METAMODEL
MANAGER
users manual • version 1
N
W
E
S
Chicago Zoological Society
Inspiring Conservation Leadership
250
200
150
100
50
0
N
0
10
20
30
40
50
Year

MetaModel Manager User's Manual

Software written by JP Pollak and Robert C. Lacy

Manual written by Becky E. Raboy and Robert C. Lacy

Support for the development of the MetaModel Manager software and this manual was provided by the Chicago Zoological Society and a generous grant from Donna Myers and Aaron Leibovitz. Thanks to the many colleagues of Metamodeling Network team and the IUCN/SSC Conservation Breeding Specialist Group who helped in the conceptualization and development of the program. A special thanks to Phil Miller, Philip Nyhus, and Sara Zeigler in addition to the developers and author of this manual for hours of brainstorming, head-scratching, and testing of the program.

Cover design by Stephanie Raboy

Bacteria photo credit (upper right photo on cover): Photo modified from Flickr contributor kaibara87 (http://www.flickr.com/photos/kaibara/2234750993/) according to the creative commons.

Wolf tracks photo credit (lower left photo on cover): NPS Photo – wolf images #16190.

Chicago Zoological Society
Inspiring Conservation Leadership

Citation of the software:
Pollak JP and Lacy, R.C. 2013. *MetaModel Manager. Version 1.0*. Brookfield, IL: Chicago Zoological Society.

Citation of this manual:
Raboy, B.E. and Lacy, R.C.2013. *Metamodel Manager. User's Manual*. Brookfield, IL: Chicago Zoological Society.

Contents

Statement of Use………………………………………………………………………… 1

Chapter 1. Introduction……………………………………………………………..…… 2
What is the MetaModel Manager?.. 2
Philosophy ……………………………………………………………………...…. 2
Implementation……………………………………….…………………………… 4

Chapter 2. Getting Started with MMM…………………………………………………… X

Chapter 3. Tutorial………………………………………………………… X

Chapter 4. Template for Open Data Exchange……………………… X

Chapter 5. Understanding State Variables……………………………………………. X

Appendix I. Literature Cited…………………………………………………………….. X
X
Appendix II. Nyhus et al., 2007………..………………………………………………..
X
Appendix III. Evaluator……...…………………………………………………………..

Appendix IV. MMMacro……...……………………………………………………..

Statement of Use

On behalf of the team of collaborating scientists developing this software, we wish to convey an important cautionary note: This manual does not explain to users how to set up scenarios in the individual modules (herein referred to as linked-programs) brought together by the MetaModel Manager interface except where the implementation of the MetaModel Manager requires certain additional formatting of the linked-program input. We presume you will have previous knowledge of these programs (such as Outbreak or Vortex) or will consult the appropriate manuals and available resources. Our principal goal by writing this manual is to explain the metamodeling approach and how the MetaModel Manager program works.

This is a first draft of the manual, published so that users can get started using MetaModel Manager. A more detailed manual, with further explanations and sample analyses, will be produced in the near future.

For now, please send bug reports, suggestions and any requests for additional information to help@vortex10.org. We cannot guarantee that all comments and requests will be answered, but responses will be provided when possible. We do, however, value all feedback!

- Biocomplexity Network, January 2013

Chapter 1. Introduction

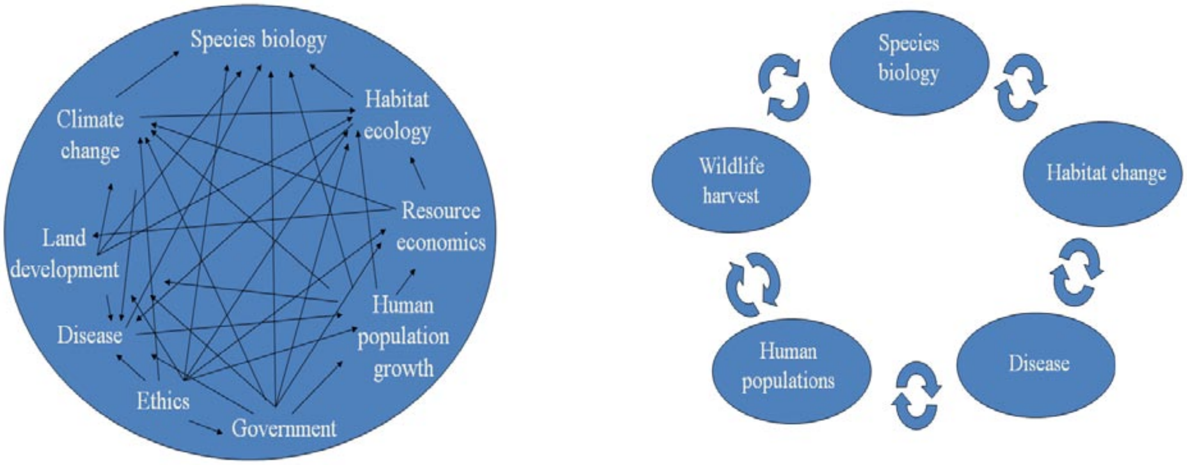
1.1 What is the MetaModel Manager?

