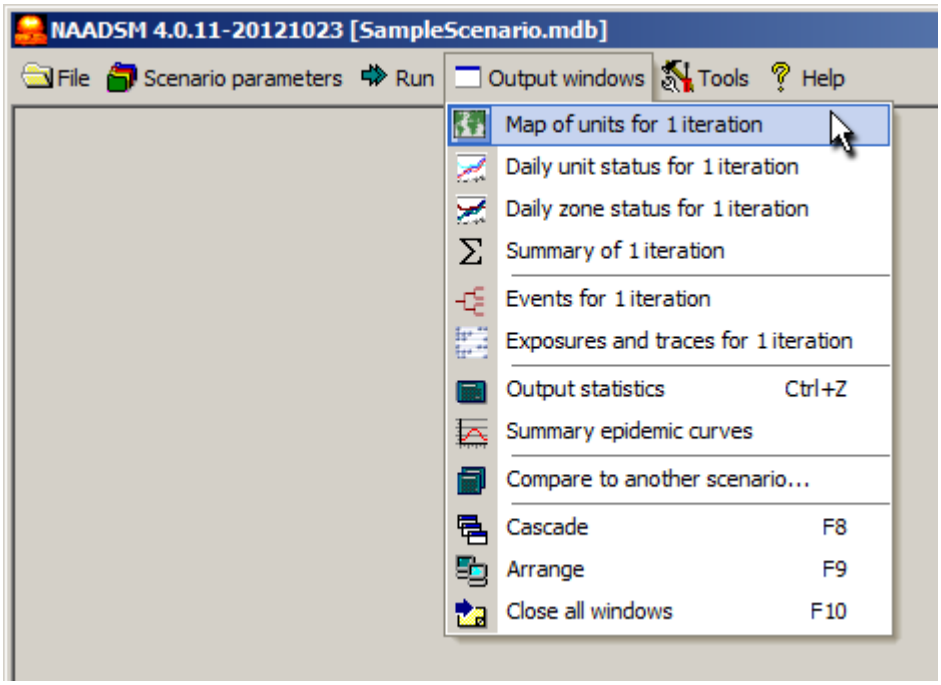


Figure 11-1. Commands of the **Output windows** menu.

Note that **Events by day** and **Contact-exposure events by day** are disabled by default. **Daily zone status for 1 iteration** is only enabled when your simulation scenario is using zones.

> *Box 11-1. 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 the *Microsoft Access*-compatible (**`*.mdb`**) *NAADSM* scenario file.

11.1. "Actual" versus "apparent" events

Several of the outputs discussed below make a distinction between "actual" and "apparent" events. *NAADSM* is omniscient, 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 other cases, an uninfected herd may test positive for disease as a result of an imperfect test. 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 investigations are examples of apparent or observable events. Transitions between disease states, the onset of vaccine immunity, and exposures are examples of actual events that generally cannot be observed.

11.2. 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.

11.2.1. The Map of units for 1 iteration window

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. The map is dynamically updated as disease progresses within herds, as disease spreads among herds, and as control measure are applied (Figure 11-2).

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.

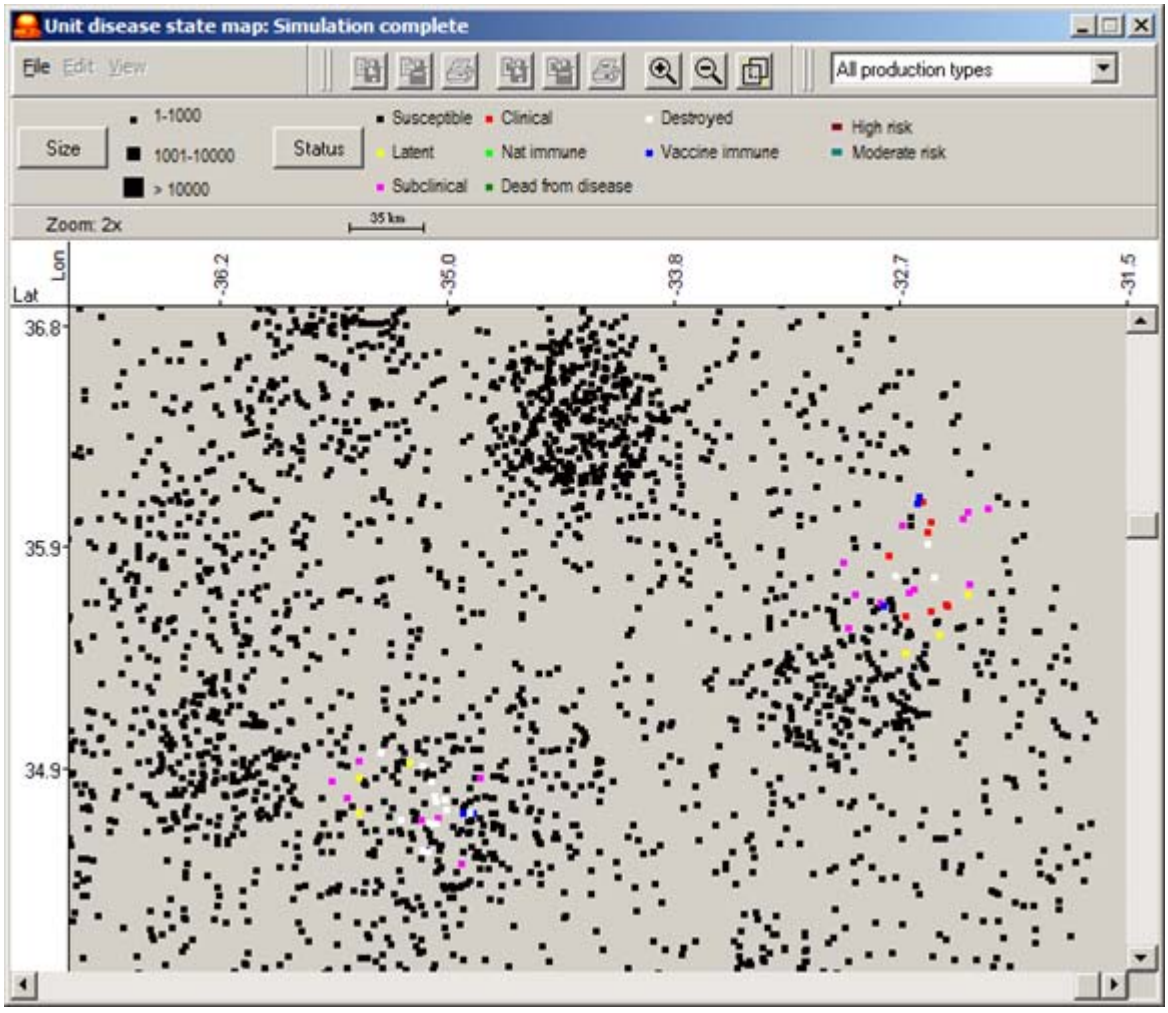


Figure 11-2. A map of herds from a completed simulation, showing actual disease status.

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 11-3 and Figure 11-4.

11.2.1.1. Map options

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

Status: this button toggles between three possible views of each herd's status. Figure 11-2 shows herds based on their actual disease status (susceptible, latent, subclinical, clinical, naturally immune, dead from disease, vaccine immune, or destroyed). By contrast, Figure 11-3 shows the same herds on the same simulated day based on their known detection status. In this view, herds may have no status at all because they are uninfected and no action has been taken that might have resulted in disease detection, or may be shown based on what is known about those herds after some activity that might have resulted in disease detection (for example, after a herd exam or a diagnostic test). A third view (control status,) shows which if any disease control measures have been used for particular herds.

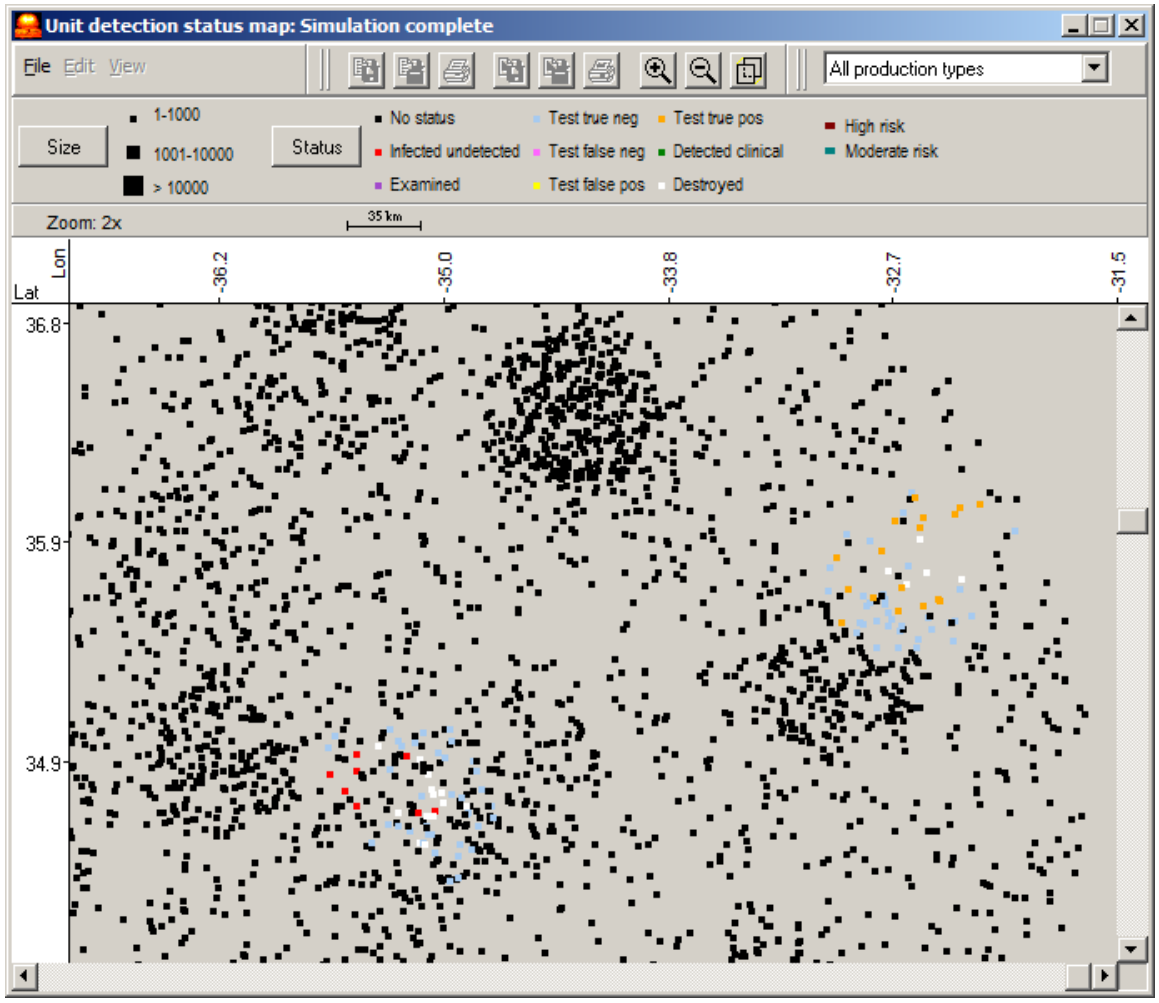


Figure 11-3. A map of herds from a completed simulation, showing detection status.

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 11-2 and Figure 11-4.

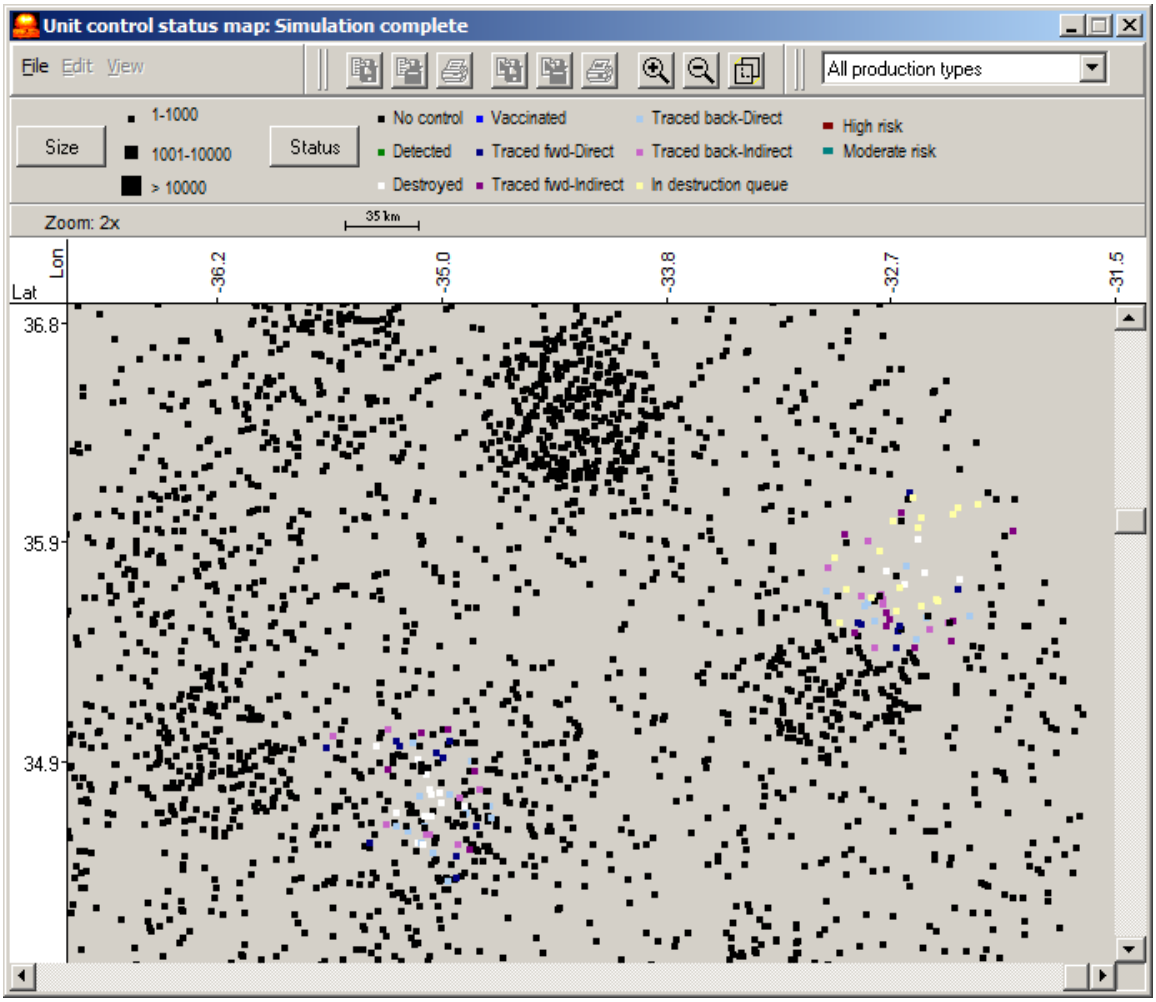


Figure 11-4. A map of herds from a completed simulation, showing control status.

This view shows the disease control status of affected herds, according to the color key shown at the top of the window. Contrast this map with the display in Figure 11-2 and Figure 11-3.

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.

The **File → Export KMZ file** menu item produces a file for import into Google Earth™ , an easy way to display the herds of the scenario on aerial imagery. The production type list box selection on the map window determines if units for all production types or only one production type will be included in the KMZ file. Selecting **Export KMZ...** from the menu will open a file browser window where you can name the file and save it to a specific location. In Google Earth™, the location of each herd will be displayed by a colored marker and the unit ID. Larger herds have a larger marking symbol than smaller herds and symbol color varies by production type. Clicking on a marker produces a table listing the production type, initial size, and initial state of the unit. The KMZ file is a zipped XML formatted text file (KML file) that can be

extracted with a zip compression utility. At this printing, Google Earth™ can be downloaded for free from <*http://earth.google.com*>. If Google Earth™ is installed on your computer, double-clicking the KMZ file will start the application and zoom to the vicinity of the herds.

11.2.2. The Daily unit status for 1 iteration window

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 11-5). The production type and iteration run can be selected in the top panel of the form.