The MetaModel Manager is a new conservation biology software program designed to link population demographics of multiple interacting species with other processes such as emerging diseases, habitat and climate change, landscape dynamics, and/or animal movement patterns. MetaModel Manager integrates computer simulation models such as Vortex, Outbreak and Spatial, in addition to other user-created models—to understand the influence of environmental, demographic, and genetic factors on the probability of extinction of threatened species. Like the programs its works with, MetaModel Manager's function is to serve as a predictive modeling tool used to determine likely outcomes for a given scenario or to compare outcomes across multiple scenarios. The program runs iteratively and its results are generated in the form of means or probabilities, within the linked-program platforms. Analysis and interpretation of results can be used in the establishment of conservation goals and to direct or prioritize conservation action. MetaModel Manager can be used in an interdisciplinary workshop setting, or by individual users. The software was recently released as version 1.0, and further enhancements are planned for the near future.

1.2 Philosophy

Wildlife conservation problems are complex and must be looked at through the perspectives of many different disciplines. MetaModel Manager offers a way to achieve transdisciplinary modeling with general facility and creates a central convening point for people to come together to make critical conservation decisions using a wider variety of information.

The principle behind the development of MetaModel Manager was to create a robust interface that calls in existing programs to work synergistically. In so doing, the user can tap into the unique capabilities of each program without the need to create new potentially cumbersome "mega-" models (Figure 1). The program provides an interface for the transfer of information across individually based models that carry biological information attributable to specific indexed individuals in a population. The information that is sent back and forth between programs is considered "open-data," and can be modified by any of the linked-programs operating within the MetaModel Manager framework.



MEGAMODEL **METAMODEL**

Figure 1. Conceptual comparison of a megamodel versus a metamodel of interacting factors influencing species biology.

The roots of the MetaModel Manager software originated from discussions amongst an informal grouping of people known as the "Biodiversity Research Network" created to build "interdisciplinary connections and stimulate an exchange of expertise among specialists concerned with the conservation of biodiversity" (Westley, 2003). An initiative was identified by the Network to develop "tools and processes for securing the involvement, collaboration, and responsibility of a wider range of local stakeholders in conserving species in their habitats and the ecosystem management required to achieve this" (Westley). The idea of enhancing interdisciplinary information exchange gave rise to a specific focus on developing new software to attain this goal, and the Network transitioned from the Biodiversity Research Network to the "Biocomplexity Network" made up of many of the same participants and new members as well. The Biocomplexity Network targeted the creation of modeling and process tools to facilitate the study of biocomplexity by 1) updating existing conservation biology software (Vortex, Outbreak), 2) creating new software (Spatial, Vaccinator), 3) identifying other complementary software (HexSim, RAMAS, the ARC-GIS platform), 4) identifying potential new programs for development (Sim-Social), 5) and by creating the MetaModel Manager to serve as an interface between these and other user-created programs.

Metamodel Manager is a significant step towards developing truly integrated solutions for species conservation by working across disciplines and by considering the larger picture of interacting and often spatially and temporally explicit phenomena that affect a species outcome. The issue of how accurate such complex modeling can be remains a concern. Program users and those interpreting results may expect that as model complexity increases, the overall predictive accuracy should increase. However the opposite could also occur when complexity is added but not conveyed or modeled properly. Sensitivity testing can indicate which variables are most critical to be modeled accurately and can be carried out within linked-programs or at higher levels (by adding and subtracting linked-programs). Arguably, the success of the MetaModel Manager program for conservation biology is not in the accuracy results that are generated per se, but in how well the results are interpreted when keeping in mind its aforementioned limitations.

1.3 Implementation

System Requirements

MetaModel Manager was developed in the C#.NET programming language, for use on computers running MS Windows. We believe that it will run properly under Windows7 and all prior versions of Windows, but we cannot guarantee that it will work under all implementations of Windows. It can be run on Apple Mac computers if they are running implementations or emulations of Windows.

MMM does not itself require much computer memory or processor speed to run. However, often the models linked with MMM will work best (or only) if the computer has ample RAM and a fast processor.

Installing MMM

MetaModel Manager is available for free download online at www.vortex10.org. The installation for MMM (MMMInstallation.msi) automatically configures necessary folders and includes the latest Outbreak, Spatial, PathHistory, and the test version (without yet a full GUI) of Vortex10.

Some points/advice on folder locations:

- It is easiest, and probably best, to install MMM in the default Program Files folder on the computer.
- It is easiest, and probably best, to create a separate folder for holding your MMM projects. The installation automatically creates a C:\MetaModelManager folder and a C:\MetaModelManager\Projects subfolder for you.
- We recommend keeping Outbreak, Spatial, Vortex and other projects in a folder(s) other than the program file folder where the program is installed. The installation creates C:\SpatialProjects and C:\OutbreakProjects folders for you.

The installation automatically creates program short-cuts and puts them on your desktop.

Overview of program's mechanics

The combination of making and choosing appropriate specification files that are all linked by unique identifiers with correct process order and time steps and structured with biologically meaningful inter-program linkages enables MMM to run as intended to link multiple programs pseudo-synchronously (see discussion below on process order), passing back and forth and modifying shared information. The following sections in this chapter further explain some program technicalities.

Program Types

There are three types of linked-programs that MMM can incorporate:

- System level programs– these initialize or define the individuals in the starting population/s. The system level program will often handle population demographics, although this is not a requisite. For multiple species models, a system level program can be run for each species within a metamodel.

- Modifier programs – these may modify characteristics of the individuals in the population/s.

- Utility programs to provide data transformations – these programs provide added functionality to the metamodel, but are not necessarily an attribute of the population being modeled or derived from individually based models.

Specification files

As a first step, the user must create specification files in each linked-program (see table below for existing programs adapted for MetaModel Manager). The MetaModel Manager will not create these files for you, it is something that must be done independent of the MetaModel Manager, within the interfaces provided by the linked programs (see the introductory user statement at the start of the manual).

Process order

The programs executed via the MMM interface do not technically run synchronously – rather they are executed sequentially according to user-defined order. Switching the order in which the programs run may affect the final metamodel outcome and therefore it is important to determine the best ordering for your specific model.

Time steps

Time steps can vary among programs. A cycle in one linked program may run once a year, whereas it runs daily for another (thus occurring 365 times for each time the other runs). Any amount of time steps can be input by the user.

Inter-program prep work

One of the advantages of using MMM is that the user may model particular components of one program that are known or thought to influence the outcome of another system component. For example, the disease state of an individual obtained from an epidemiological modeling program may influence the likelihood of reproducing in a Population Viability Analysis (PVA) model. Preparatory work to facilitate inter-program information exchange is critical for the linkage between any two programs to carry biological meaning. The coding for the linkage can be done directly within the independent specification files, through a data translator (Evaluator) that is provided within MMM, and/or by using the utility program MMMacro (installed with MMM) that allows more complex, multi-step macros or scripts for manipulating data. In many cases (especially so for Vortex), state variables will need to be set up. Evaluator and MMMacro provide the user platforms to manipulate variables passing through MMM to/from other programs. Evaluator provides the user with a way to write one-line transformations of variables describing individuals or the population. MMMacro provides a simple way for a user to write, "on the fly", short BASIC-like programs (often sequences of transformations, each handled by Evaluator) to process the shared variables in MMM. Using MMMacro is not required to establish linkage in cases where specific linkage functions are instead created within the original specification files.

Existing Program Suite Pre-Adapted for compatibility with MMM

Program	*Type*	*Developer*	*Web Address for download*
Vortex	System level	R. Lacy and J.P. Pollak	http://www.vortex10.org/Vortex10.html
Outbreak	System level or modifier	J.P. Pollak, R. Lacy, and others	http://www.vortex10.org/Outbreak.html
Spatial	Modifier	J.P. Pollak	http://www.vortex10.org/Spatial.html
MMMacro	Utility	R. Lacy	Included in the MMM installation
Path History	Utility	J.P. Pollak	Included in the MMM installation

See Chapter 4 to learn how to adapt user-created models to run within MMM.