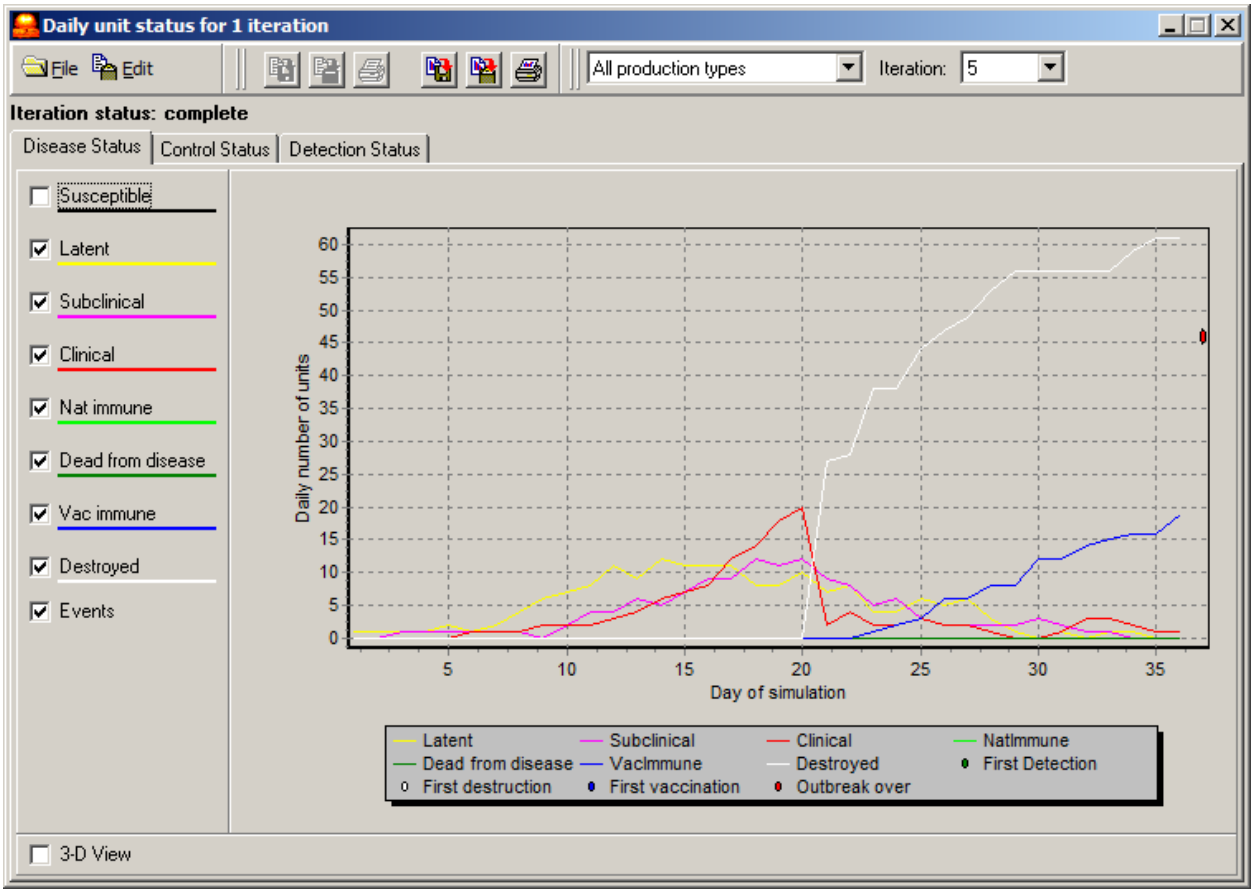


Figure 11-5. The **Daily unit status for 1 iteration** window.

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: ●). Checking or un-checking **Events**, in the lower panel of the chart shows or hides events. The **3-D View** check box toggles the chart between 2-D and 3-D views.

The chart is located on a tab sheet, labeled **Disease Status**. Charts for **Control Status** and **Detection Status** can also be viewed by clicking on their tab. Any of these charts can be viewed

while a scenario is running or at the end of the simulation. The **Control Status** chart shows the proportion of the units by day that were detected, destroyed, vaccinated, and traced. The **Detection Status** chart shows the proportion of units by day that were examined, destroyed, detected, and the results of diagnostic testing – the proportion of units testing true positive, true negative, false positive, and false negative.

11.2.3. The Summary of 1 iteration window

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 9.15).

11.2.3.1. Epidemiological outputs

The **Epidemiology** tab, **Tabular view** tab sheet, summarizes the number of herds and animals detected, mode of infection, destroyed, and vaccinated during the outbreak from one iteration. The form also shows, on the **Graphical View** tab sheet, the 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).

Four checkboxes divide the **Epidemiology** tab **Tabular view** into four main sections, each of which is discussed below. Check or uncheck one of these boxes (shown in Figure 11-6) to display or hide the outputs for that section.

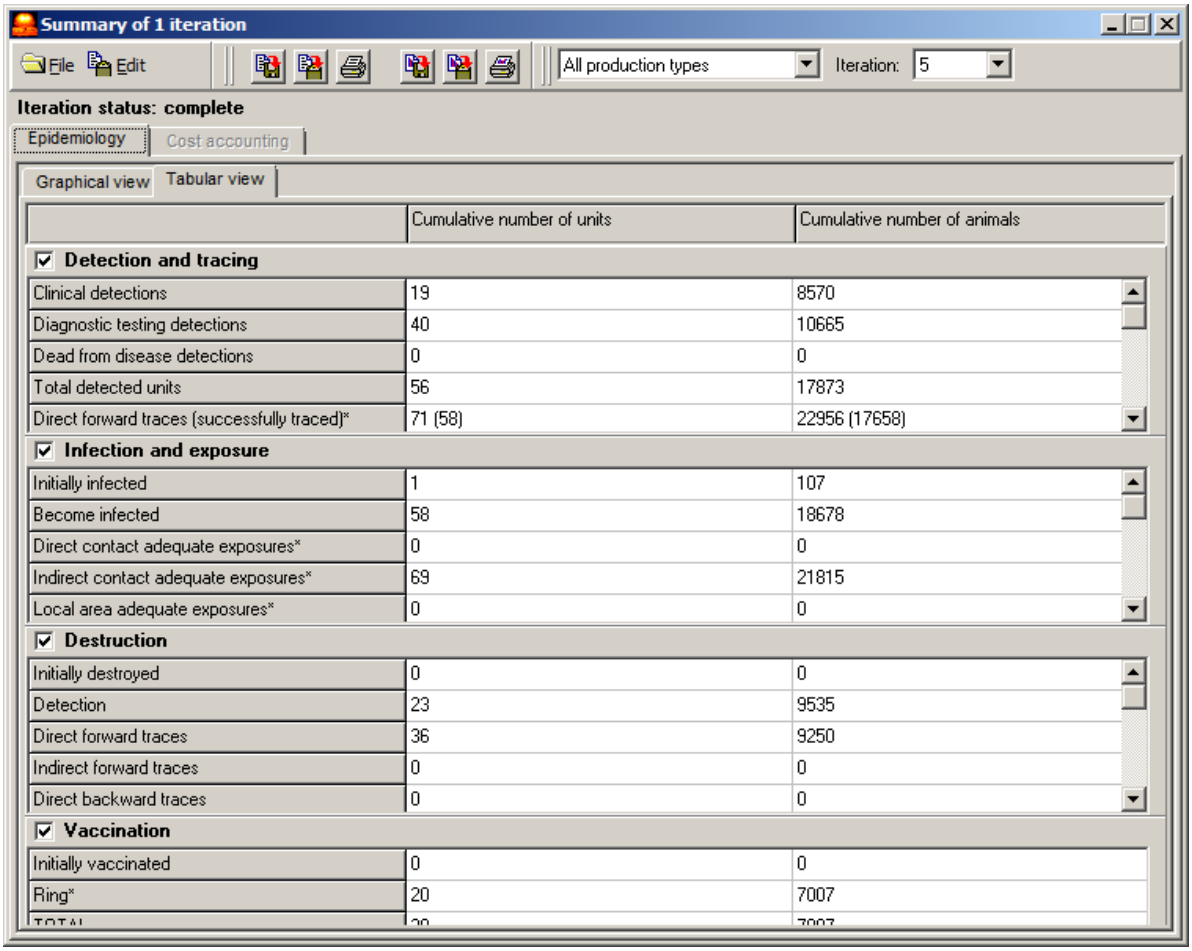


Figure 11-6. The **Epidemiology** tab (**Tabular view**) of the **Summary of 1 iteration** window.

11.2.3.1.1. Detection and tracing

The **Detection and tracing** section of the **Epidemiology** tab (Figure 11-6) shows the number of units (herds) detected based on the appearance of clinical signs, the number of units detected by diagnostic testing, the number of tested units that test positive or negative, and the number of units examined. The section also shows the numbers of trace investigations of direct or indirect contacts.

The total number of trace investigations attempted as well as the number of successful trace investigations is shown: the former number is given first, while the latter appears in parentheses as shown in Figure 11-6. Note that it is possible for a single unit to be identified multiple times by trace 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.

11.2.3.1.2. Infection and exposure

The **Infection and exposure** section of the **Epidemiology** tab (Figure 11-6) shows the number of units that are infected over the course of an iteration. Units may be infected at the beginning of a scenario **(Initially infected)**, or may become infected as a scenario runs. The numbers of units exposed by direct contact, indirect contact, local-area spread, and airborne spread are also shown. Note that units may be exposed multiple times by different sources, and that not all exposures will result in spread of disease.

11.2.3.1.3. Destruction

The **Destruction** section of the **Epidemiology** tab (Figure 11-6) shows the number of units that are destroyed (depopulated) over the course of an iteration. Destructions are broken down by all of the different 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 forward traces**, **Indirect forward traces, Direct backward traces**, and **Indirect backward traces**: see Section 9.9); and units preemptively destroyed because of proximity to a detected herd (**Ring** destruction: see Section 9.13). The number of animals in the destroyed herds is also shown.

11.2.3.1.4. Vaccination

The **Vaccination** section of the **Epidemiology** tab (Figure 11-6) 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 9.14). 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.

11.2.3.1.5. Reasons for infection

The **Reasons for infection** section of the **Epidemiology** tab (Figure 11-6) 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.

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.

11.2.3.1.6. Actual and apparent epidemic curves

The **Actual epidemic curve** and **Apparent epidemic curve** sections of the **Epidemiology** tab **Graphical view** tab sheet (Figure 11-7) 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 11.1). 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.

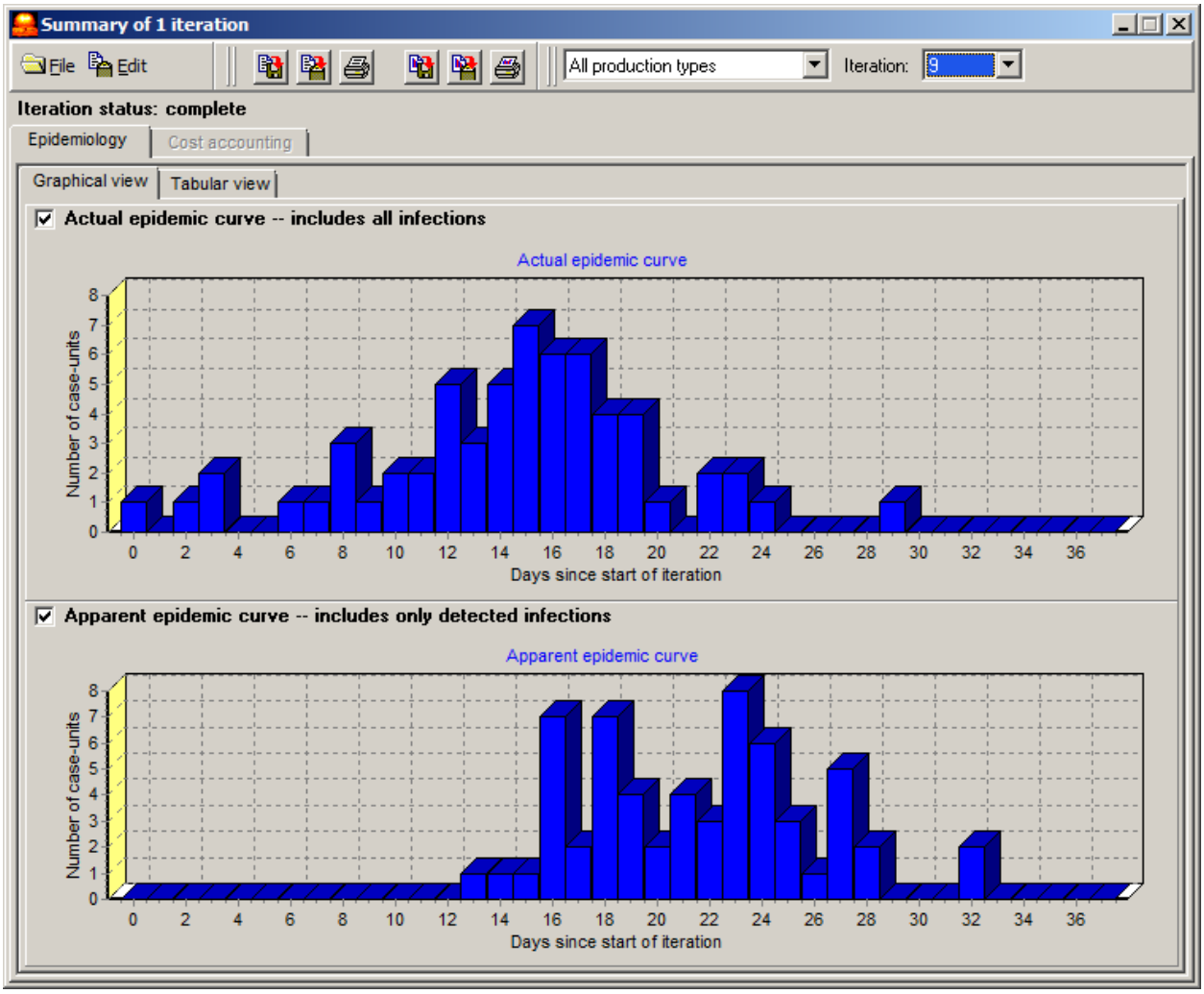
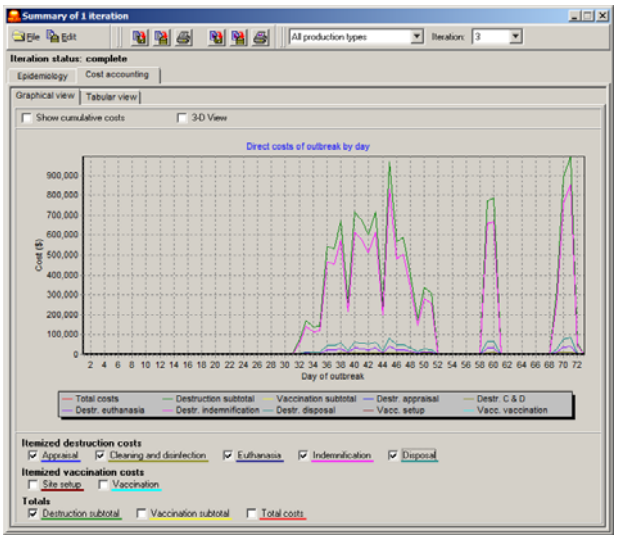


Figure 11-7. Actual and apparent epidemic curves.

11.2.3.2. Cost accounting outputs

If cost accounting was enabled when the scenario was set up (Section 9.15), the **Cost accounting** tab of the **Summary of 1 iteration** window will be activated (Figure 11-8). This tab has two tabs - for the **Table of daily costs (Tabular view)**, and the **Graph of daily costs (Graphical view)**.



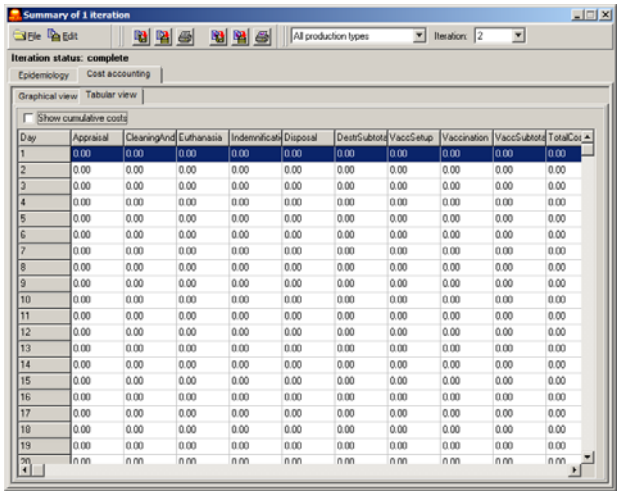


Figure 11-8. Cost accounting output views from one iteration.

The tab shows the **Graphical view** and **Tabular view** of **cost accounting** on the **Summary of 1 iteration** window.

11.2.3.2.1. 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 each of the categories used for cost accounting (Section 9.15). 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.

11.2.3.2.2. 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.

11.3. 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.

Box 11-2. 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 9.16).

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 9.5) and destruction of an infected unit. Several the events recorded by *NAADSM* is given in Table 11-1.

Table 11-1. Examples of the events recorded by *NAADSM*.

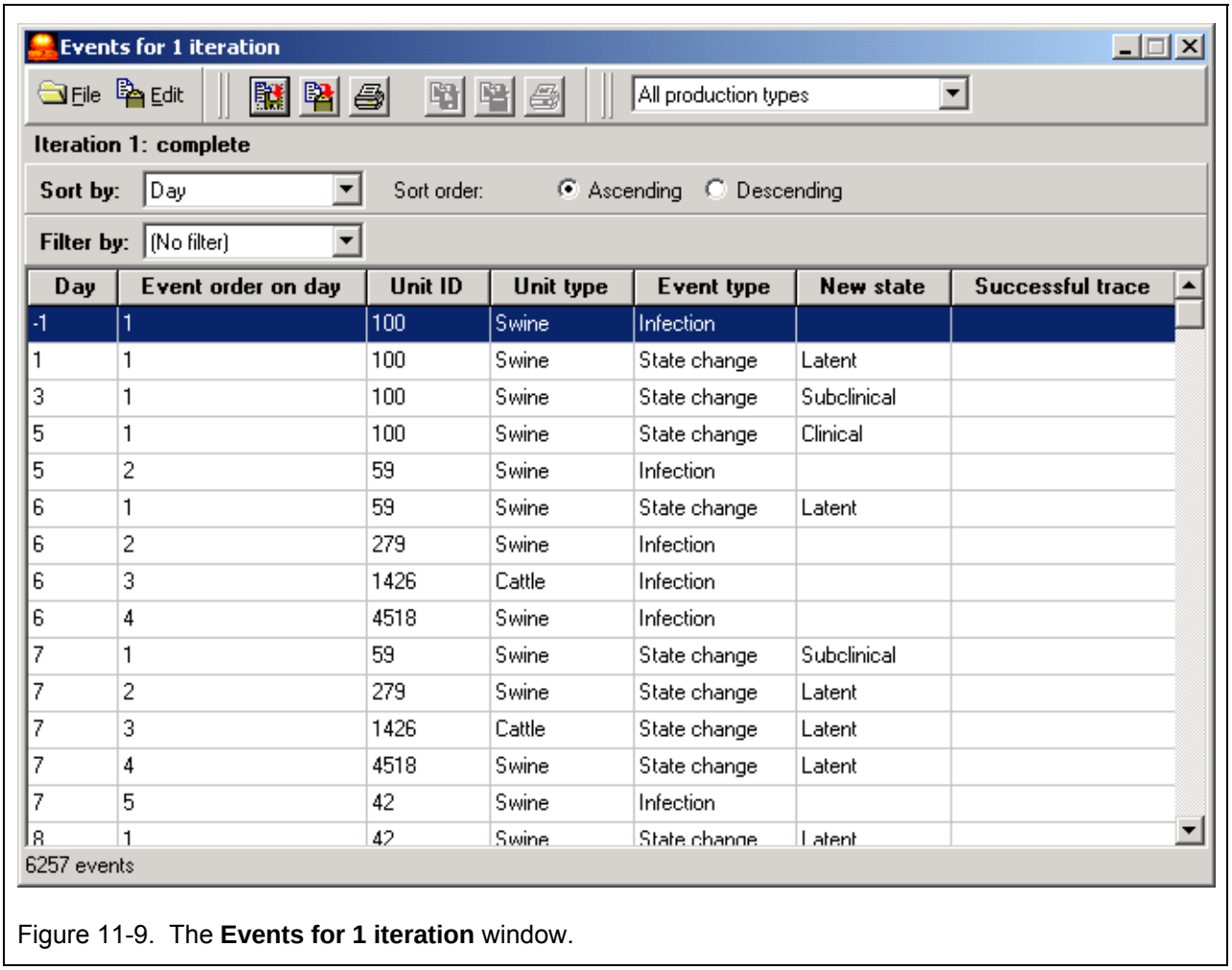
Event type	***Description***
State change	Any change in disease transition state, either by natural progression (*e.g.* "latent" to "subclinical" or "clinical" to "naturally immune") or by intervention (*e.g.* "susceptible" to "destroyed"). See Figure 9-15 for all possible state changes.
Infection	Susceptible units are infected upon effective exposure from a contagious herd.
Detection	Herds showing clinical signs may be detected according to the criteria described in Section 9.8.
Destruction	A destruction event is recorded on the day that a unit is destroyed for reasons described in Section 9.13. 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 a unit is vaccinated for reasons described in Section 9.14. 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 investigation of a direct contact (see Section 9.9).

Exposures involve two units and the possible transmission of disease. There are four types of exposures (mechanisms of disease spread) in *NAADSM*: these are direct contact, indirect contact, local-area spread, and airborne spread (see Section 9.6).

11.3.1. The Events for 1 iteration window

Output windows → Events for 1 iteration is used to open the **Events for 1 iteration** window (Figure 11-9). This window displays all of the events recorded for the most recent iteration of a scenario in a tabular format. Table 11-2 describes all of the columns in this table. **Error! Reference source not found.** 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.



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		
6	3	1426	Cattle	Infection		
6	4	4518	Swine	Infection		
7	1	59	Swine	State change	Subclinical	
7	2	279	Swine	State change	Latent	
7	3	1426	Cattle	State change	Latent	
7	4	4518	Swine	State change	Latent	
7	5	42	Swine	Infection		
8	1	42	Swine	State change	Latent	

Figure 11-9. The **Events for 1 iteration** window.

Table 11-2. 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 11-1.
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).

Box 11-3. Events on day "-1"

A close look at Figure 11-9 will show that the first event recorded occurred on day "-1" of the iteration in question. "-1" indicates that an event occurred before 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.

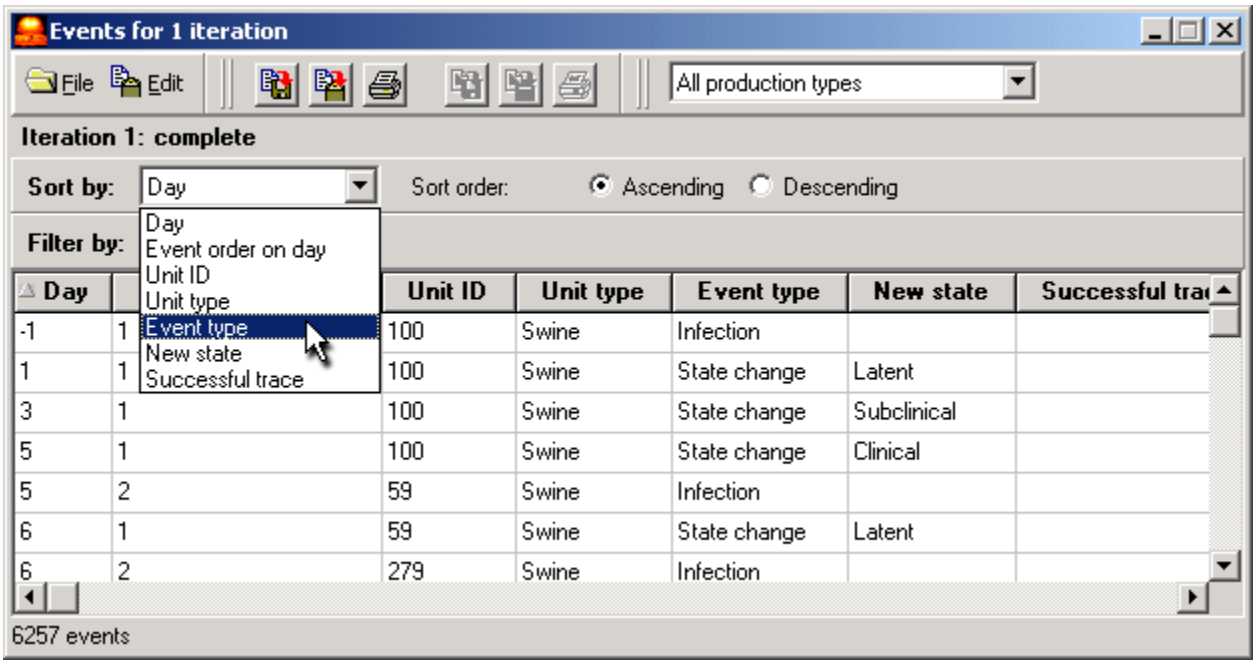
Day 1 in the events and exposures tables represents the first day simulated by *NAADSM* for an iteration.

Box 11-4. The order of events on a particular day

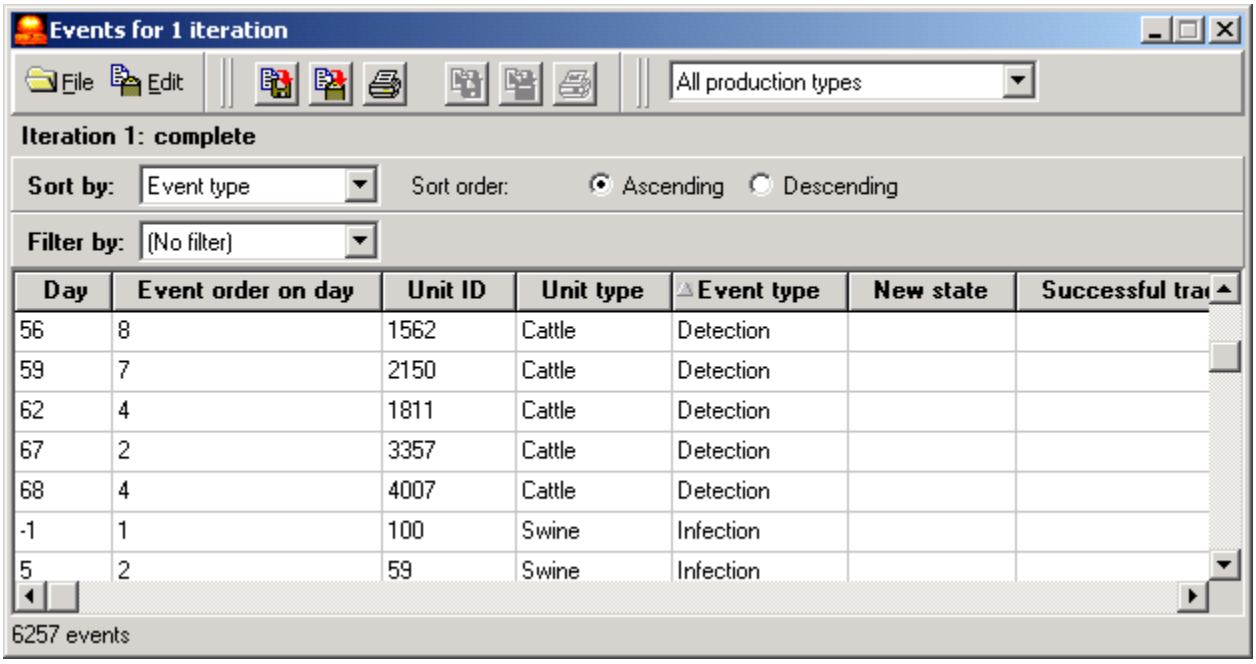
For each simulation day, tables like the one in Figure 11-9 will show the order of events as they occur. The ordering of events on a particular day is arbitrary. It is more appropriate to think of every event that occurs on a particular simulation day as occurring simultaneously than occurring in the precisely the order listed. For a further discussion of this topic, please see the section entitled "Priorities of action" in the *NAADSM* model description, which is available at <*http://www.naadsm.org/documentation*>.

11.3.1.1. 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 11-10a). 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 11-10b).



(a)



(b)

Figure 11-10. 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**.

11.3.1.2. 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 11-11a). 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 11-11b shows an example of the latter).

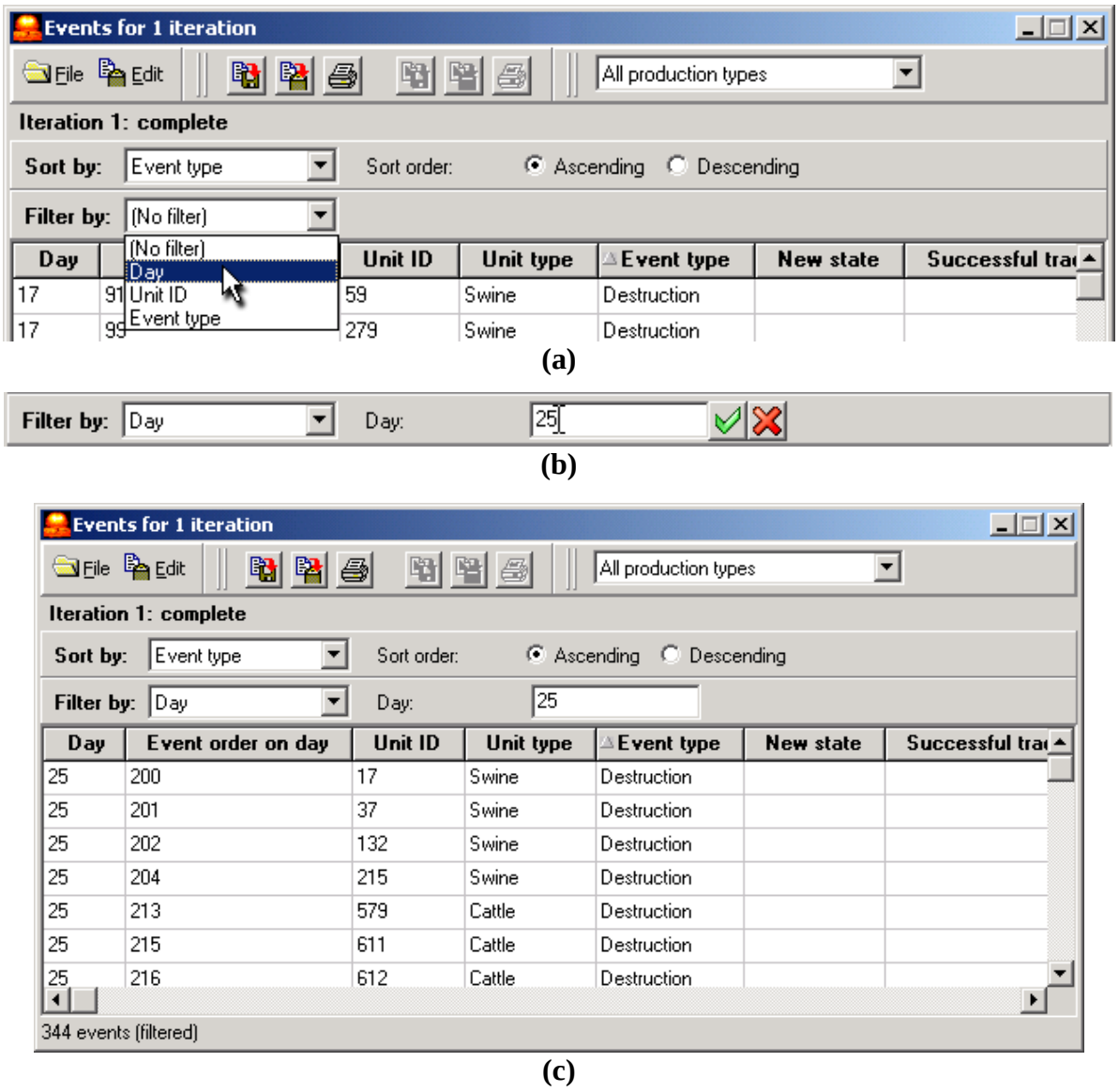


Figure 11-11. Filtering events by day.

(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.

11.3.2. The Exposures for and traces 1 iteration window

Output windows → Exposures and traces for 1 iteration is used to open the **Exposures and traces for 1 iteration** window (Figure 11-12). This window displays all of the exposures and traces recorded for the most recent iteration of a scenario in a tabular format. These events are listed on the simulation day in which they occur, and in the order on which they occur on each day (columns **Day** and **Event order** in the table).

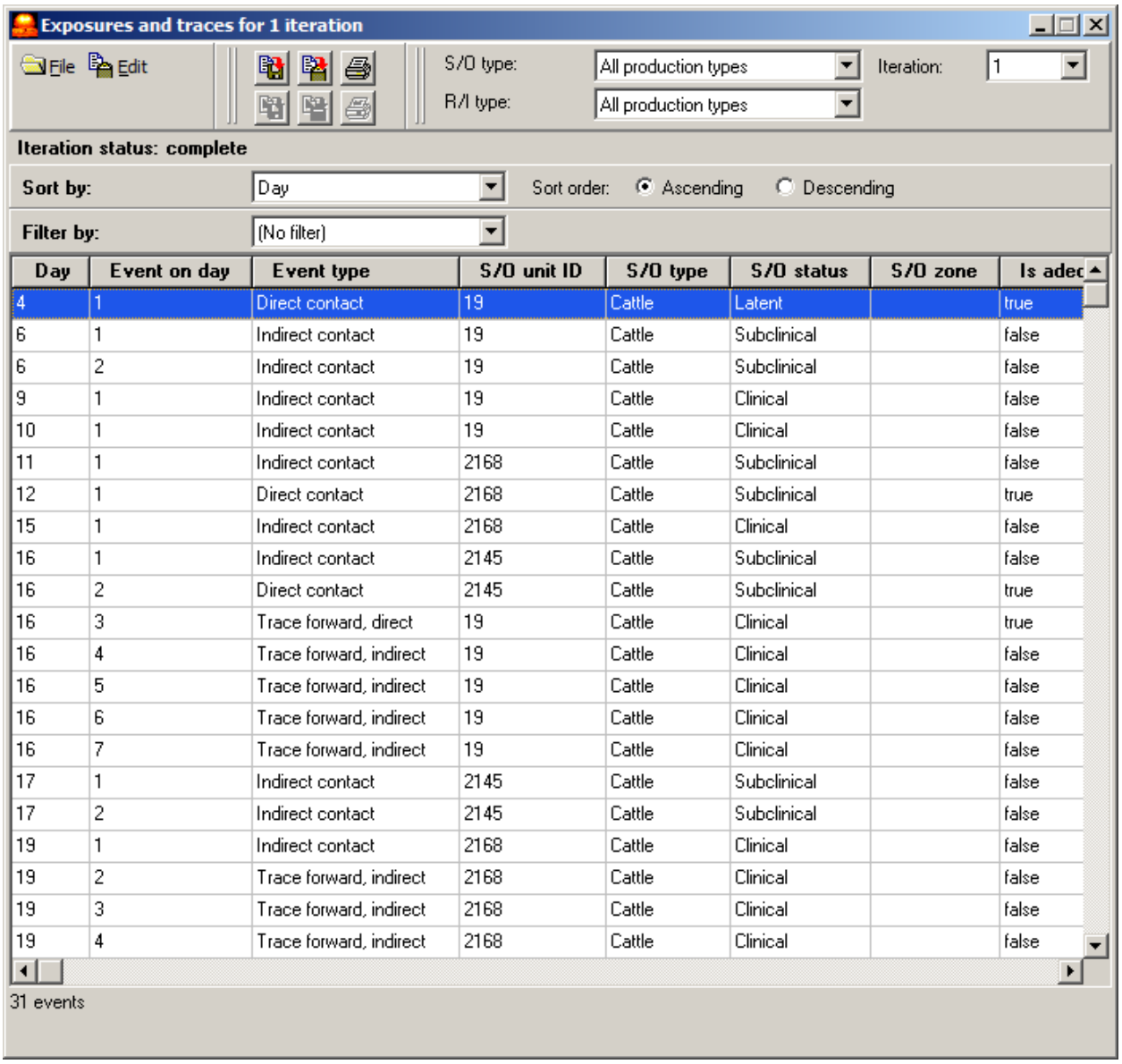


Figure 11-12. The **Exposures for 1 iteration** window.