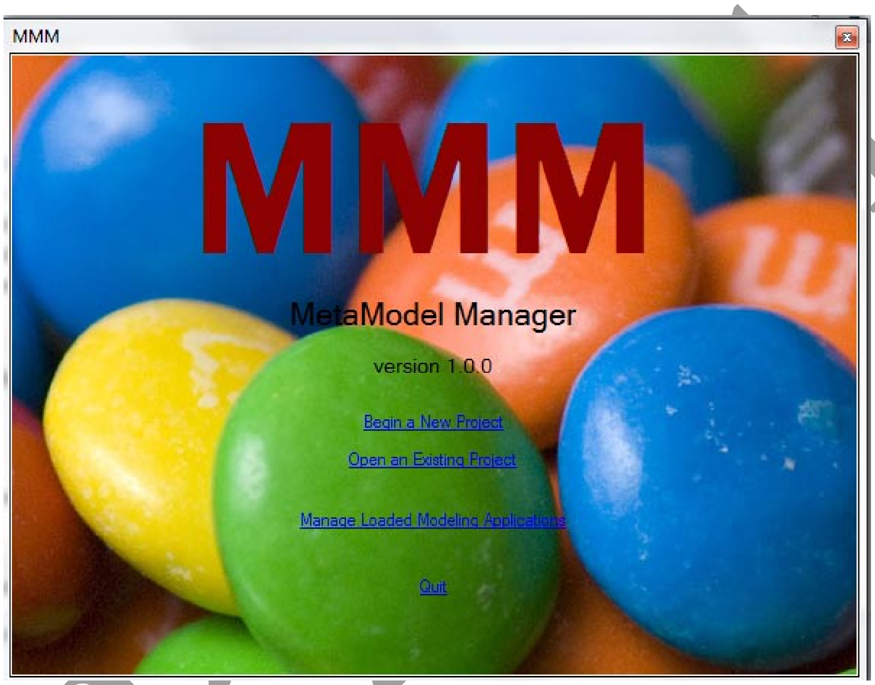
Processing speed

Depending on your input programs, MetaModel Manager may run slowly. We suggest that you run test models first with less iteration, over less time, and with fewer metamodel complexities, before attempting longer and more intricate runs. For example you may limit the number of iterations, time steps, cycles, or even the numbers of programs interacting at first to make sure you are satisfied with your model construction.

Chapter 2. Getting Started with MMM

This chapter explains the choices available on each screen of the MetaModel Manager. An examples showing some of the functionality of the MMM program will be presented in Chapter 3. The choices that a user has per screen are indicated herein in bold lettering, with descriptions following.

MMM Opening screen:

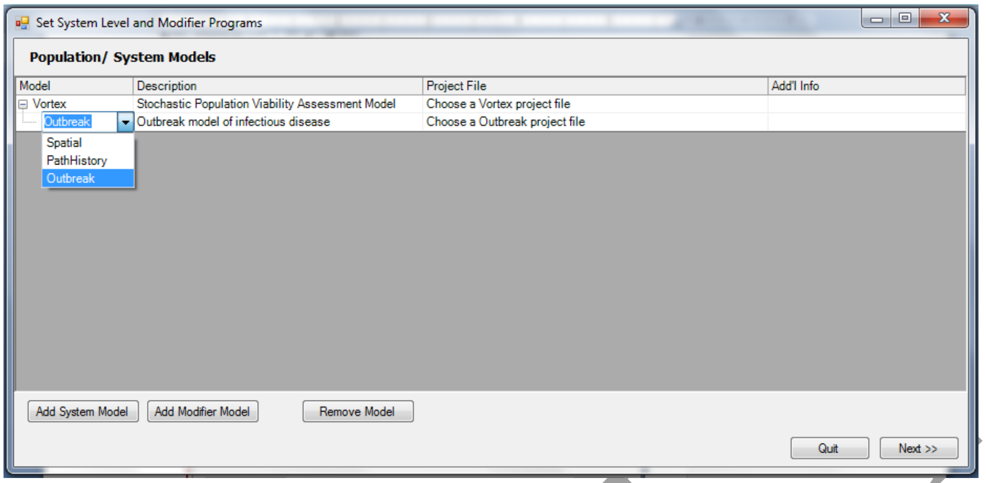


Begin a New Project- choose this option if you are setting up a project for the first time
Open an Existing Project- choose this option if you are returning to a project that you have been working with previously
Manage Loaded Modeling Applications- this page will allow you to add additional applications that are not part of the pre-adapted MMM program suite. Options include:

- **Add-**
- **Add from XML-**
- **Delete-**
- **Modify-**

Creating and managing loaded applications will usually be a process for prorgamn developers, not users of existing programs. See chapter 4 for more details on using additional applications
Quit- select to leave the program.