11.3.2.1. Viewing exposures

Exposures are identified by their source and recipient units (see Section 9.6.1.1), which are designated "S" and "R" in the table. For each exposure, the ID, type, disease status, and zone of the source unit are recorded in the columns labeled **S/O unit ID**, **S/O unit type**, **S/O status**, and **S/O zone**. The properties of the recipient unit are given in columns labeled **R/I unit ID**, *etc*. Adequate exposures (Section 9.6.1.1) as well as inadequate exposures will be identified in the column **Is adequate/successful**.

Box 11-5. The order of exposures on a particular day

Recall from above that the order of events on any particular day is arbitrary, and that all exposures recorded for a particular day should be considered to occur simultaneously.

One practical significance of this characteristic is that a herd may be adequately exposed to disease by multiple sources on the same day. In such a case, it is inappropriate to think of the "first" exposure as the one that resulted in infection, when any one of these exposures could have resulted in infection on that day. In cases like this, *NAADSM* will record each exposure individually, but will record the exposed herd as having been infected only once.

When such cases arise in *NAADSM* 4, no attempt is made to attribute the infection to one particular source. Long-time users of *NAADSM* may recognize this as a change from earlier versions, in which the source of every infection was precisely (but completely arbitrarily) identified.

Box 11-6. Recording units exposed by local-area or airborne spread

Because of the very large number of units exposed by the local-area and airborne spread mechanisms, it is impractical to record these exposures. Only direct and indirect contacts are available in the **Exposures for 1 iteration** window.

11.3.2.2. Viewing traces

Each trace is identified by the unit at which the trace originated ("O") and the unit identified by the trace investigation ("I"). Recall from Section 9.9.1 that whether a trace origin or the unit identified by the trace is the source or recipient of contact will depend on the direction of the trace.

For each trace shown in the table in Figure 11-12, the origin unit is identified by its ID, type, status, and zone, in the columns labeled **S/O unit ID**, **S/O unit type**, **S/O status**, and **S/O zone**, respectively. Properties of the unit "at the other end" of the trace are given in columns labeled **R/I unit ID**, *etc*. A trace is successful (column **Is adequate/successful**) if the unit at the other end of a trace is successfully identified.

11.3.2.3. 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 (Sections 11.3.1.1 and 11.3.1.2).

11.4. 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 (Chapter 4). The windows discussed in this section allow users to explore the range of possible outcomes.

11.4.1. 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.

11.4.1.1. Epidemiological and cost accounting outputs

The **Output statistics** window (Figure 11-13) has several tabs. All of these tabs have the same features and are formatted identically: the only difference is the set of calculations that they display. The **Epidemiology** tab shows output statistics such as the number of units infected, the number of units depopulated, and the number of units vaccinated. The Cost accounting tab (which will be enabled only if cost accounting parameters were specified when the scenario was run: see Section 9.15) shows cost outputs. The **Zones/production types** tab summarizes outcomes like the number of days that units of particular production types spent in the different zones (Section 9.12). The **Zones** tab displays outputs such as the maximum area of each zone, and the day at which the maximum area was reached. The **Zones/production types** and **Zones** tabs will be enabled only if zones were used when the scenario was run. Each tab displays an output table and a histogram. These components are described in detail in the following sections.

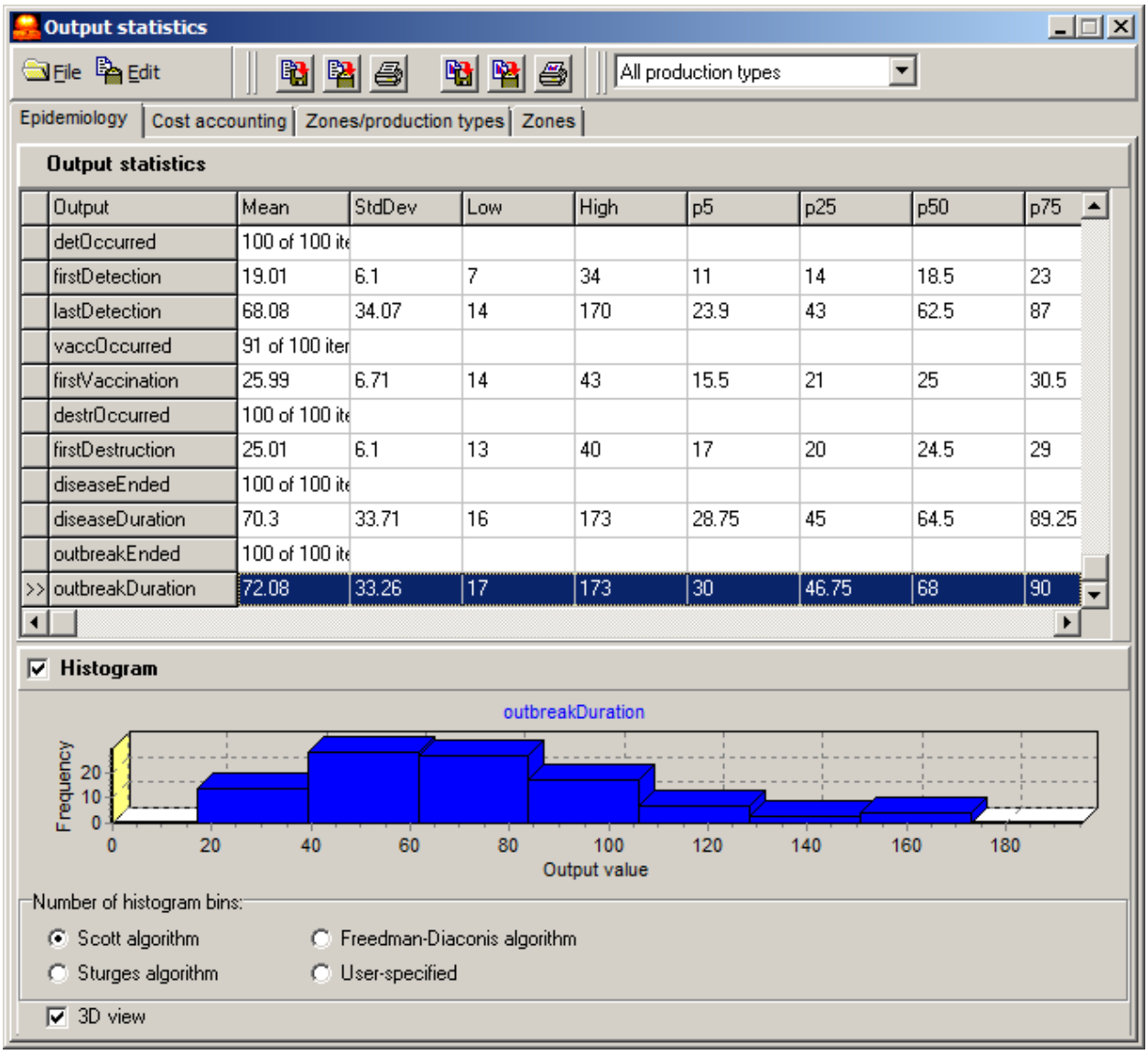


Figure 11-13. The **Output statistics** window.

The variable **diseaseDuration** is selected in the **Output statistics** table. Values of **diseaseDuration** from each iteration were used to generate the **Histogram**.

11.4.1.2. 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 11-14). Select a row in the output table to view a histogram for the output in that row (see Section 11.4.1.3).

Box 11-7. Variable definitions

Complete descriptions of the outputs on the **Epidemiology** and **Cost accounting** tabs are provided on the *NAADSM* website, at <*http://www.naadsm.org/documentation*>.

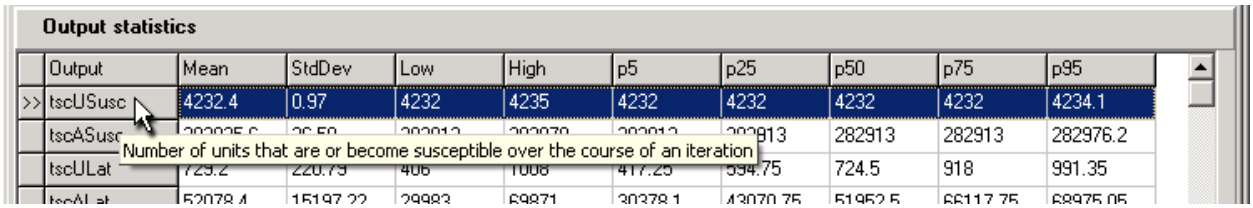


Figure 11-14. Displaying output definitions.

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 11-15). 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), **diseaseEnded** (the number of iterations in which the active spreading disease phase of the outbreak ended), and **outbreakEnded** (the number of iterations in which disease spread ended and all associated control activities were completed). These values reported for these variables show the count and the proportion of iterations for which the indicated condition was met.

detOccurred	50 of 50 iterations						
firstDetection	17.18	4.91	8	31	9	14.25	18
lastDetection	39.76	15.16	9	73	15.9	31.25	39.5
firstDetUInf	10.3	9.5	1	41	1	4	7
firstDetAInf	2904.74	2698.79	107	11987	107	1023.5	2187.5
vaccOccurred	40 of 50 iterations						
firstVaccination	23.15	4.32	15	33	15.95	19.75	23.5
destrOccurred	50 of 50 iterations						
firstDestruction	23.18	4.91	14	37	15	20.25	24
diseaseEnded	50 of 50 iterations						
diseaseDuration	40.96	14.02	16	71	19.35	32	41
outbreakEnded	50 of 50 iterations						
outbreakDuration	42.7	14.68	16	74	19.35	32.5	41.5

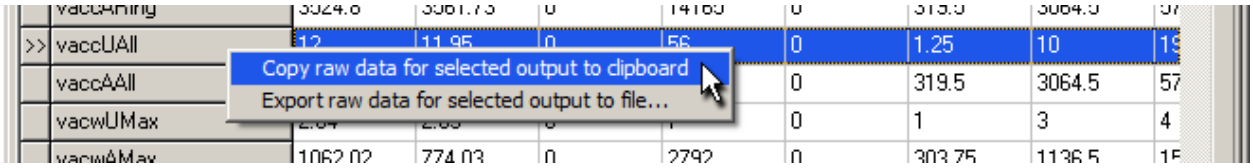
Figure 11-15. Qualitative variables on the **Epidemiology** tab.

These are labeled **detOccurred**, **vaccOccurred**, **destrOccurred**, **diseaseEnded**, and **outbreakEnded**. The statistical calculations shown for the other variables do not apply in these cases.

11.4.1.2.1. Obtaining the data used to generate summary statistics

The raw values used to calculate the summary statistics shown in the **Output statistics** window can be obtained by right-clicking on a row in the statistics table, and selecting either of the context menu options **Copy raw data for selected output to clipboard** or **Export raw data for selected output to file…** (Figure 11-16). From either the clipboard or from the resulting plain-text file, it should be possible to use raw values in any common statistical package.

Alternatively, raw values can be obtained directly from the *Microsoft Access*-compatible (`*.mdb`) scenario file.



(a)

```
## NAADSM raw data for all iterations
## Application version: 4.0.11-20121023
## Date: 12/15/12 2:59:57 PM
## Scenario file: C:\Research\SampleScenario.mdb
## Production type: All production types
## Output name: vaccUAll

18
0
19
10
1
34
5
9
4
14
1
3
11
```

(b)

Figure 11-16. Obtaining raw values used to generate summary statistics.

(a) The context menu options appear when a row in the statistic table is right-clicked. **(b)** The **Export raw data for selected output to file...**option can be used to generate a plain-text file with the format shown.

11.4.1.3. 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 11-17). *NAADSM* provides support for three commonly used algorithms for generating the number of histogram bins, based on work conducted by Sturges (1926), Scott (1979), and Freedman and Diaconis (1981). Users may also specify the number of desired histogram bins.

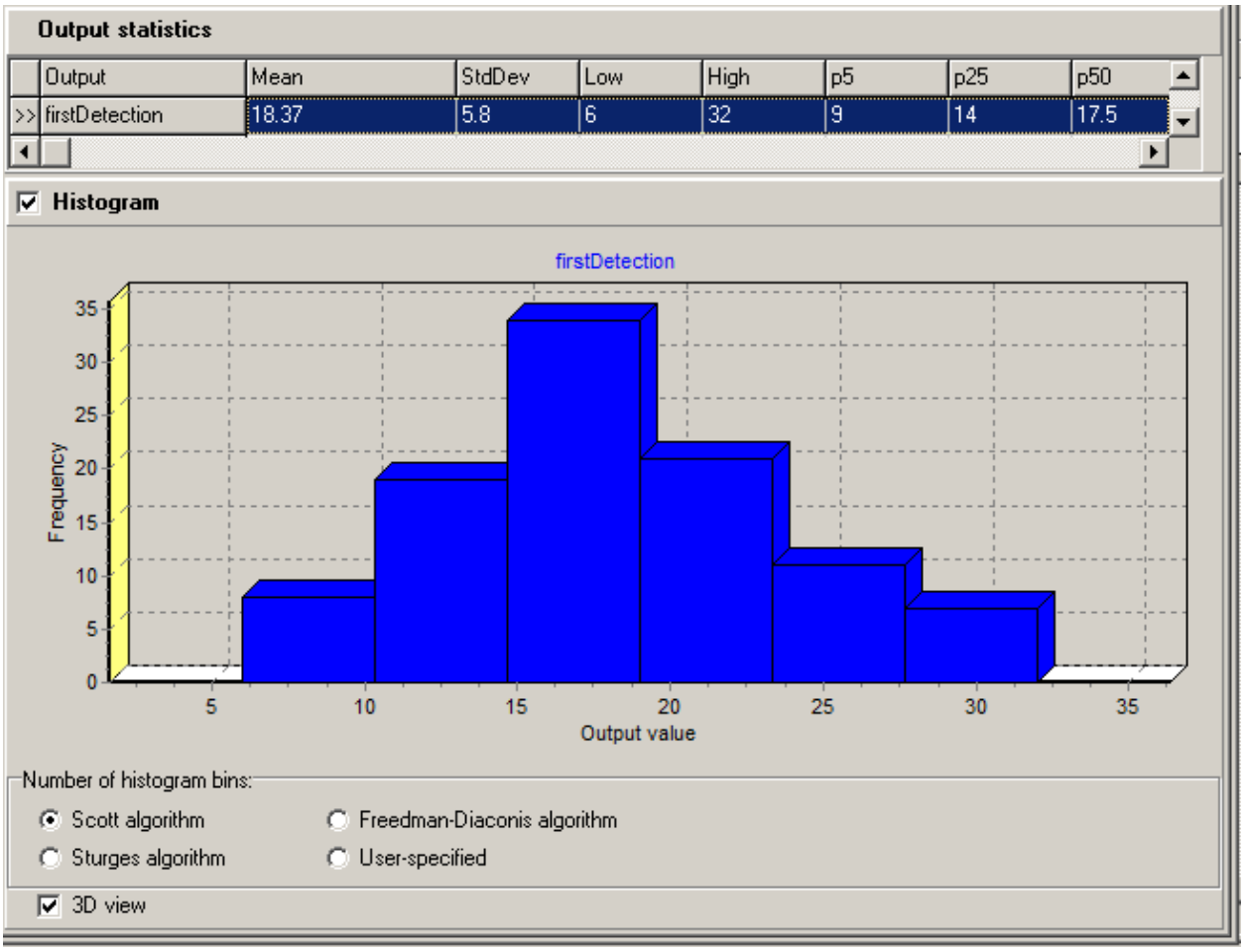


Figure 11-17. Histograms generated for **firstDetection** data from 50 iterations.

Use the buttons near the bottom of the window to select an algorithm for determining the number of histogram bins to display.

11.4.2. The Summary epidemic curves window

An epidemic curve can be generated pretty easily for a single disease outbreak. Typical output of a *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 a *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.

11.4.2.1. Summary epidemic curves by example

Table 11-3 shows output from a *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 11-18. Each series in this figure represents an epidemic curve for a single iteration.

Figure 11-18. Epidemic curves (number of new cases by day) generated for three of the individual iterations shown in Table 11-3.

The lower portion of Table 11-3 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 11-3, 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 11-19 shows "epidemic curves" drawn from the summary statistics of Table 11-3.

Table 11-3. 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 11-18.
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 11-19. 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	9	2	2	2	1	0	19
3	2	0	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	1	0	6	6	10	10	5	5	5	7	4	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	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	7	6	9	5	2	1	3	2	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
p95	3	3	3	3	3	2.05	10	10	10	10	10	9.05	10	10	10	4	2.1	2.05	2.1	0	
p50	2	1	1.5	1	1.5	0.5	7	6	7	7	7	7	6	6	4.5	2	0	0	0	0	
Mean	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	
pP5	1	0	0	0	0	0	4.95	4	4	4.95	4	4	0.95	2.85	0	0	0	0	0	0	

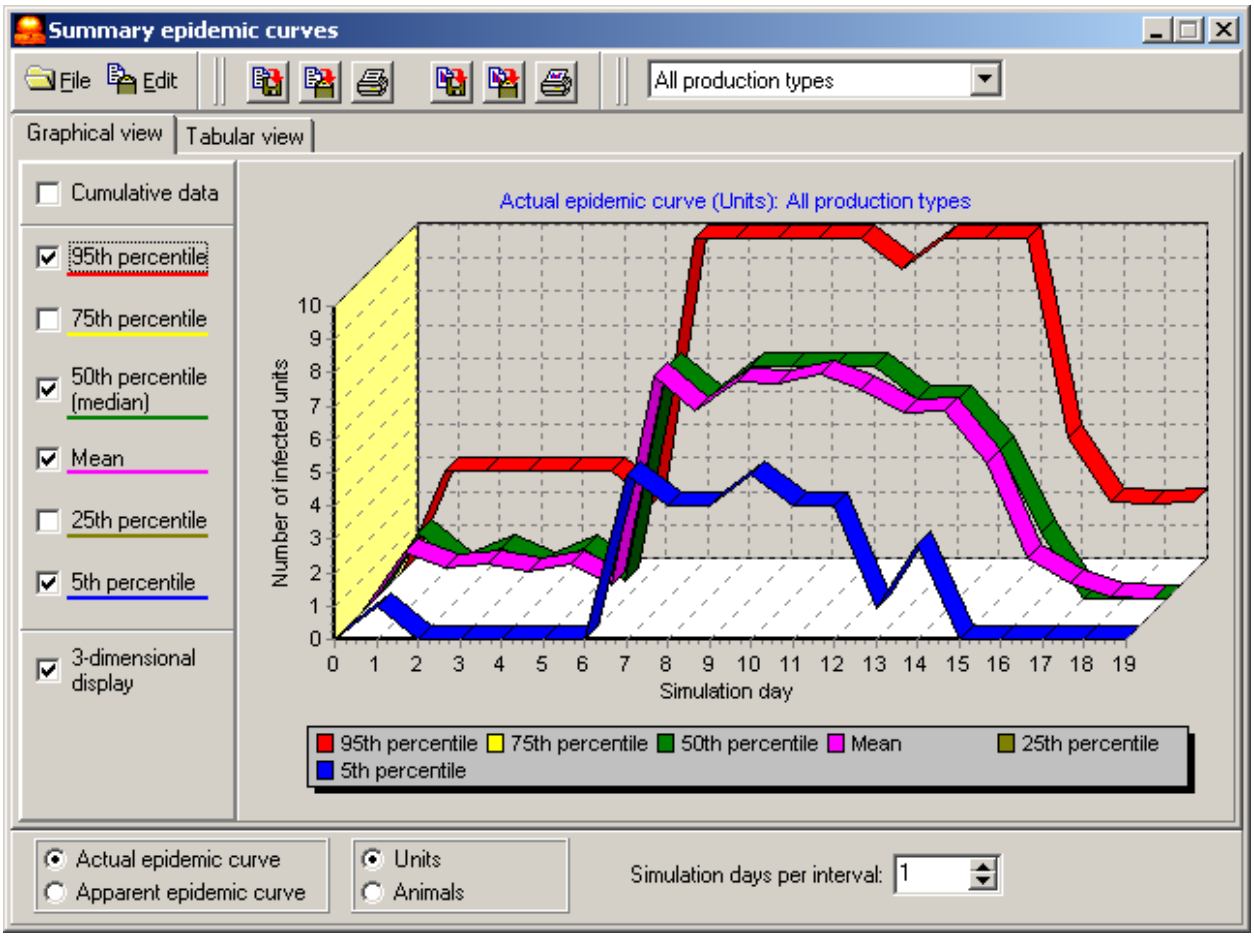


Figure 11-19. The **Summary epidemic curves** window, showing summary curves for the data from Table 11-3.

Box 11-8. 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.

11.4.2.2. 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 11-19). Select one of 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. Check the **3-dimensional display** box to toggle between a three-dimensional and a "flat" view of the graph.

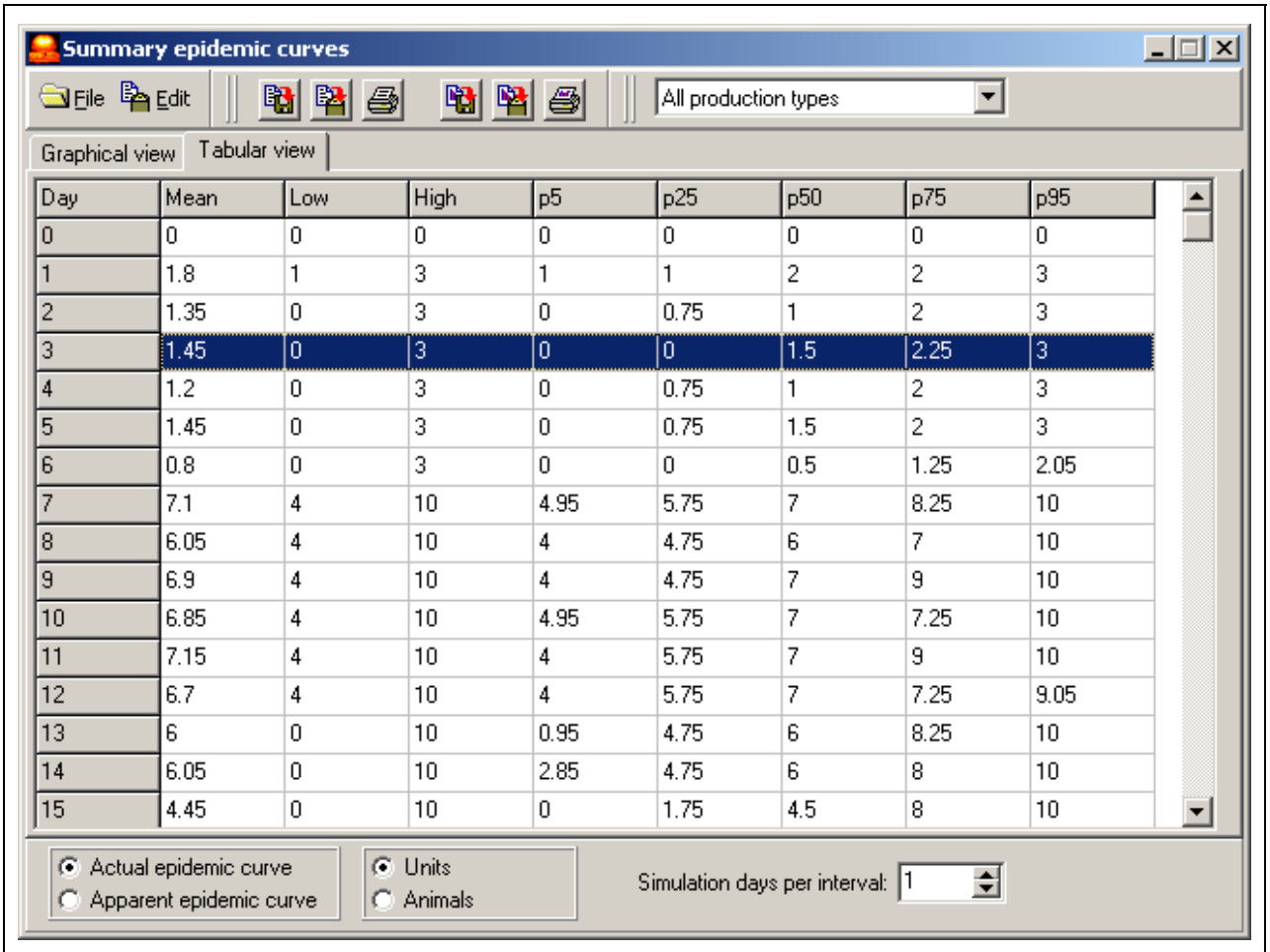
Users may choose to view summary curves for all disease cases (**Actual epidemic curve**) or for only detected cases (**Apparent epidemic curve**). Numbers of herds (units) or animals may be selected.

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 11.5).

11.4.2.3. Using the Summary epidemic curves tabular view

The **Tabular view** tab of the **Summary epidemic curves** window (Figure 11-20) 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.



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 11-20. The **Tabular view** tab of the **Summary epidemic curves** window.

11.5. 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 11-21. The production type dropdown menu is used to change the production type for which data is displayed (Section 11.5.1). The **File** menu, **Edit** menu, and toolbar buttons are used for saving, copying, and printing data or graphics (Sections 11.5.2 and 11.5.3).

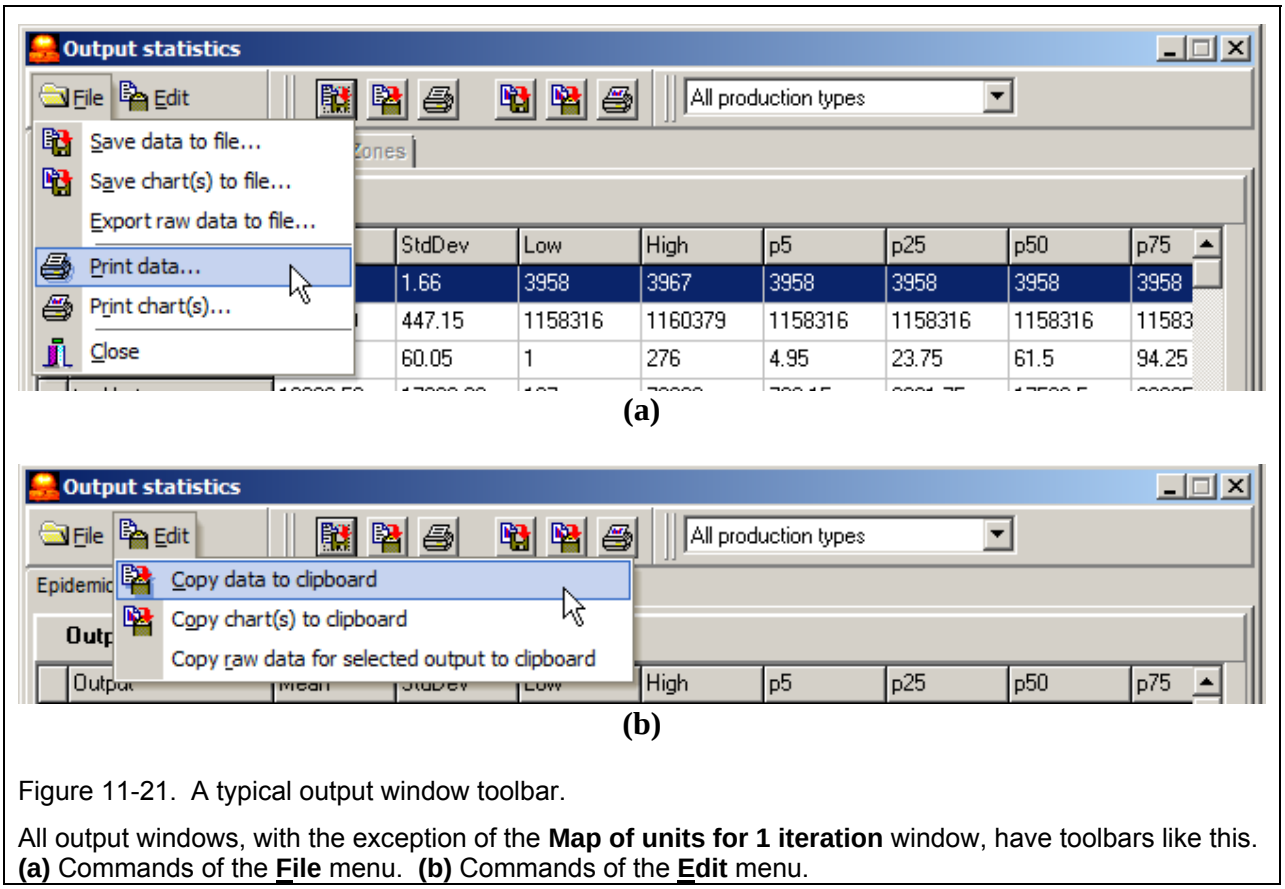


Figure 11-21. A typical output window toolbar.

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.

11.5.1. 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 and traces for 1 iteration** window (Section 11.3.2) has two production type dropdown menus: the upper menu allows you to select a source production type for the displayed exposures (or the origin unit for displayed traces), while the lower menu allows you to select a recipient production type for exposures (or the identified unit for displayed traces) (Figure 11-12).

11.5.2. 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 11.2.3), 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.

11.5.3. 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 11.4.1), 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.

11.6. Arranging the output windows

The three commands (Cascade, Arrange, Close all windows) at the bottom of the **Output windows** menu (Figure 11-1) 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.

11.6.1. 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 11-22a).

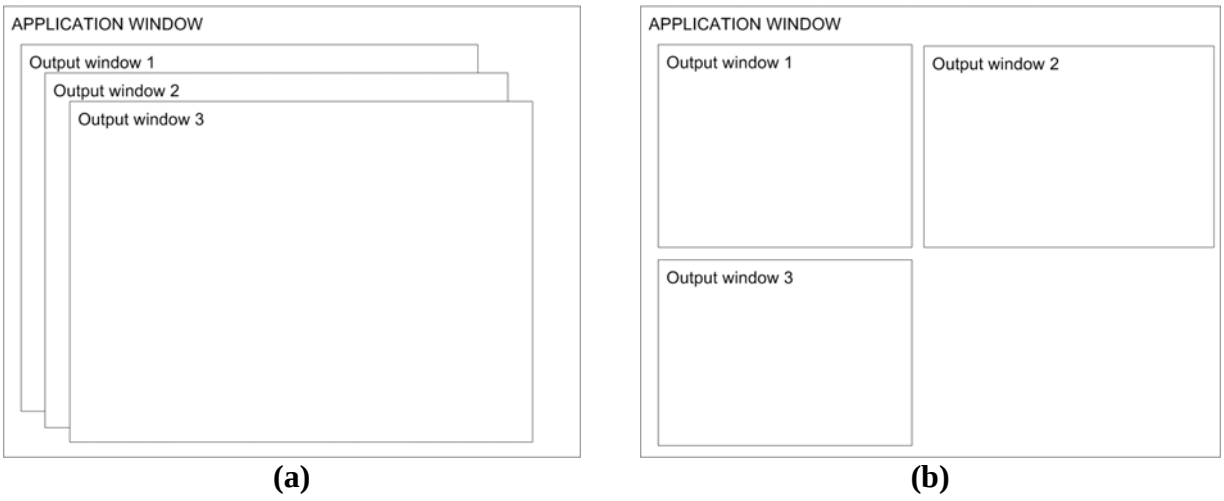


Figure 11-22. Schematic diagram showing output window arrangement.
(a) Cascade or **(b) Arrange** style arrangements.

11.6.2. 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 11-22b).

11.6.3. The Close all windows command

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

12. The Help menu

Three 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 12-1).

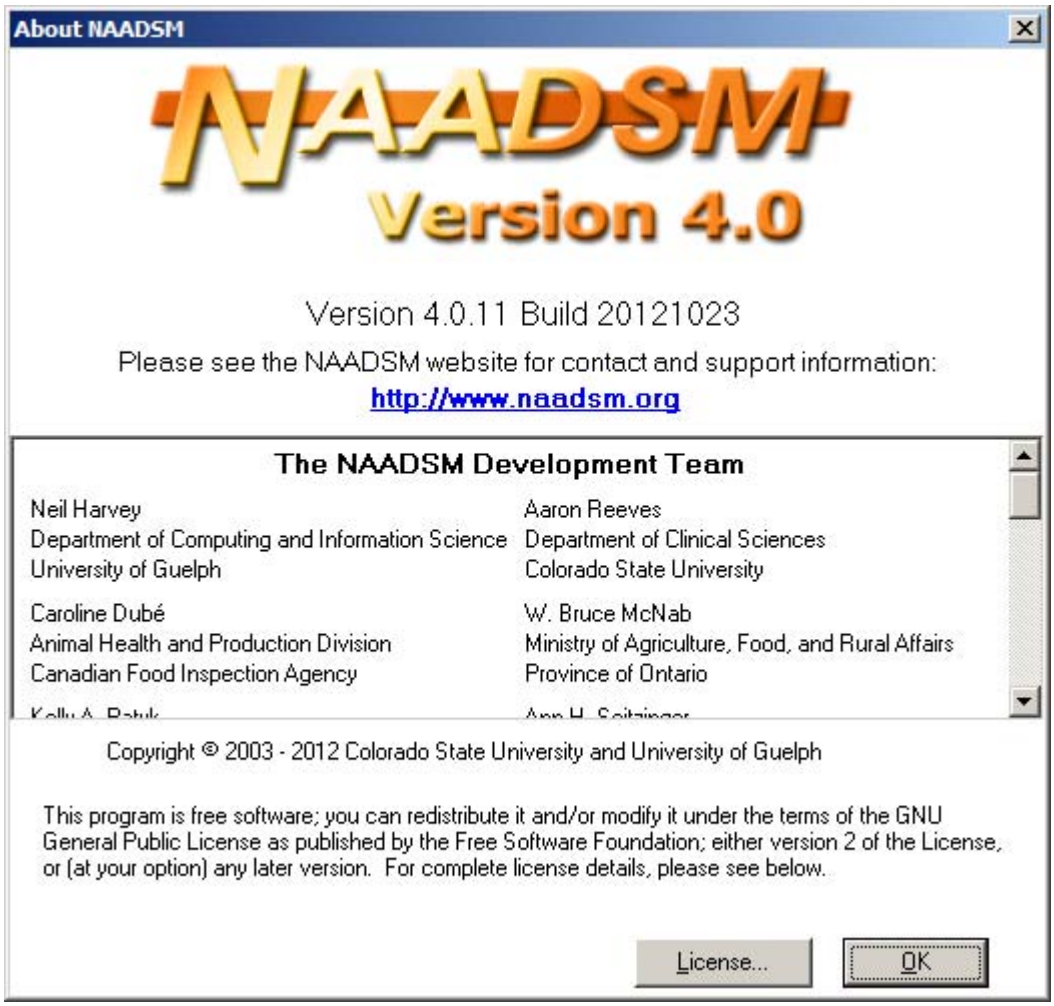


Figure 12-1. The **About NAADSM** screen.

Box 12-1. NAADSM *version numbers*

The *NAADSM* version and build identifiers, as shown in Figure 12-1, convey a great deal of information. Each major version number (for example, "4.0") is associated with one particular form of the conceptual model behind the application (see <*http://www.naadsm.org/documentation/specification*>). 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 ".10" in "4.0.10") 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, 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*.

Box 12-2. 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!

Help → Go to the NAADSM online support forum will open your default Internet browser and take you automatically to the support section of *NAADSM* website, <*http://www.naadsm.org/forum*>. *NAADSM* users are strongly encouraged to post questions, comments, or requests for support to this site, which is open to the public without the need for any kind of registration.

13. Useful references

Carpenter, T.E., Thurmond, M.C., Bates, T.W., 2004. A simulation model of intraherd transmission of foot and mouth disease with special reference to disease spread before and after clinical diagnosis. *J. Vet. Diagn. Invest*. 16:11–16.

Freedman, D., and Diaconis, P., 1981. On the histogram as a density estimator: L_2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* 57:453–476.

Harvey, N., Reeves, A., Schoenbaum, M.A., Zagmutt-Vergara, F.J., Dubé, C., Hill, A.E., Corso, B.A., McNab, W.B., Cartwright, C.I., Salman, M.D., 2007. The *North American Animal Disease Spread Model*: A simulation model to assist decision making in evaluating animal disease incursions. *Prev. Vet. Med*. 82:176–197.

Hill, A., Reeves, A., 2006. *User's Guide for the North American Animal Disease Spread Model 3.0*, 2nd ed. Animal Population Health Institute, Colorado State University, Fort Collins, Colorado, USA. Available at <*http://www.naadsm.org/documentation*>.

Kostova-Vassilevska, T., 2004. On the use of models to assess foot-and-mouth disease transmission and control. Lawrence Livermore National Laboratory, Livermore, California, USA.

Law, A.M., 2006. Simulation modeling and analysis, 4th ed. McGraw-Hill Higher Education, Columbus, Ohio, USA.

NAADSM Development Team, 2009. Versioning, stability, verification, and validation of *NAADSM*. Web page <*http://naadsm.org/documentation/techpapers*>.

NAADSM Development Team, 2010a. *Model Description: North American Animal Disease Spread Model 3.2*, version 1.2.1. (N. Harvey, A. Reeves, Eds.). Web page <*http://www.naadsm.org/documentation/specification*>..

NAADSM Development Team, 2010b. *Model Description: North American Animal Disease Spread Model 3.1*, version 1.1.2. (N. Harvey, A. Reeves, Eds.). Web page <*http://www.naadsm.org/documentation/specification*>.

NAADSM Development Team, 2012. *Model description for the North American Animal Disease Spread Model 4.0*, version 2.0.0. (N. Harvey, A. Reeves, Eds.) Web page <*http://www.naadsm.org/documentation/specification*>.

Palisade Corporation, 2008. *@Risk* version 5.0.1. <*http://www.palisade.com*>.

Patyk, K.A., Helm, J., Martin, M.K., Forde-Folle, K.N., Olea-Popelka, F.J., Hokanson, J.E., Fingerlin, T., Reeves, A., *submitted for publication*. An epidemiologic simulation model of

the spread and control of highly pathogenic avian influenza (H5N1) among commercial and backyard poultry flocks in South Carolina, United States.

Reeves, A., 2011. *Understanding vaccination queues in NAADSM 3 and NAADSM 4*. Animal Population Health Institute, Colorado State University, Fort Collins, Colorado, USA. Available at <*http://www.naadsm.org/techpapers*>.

Reeves, A., 2012. *Developing unit-level disease state and within-unit prevalence parameters for NAADSM.* Colorado State University, Fort Collins, Colorado, USA. Available at <*http://www.naadsm.org/techpapers*>.

Reeves, A., Hupalo, R., 2012. *Calculating probabilities of local-area and airborne disease spread in NAADSM 4*. Colorado State University, Fort Collins, Colorado, USA. Available at <*http://www.naadsm.org/techpapers*>.

Reeves, A., Hupalo, R., Patyk, K.A., Hill, A.E., 2012. *User's Guide for the North American Animal Disease Spread Model 3.2*, 3rd ed. Colorado State University, Fort Collins, Colorado, USA. Available at <*http://www.naadsm.org/documentation*>.

Reeves, A., Salman, M.D., Hill, A.E., 2011. Approaches for evaluating veterinary epidemiological models: verification, validation and limitations. *Rev. Sci. Tech. Off. Int. Epiz*. 30:499–512.

Reeves, A., Talbert, M., Salman, M.D., Hill, A.E., *submitted for publication*. Development of a stochastic, individual-based modeling framework for within-unit transmission of highly infectious animal diseases.

Savill, N.J., Shaw, D.J., Deardon, R., Tildesley, M.J., Keeling, M.J., Woolhouse, M.E.J., Brooks, S.P., Grenfell, B.T., 2007. Effect of data quality on estimates of farm infectiousness trends in the UK 2001 foot-and-mouth disease epidemic. *J. R. Soc. Interface* 4:235–241.

Schoenbaum, M.A., 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.

Scott, D.W., 1979. On optimal and data-based histograms. *Biometrica* 66:605–610.

Sturges, H.A., 1926. The choice of a class interval. *J. Am. Stat. Assoc.* 21:65–66.

Vose, D., 2000. Risk analysis: A quantitative guide, 2nd ed. John Wiley & Sons, New York, New York, USA.

Appendix A. Probability density functions supported by *NAADSM*

As described in Chapter 4 and Section 7.1.5, many parameters in a *NAADSM* scenario are probability density functions (*pdf*s). 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 23 general *pdf* types, described in the following sections, along with the parameter order expected by *NAADSM*. Some distributions are more suitable to some applications than others, but all are provided to ensure maximum flexibility to model users. 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 *pdf*s. A discussion of distribution fitting is beyond the scope of this guide, but readers are referred to other sources (Law, 2006; Vose, 2000).

Most 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 9.5.1), values obtained from these distributions will be rounded to the nearest whole day.

A1. The Beta distribution

A Beta distribution (Figure A-1) is a continuous distribution 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.

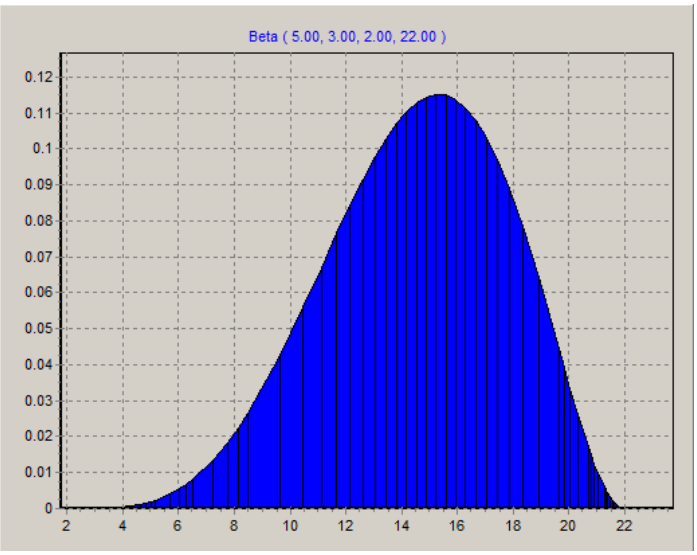


Figure A-1. A sample Beta distribution.

A2. The BetaPERT distribution

The BetaPERT distribution (Figure A-2) is a continuous distribution defined by its minimum, its most likely value (mode), and its maximum. In this way, the BetaPERT distribution is similar to the Triangular distribution (Section A21).

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, 2008). 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:

$$\mu = \frac{Min + 4Mode + Max}{6}$$

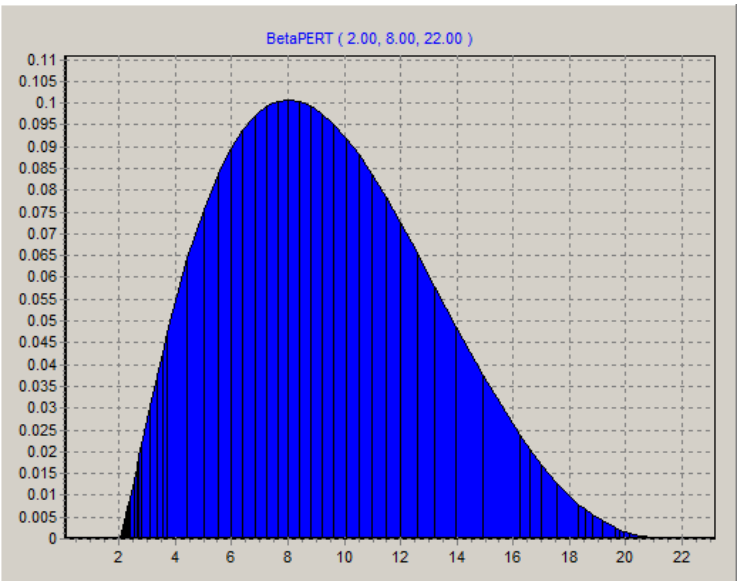


Figure A-2. A sample BetaPERT distribution.

A3. The Bernoulli distribution

The Bernoulli distribution is a discrete distribution used to model whether an event will occur. A single parameter *p* represents the probability of the event. The occurrence of getting a head on one toss of a coin could be modeled with the distribution Bernoulli(0.5), or the occurrence of getting a 6 on a single roll of die could be modeled with Bernoulli(0.1667). The outcome of a Bernoulli trial is always 0 or 1. The Bernoulli distribution is a special case of the Binomial distribution (Section A4), in which the number of trials is 1.

The probability density function *f(x)* for a Bernoulli distribution is calculated as:

$$f(x) = p^x (1-p)^{1-x}$$

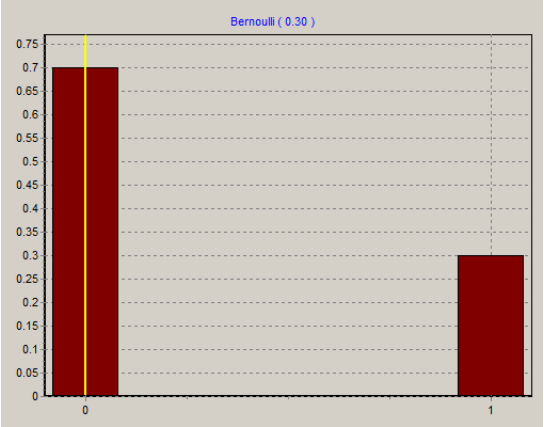


Figure A-3. A sample Bernoulli distribution.

A4. The Binomial distribution

The Binomial distribution is a discrete distribution used most often to model the number of *x* successes from *n* independent trials where there is a probability *p* of success in each trial (Figure A-4). It is used when there are exactly two mutually exclusive outcomes of a trial. The probability density function *f(x)* is calculated as:

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x}$$

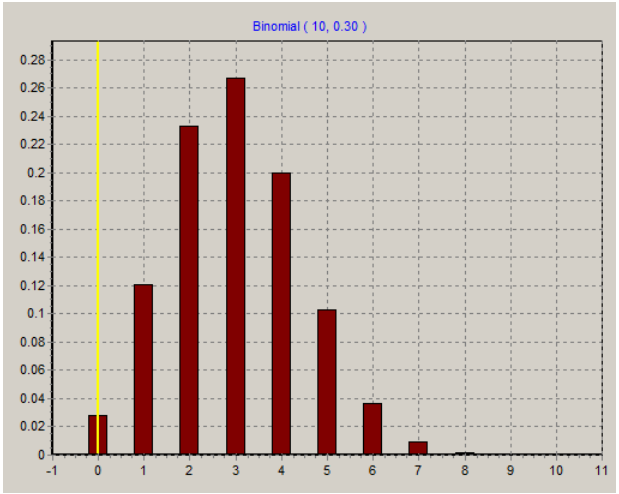


Figure A-4. A sample Binomial distribution.

A5. The Discreet Uniform

The Discreet Uniform distribution, sometimes called the "equally likely outcomes" distribution, has a set of *n* elements where each element *i* has the same probability of occurring (Figure A-5). A simple example of this distribution is the outcome of throwing one fair die. The probability density function *f(x)* is calculated as:

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x}$$

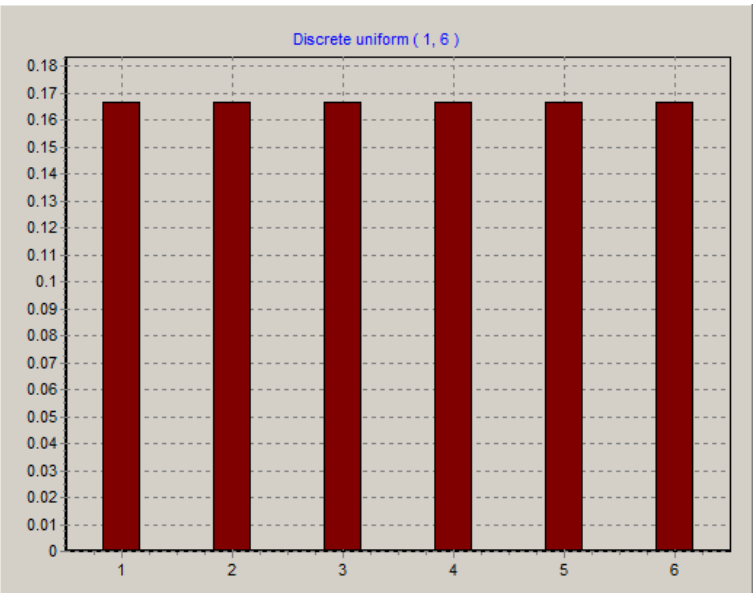


Figure A-5. A sample Discreet Uniform distribution.

As noted above, *NAADSM* sometimes rounds values from continuous distributions to the nearest integer value, when they are applied to discrete values like the duration in days of a disease state. For parameters like these, the Discrete Uniform distribution might be a better choice than the Uniform distribution discussed in Section A22, to prevent excessive error due to rounding.

A6. The Exponential distribution

An Exponential distribution (Figure A-6) is a continuous, highly skewed distribution defined by its mean μ, which must be greater than 0. The probability density function *f(x)* is calculated as:

$$f(x) = \frac{e^{-x/\mu}}{\mu}$$

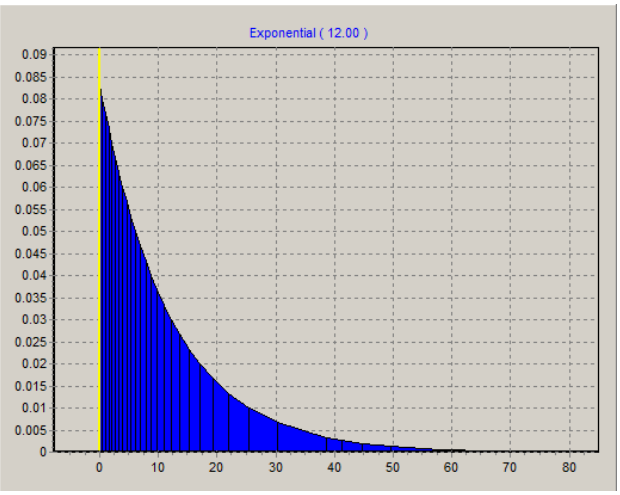


Figure A-6. A sample Exponential distribution.

A7. The Gamma distribution

A Gamma distribution (Figure A-7) is continuous, and is defined by two parameters: its shape (α), and its scale (β), where α and β must be greater than 0. Gamma distributions can take a wide variety of shapes, depending on the values of the shape and scale parameters. This distribution has a mean of α×β. 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.

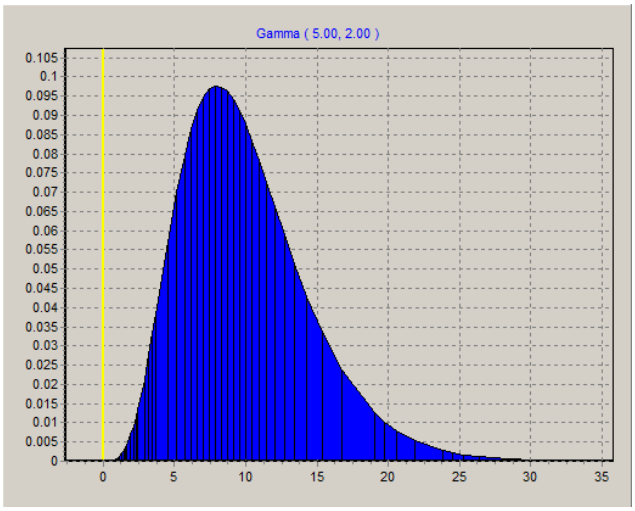


Figure A-7. A sample Gamma distribution.

A8. The Gaussian (Normal) distribution

A Gaussian or Normal distribution (Figure A-8) is a continuous, 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:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

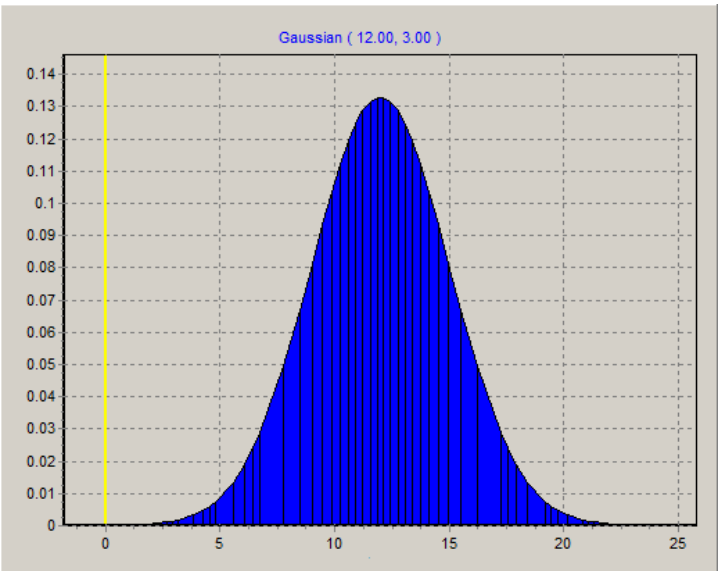


Figure A-8. A sample Gaussian distribution.

A9. The Histogram distribution

The Histogram distribution (Figure A-9) is a continuous empirical distribution: that is, a histogram distribution directly makes use of data to define its shape and properties, rather than one that is mathematically defined by a formula and its parameters. The probability density function *f(x)* is calculated from the set of minimum and maximum values for each histogram bin as follows:

$$f(x_i) = p_i \quad \text{where}$$

$$x_i = i\frac{\max - \min}{n} + \min$$

and the p_i are normalized so that

$$1 = \frac{n\sum p_i}{\max - \min}$$

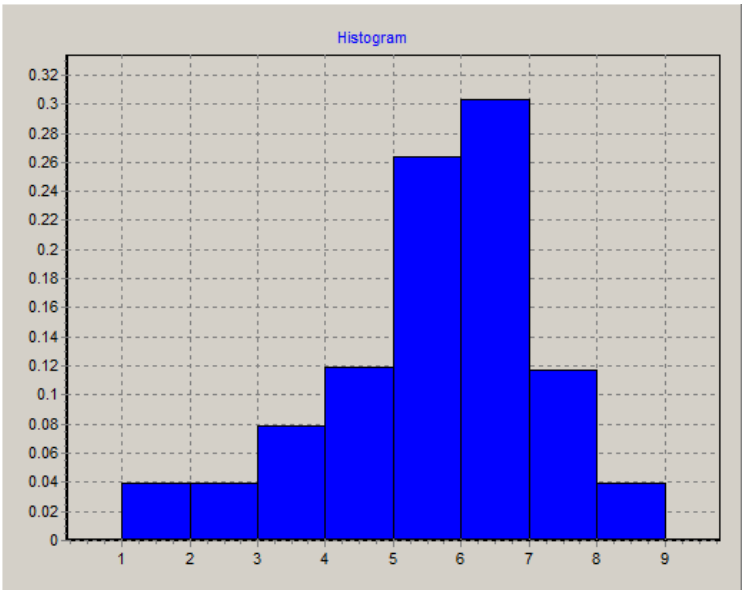


Figure A-9. A sample Histogram distribution.

Histogram distributions are discussed in greater detail in Section 7.1.5.4. They are particularly useful in *NAADSM* for generating data-derived distributions when theoretical distributions cannot be readily fit to data using conventional fitting techniques (Law, 2006; Vose, 2000).

A10. The Hypergeometric distribution

A Hypergeometric distribution (Figure A-10) is a discrete distribution commonly used to estimate the number of items of type *X* in sample *n* when the sample is drawn from population *M* that has *D* items of type *X*. For example, the number of infected animals in a shipment of animals selected at random from a herd of a particular size with a known prevalence of disease could be modeled using a hypergeometric distribution: in this example, *n* is the number animals in the shipment, *M* is the herd size, and *D* is the herd size times the prevalence. The probability density function *f(x)* for a Hypergeometric distribution is calculated as:

$$f(x) = \frac{\binom{D}{x}\binom{M-D}{n-x}}{\binom{M}{n}}$$

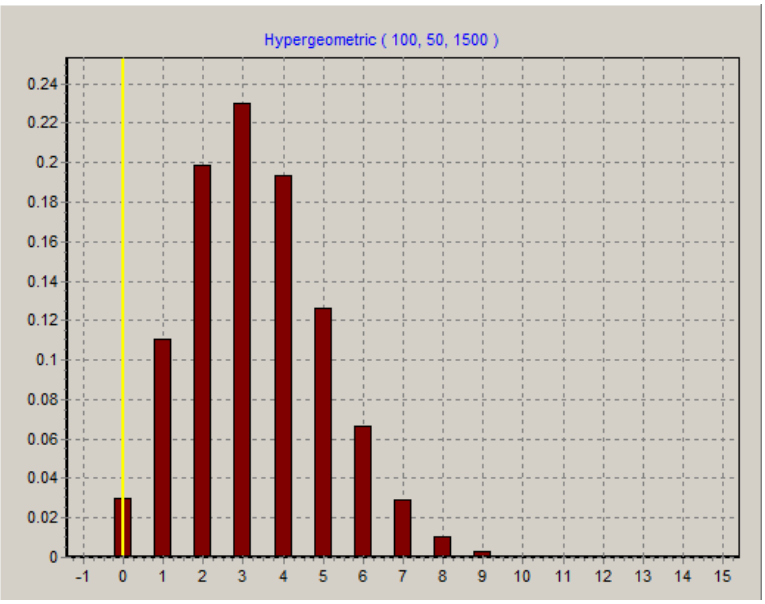


Figure A-10. A sample Hypergeometric distribution.

A11. The Inverse Gaussian distribution

An Inverse Gaussian distribution (Figure A-11) is a continuous distribution characterized by two parameters: a mean (μ) and scaling factor (λ). This distribution has been used in epidemiology to model mean infectious and latent periods. The probability density function *f(x)* is calculated as:

$$f(x) = \sqrt{\frac{\lambda}{2\pi x^3}} \exp\left(\frac{\lambda(x-u)^2}{2u^2 x}\right)$$

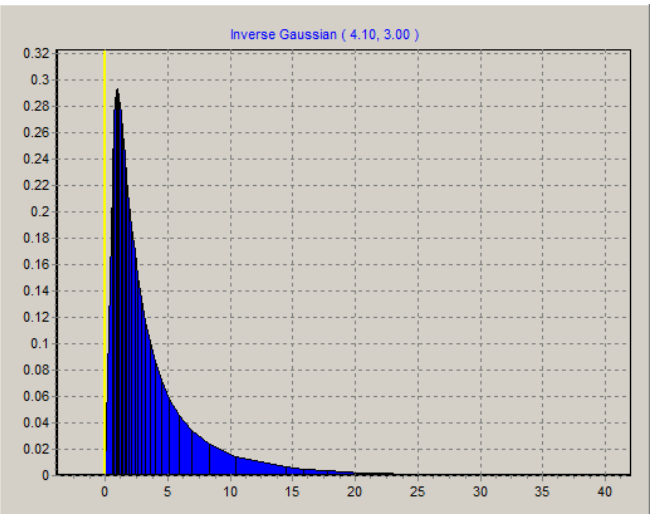


Figure A-11. A sample Inverse Gaussian distribution.

A12. The Logistic distribution

A Logistic function (Figure A-12) is a continuous distribution 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:

$$f(x) = \frac{\sec h^2\left[\frac{1}{2}\left(\frac{x-\alpha}{\beta}\right)\right]}{4\beta}$$

where *sech* is the hyperbolic secant function

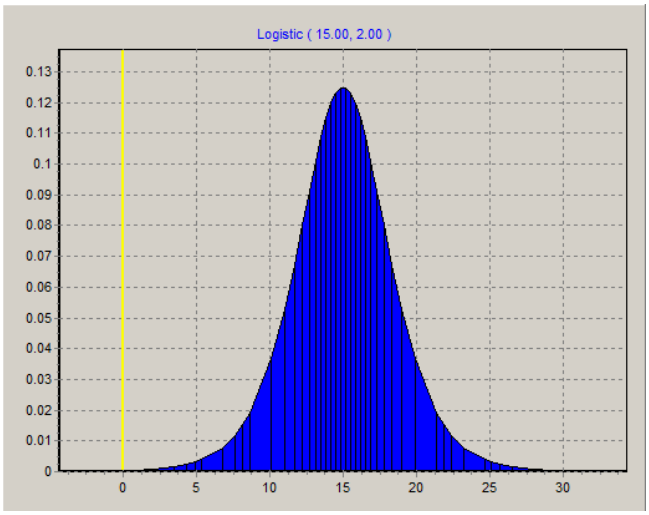


Figure A-12. A sample Logistic distribution.

A13. The Loglogistic distribution

The Loglogistic function (Figure A-13) is a continuous distribution 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:

$$f(x) = \frac{\alpha\left(\frac{x-\gamma}{\beta}\right)^{\alpha-1}}{\beta\left[1+\left(\frac{x-\gamma}{\beta}\right)^{\alpha}\right]^2}$$

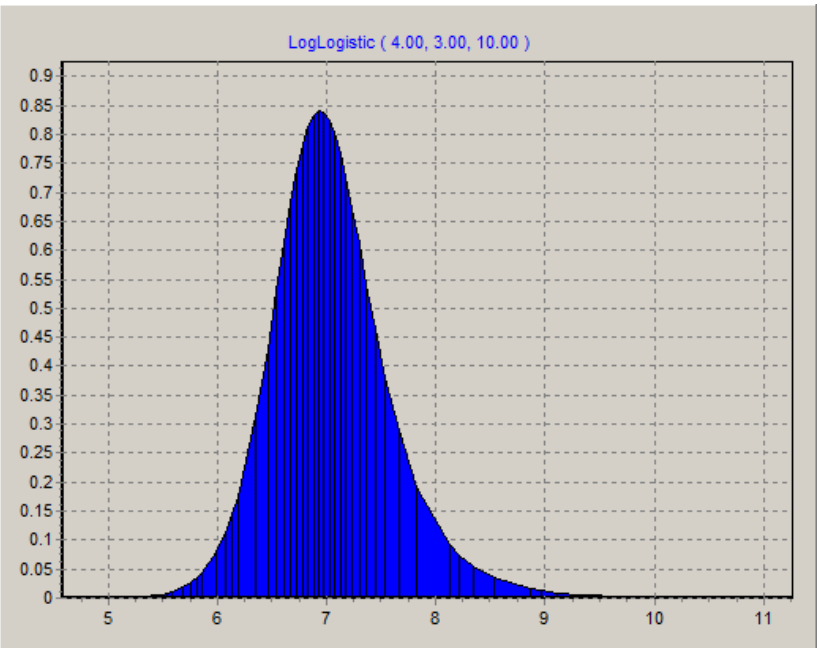


Figure A-13. A sample Loglogistic distribution.

A14. The Lognormal distribution

The Lognormal distribution (Figure A-14) 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, which is continuous, 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:

$$f(x) = \frac{1}{x\sqrt{2\pi\sigma'}} e^{-\frac{1}{2}\left(\frac{\ln(x)-\zeta}{\sigma'}\right)^2}$$

where $\zeta = \ln\left(\frac{\mu^2}{\sqrt{\sigma^2+\mu^2}}\right)$

and $\sigma' = \sqrt{\ln\left[1+\left(\frac{\sigma}{\mu}\right)^2\right]}$

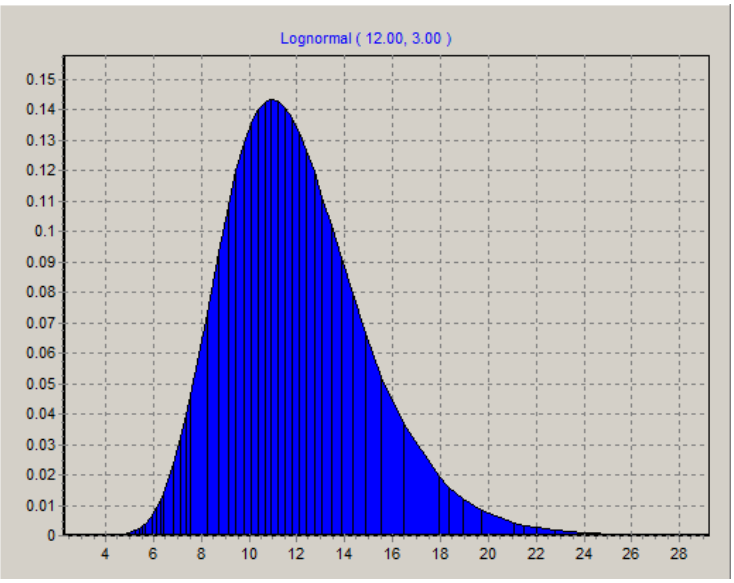


Figure A-14. A sample Lognormal distribution.

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'.

A15. The Negative Binomial distribution

A Negative Binomial distribution (Figure A-15) is a continuous distribution often used to estimate the number of failures that will occur before there are *s* successes, where there is a probability *p* of success in each trial. The parameters *s* and *p* are specified. The probability density function *f(x)* is calculated as:

$$f(x) = \binom{s+x-1}{x} p^s (1-p)^x$$

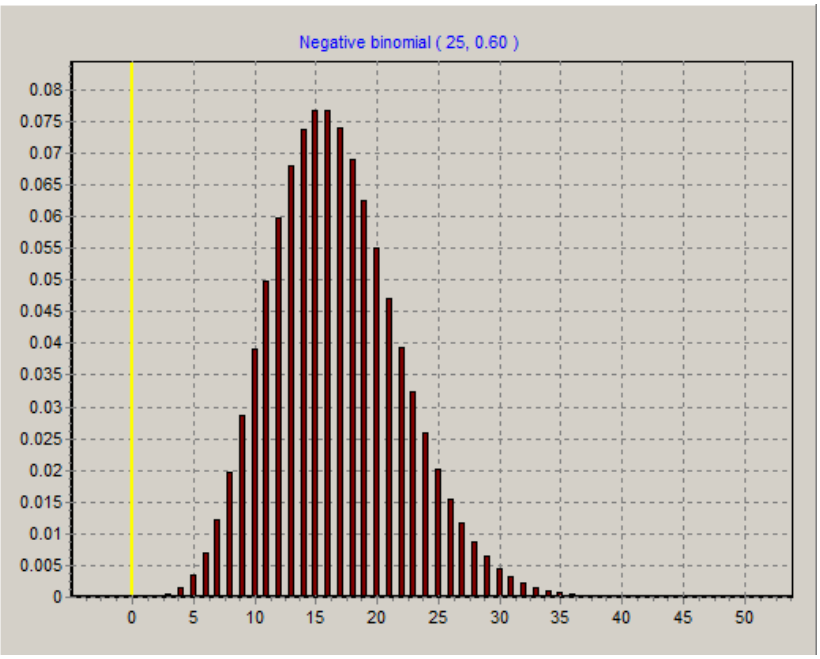


Figure A-15. A sample Negative Binomial distribution.

A16. The Pareto distribution

A Pareto distribution (Figure A-16) is a power law type probability distribution. A power-law implies that small occurrences are extremely common, whereas large instances are extremely rare. This distribution is heavily skewed to the right and has a mode and minimum that are equal. The distribution starts at the mode *a* and has a rate of decrease determined by the parameter θ. The probability density function *f(x)* is calculated as:

$$f(x) = \frac{\theta a^{\theta}}{x^{\theta+1}}$$

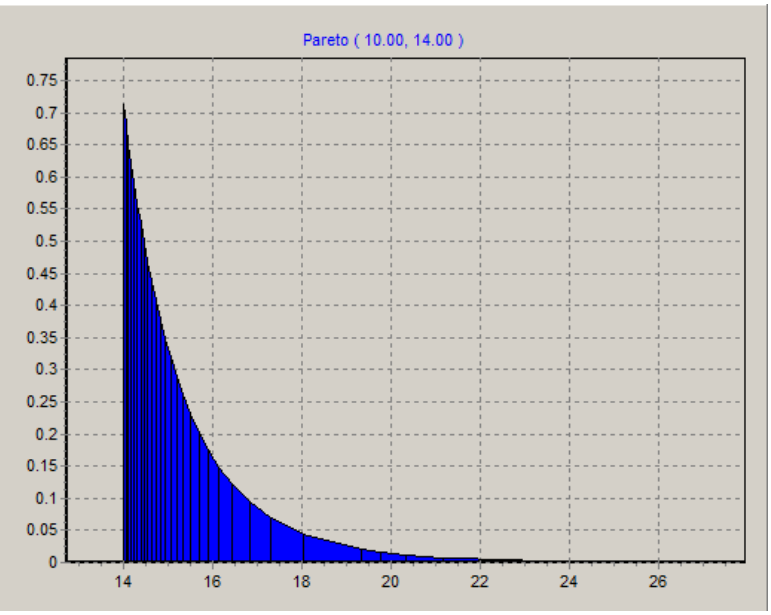


Figure A-16. A sample Pareto distribution.

A17. The Pearson 5 distribution

The Pearson 5 distribution (Figure A-17) 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 β/(α-1). The probability density function *f(x)* is calculated as:

$$f(x) = \frac{1}{\beta\Gamma(\alpha)} \bullet \frac{e^{-\beta/x}}{(x/\beta)^{\alpha+1}}$$

where Γ is the Gamma function.

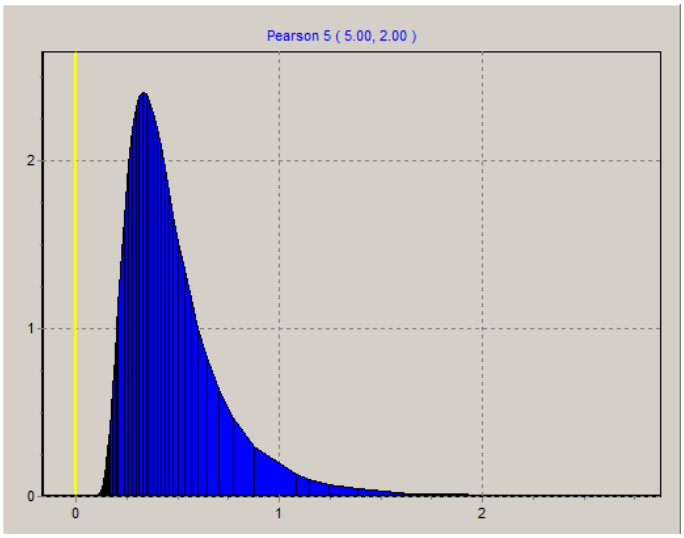


Figure A-17. A sample Pearson 5 distribution.

A18. The Piecewise (General) distribution

A Piecewise or General distribution (Figure A-18) is an empirical distribution defined by an array of points, each of which has 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.

$$f(x) = y_i + \left[\frac{x - x_i}{x_{i+1} - x_i}\right](y_{i+1} - y_i)$$

where $x_i \le x \le x_{i+1}$

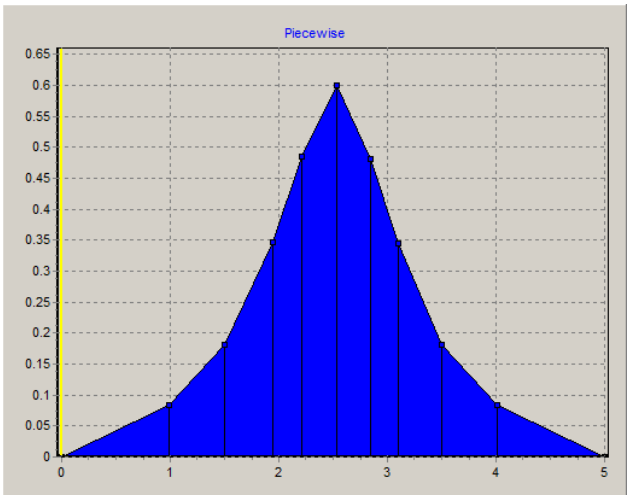


Figure A-18. A sample Piecewise distribution.

A19. The Point or fixed-value "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:

$f(x) = y$

A20. The Poisson distribution

Unlike any of the other distributions described in this appendix, the Poisson distribution (Figure A-19) is discrete rather than continuous. Poisson distributions have one specific role in a *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 9.6.3.2.

A Poisson distribution is defined by its mean, designated λ. The probability mass function *f(x)* is calculated as:

$$f(x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

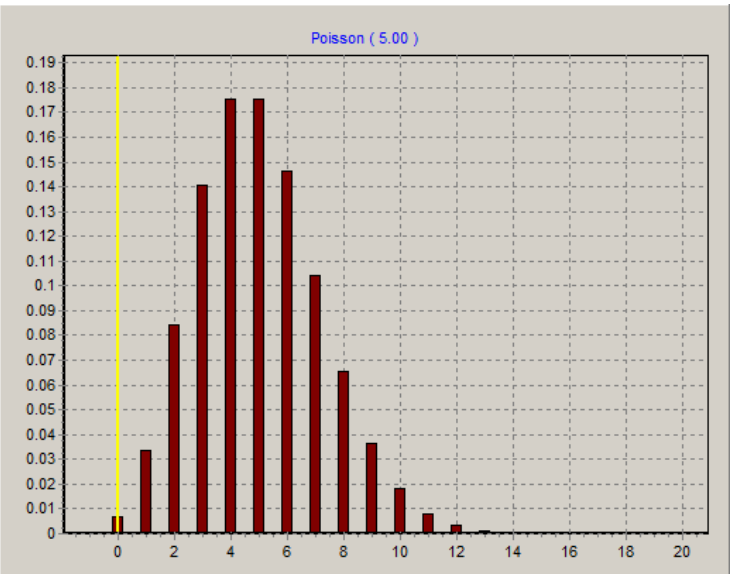


Figure A-19. A sample Poisson distribution.

A21. The Triangular distribution

A Triangular distribution (Figure A-20) 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:

$$f(x) = \frac{2(x - Min)}{(Mode - Min)(Max - Min)} \text{ if } Min \le x \le Mode$$

or

$$f(x) = \frac{2(Max - x)}{(Max - Mode)(Max - Min)} \text{ if } Mode \le x \le Max$$

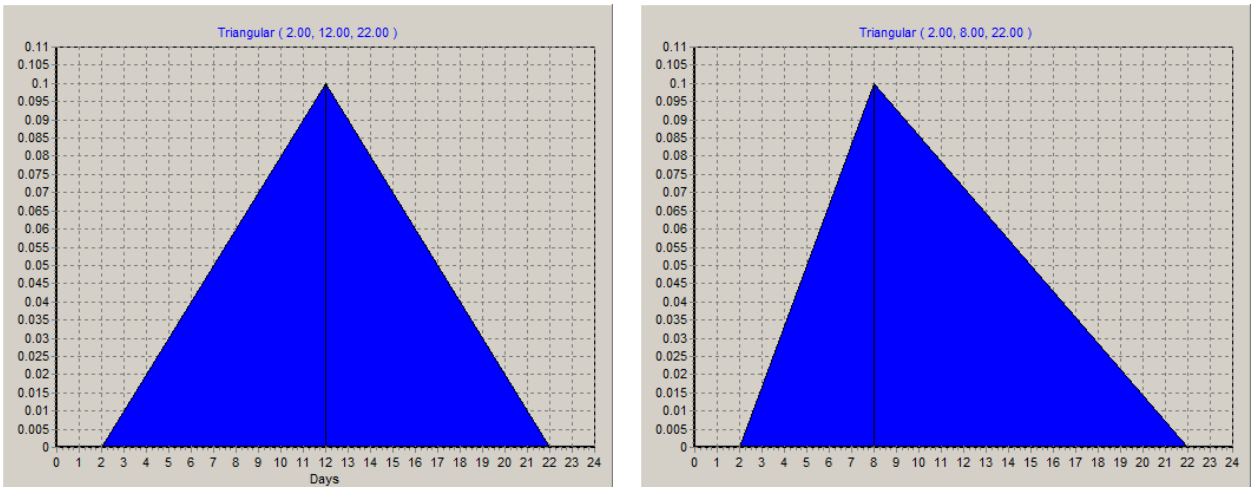


Figure A-20. Two sample Triangular distributions (symmetric and asymmetric).

A22. The Uniform distribution

The Uniform distribution (Figure A-21) 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:

$$f(x) = 1/(Max\text{-}Min)$$

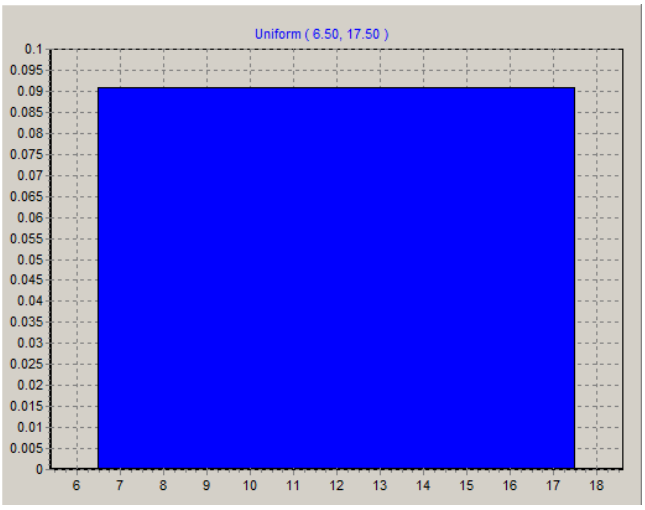


Figure A-21. A sample Uniform distribution.

Section A5 has additional information about the use of the Uniform distribution in *NAADSM*.

A23. The Weibull distribution

A Weibull distribution (Figure A-22) 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:

$$f(x) = \frac{\alpha x^{\alpha-1}}{\beta^{\alpha}} e^{-(x/\beta)^{\alpha}}$$

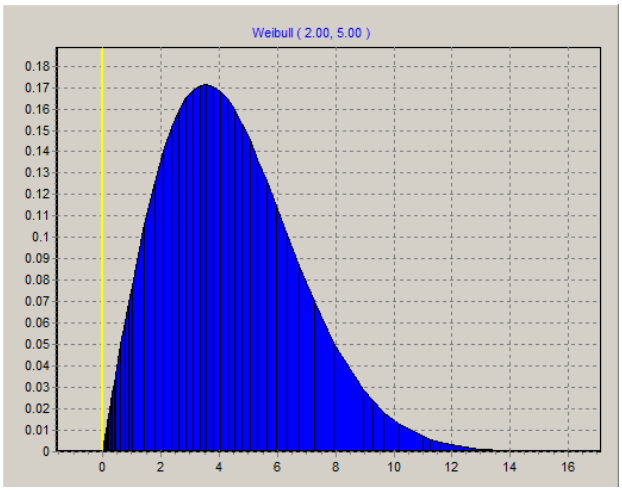


Figure A-22. A sample Weibull distribution.

A24. Useful references for *pdfs*

In addition to references cited in Chapter 13, the following resources provide additional detailed information about the calculation and application of *pdfs*:

- Galassi, M., Davies, J., Theiler, J., Gough, B., Jungman, G., Booth, M., and Rossi, F. 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, M.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*>.

- Van Hauwermeiren, M. and Vose, D. 2009. *A Compendium of Distributions*. [ebook]. Vose Software, Ghent, Belgium. The *Compendium* is available for download as an Adobe PDF file from <*http://www.vosesoftware.com/content/ebook.pdf*>.

- Vose, D.. 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 B. Plain text comma-delimited (`*.csv`) files used in *NAADSM*

B1. Files containing population (herd or unit) data

B1.1. 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 B-1 (field names are not case-sensitive and do not contain spaces).

Table B-1. Required fields for *NAADSM* `*.csv` files.

Field name	***Field description***
UnitID, HerdID or ID[1]	Unique integer identifier for each unit. ID must be greater than 0.
ProductionType	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.
UnitSize or HerdSize	Integer indicating the number of animals in the unit.
Lat	Real (floating point) number indicating the latitude of the unit. Values must be between -90 and 90, inclusive.
Lon	Real (floating point) number indicating the longitude of the unit. Values must be between -180 and 180, inclusive.
Status	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 B-2.

[1] *UnitID* is the preferred name for this field as of version 4.0.0, but CSV files containing the field name *HerdID* or *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.

Table B-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
Dead from disease	X	7
Vaccine immune	V	5
Destroyed	D	6

Comma-delimited population files may contain two optional fields, *DaysLeftInState* and *DaysInDiseaseState*. These optional fields are used primarily for application testing. Under most circumstances, users may ignore them.

The optional field *DaysLeftInState* contains an integer value for the number of days the unit has remaining in its current disease state. 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). *DaysLeftInState* applies to units in all disease states except Susceptible: if *DaysLeftInState* is specified for a susceptible unit, the value is ignored. The second optional field, *DaysInDiseaseState*, contains an integer value for the number of days that a unit has been in its current disease state. As above, a value of -1 may be used to indicate that this number is unspecified.

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.

B1.2. Population files exported by *NAADSM*

Comma-delimited population (operational unit) files generated by *NAADSM* have the eight fields described above: *UnitID*, *ProductionType*, *UnitSize*, *Lat*, *Lon*, *Status*, and *DaysLeftInState*. 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.

XML population files generated by *NAADSM* have six fields described above:*UnitID*, *ProductionType*, *UnitSize*, *Lat*, *Lon*, and *Status*. An example from a portion (two units) of a population XML file is shown in Figure B-4.

See Section 9.4.3 for details on how to export a population file from or import a herd file into *NAADSM*.

B1.3. Sample population files

Figure B-1 through Figure B-4 show sample ***.csv** and *.xml files generated by or suitable for import into *NAADSM*.

```
UnitID,ProductionType,UnitSize,Lat,Lon,Status,DaysLeftInState
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 B-1. Default ***.csv** file format generated by *NAADSM* 4.0.0 (and higher).

```
UnitID,ProductionType,UnitSize,Lat,Lon,Status,DaysLeftInState
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 B-2. A ***.csv** file generated by *NAADSM* 4.0.0 (and higher) using production type ID number.

```
UnitID,ProductionType,UnitSize,Lat,Lon,Status,DaysLeftInState
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 B-3. A ***.csv** file generated by *NAADSM* 4.0.0 (and higher) using numeric transition state code.

```
<?xml version="1.0" encoding="UTF-16" ?>
<herds
  xmlns:naadsm="http://www.naadsm.org/schema"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <herd>
    <id>3961</id>
    <production-type>Cattle</production-type>
    <size>100</size>
    <location>
      <latitude>39.44972</latitude>
      <longitude>-90</longitude>
    </location>
    <status>Susceptible</status>
  </herd>

  <herd>
    <id>3962</id>
    <production-type>Cattle</production-type>
    <size>100</size>
    <location>
      <latitude>39.22477</latitude>
      <longitude>-90</longitude>
    </location>
    <status>Susceptible</status>
  </herd>

</herds>
```

Figure B-4. Default **`*.xml`** unit file generated by *NAADSM* 3.1.23 (and higher).

Figure B-5 shows a **`*.csv`** file generated by the USDA disease modeling application *SpreadModel* (Schoenbaum and Disney, 2003) 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
2,1,70,54.766201,-2.99002695,0,50,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3,1,8,54.8438072,-3.0588901,1,5,4,1,0,3,0,0,0,0,0,0,0,0,0,0
4,1,22,54.7988701,-3.05926991,3,7,20,1,0,3,0,0,0,0,0,0,0,0,0,0
```

Figure B-5. 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, due to page width constraints, the single header row in this particular example is displayed across four rows.

B2. Files containing piecewise probability density functions

Point arrays for piecewise *pdf*s may be imported from **`*.csv`** files (see Section 7.1.5.3.2). 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 B-6 shows sample data suitable for import into *NAADSM*. Point arrays for piecewise *pdf*s are exported (Section 7.1.5.3.3) 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 B-6. 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.)

B3. Files containing relational functions

Point arrays for relational functions may be imported from **`*.csv`** files (see Section 7.1.6.4). 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 B-7 shows sample data suitable for import into *NAADSM*. Point arrays for *rel*s are exported in the same format (Section 7.1.6.5).

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

Figure B-7. 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 C. *NAADSM/PC* and *NAADSM/SC*

C1. 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.

C2. 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 the *NAADSM* website 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://www.naadsm.org/source/source-sc*>.

C3. 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 8.3).

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.

C4. 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 <*http://www.naadsm.org/contacts*>) 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*.