Setting System Level and Modifier Programs screen:



Set System Level and Modifier Programs

Population/ System Models

Model	Description	Project File	Add'l Info
Vortex	Stochastic Population Viability Assessment Model	Choose a Vortex project file	
Outbreak	Outbreak model of infectious disease	Choose a Outbreak project file	

If you choose to begin a new project you will have to load the individual system and modifier files for your model. You are first prompted to choose a model. Select your system level linked-program in the drop down bar. Then choose your specification file from the browser box that opens. If a Vortex vpj file has multiple scenarios, you will need to choose which scenario from a dropdown list. Next you can add a modifier model to that system level using the Add Modifier Model button.

Add Modifier Model: Selecting this button adds a modifier program. Choose the modifier program from the drop down menu for each system level model. In some cases you may wish to add more than one additional linked modifier program. You may do this by selecting the Add Modifier Model more than once, each time choosing a different program. It is not necessary to choose any modifier program for a system level model to run.

Add System Model: Additional system level models may be selected by pressing this button. For example, the user may run two Vortex models at the same time, one characterizing a predator, and the other characterizing its prey.

Remove Model: Will allow you to delete a model from the overall Metamodel being developed

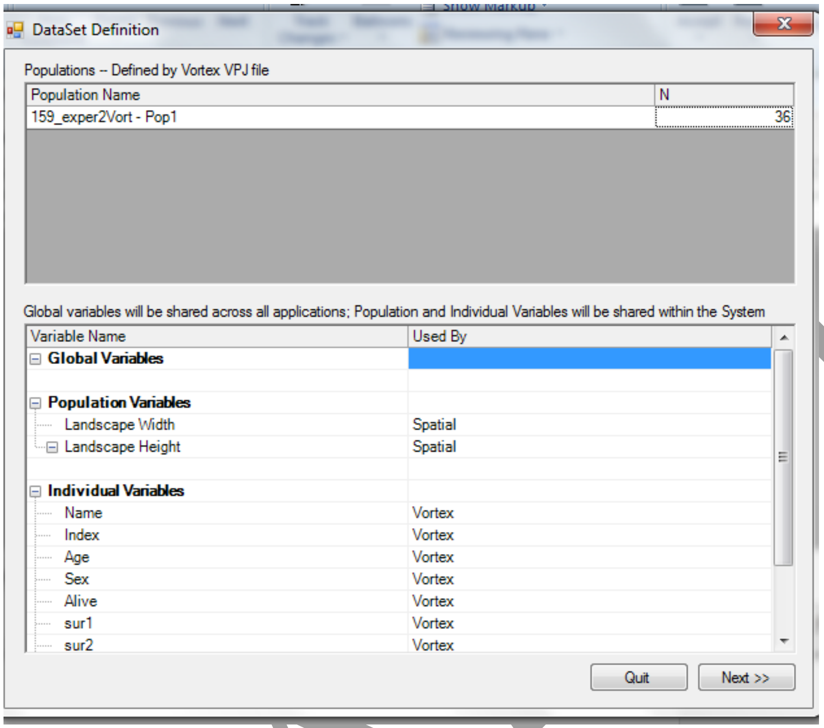
Note that Outbreak can be run as either a system model (doing its own demography) or as a modifier model (with Vortex or SimSimba or something as the system model).

If the specification files are not set, you will see the message reminding you to do so before you can advance to the next page.

When user has completed this page, select **"Next>>"**

NOTE: if you choose “**Open an Existing Project**” at the start up screen, MMM will import the specification files which the user had previously uploaded and saved to a mmmm file. You will automatically be advanced the Dataset Definition screen.

Dataset Definition screen:



Populations -- Defined by Vortex VPJ file

Population Name	N
159_exper2Vort - Pop1	36

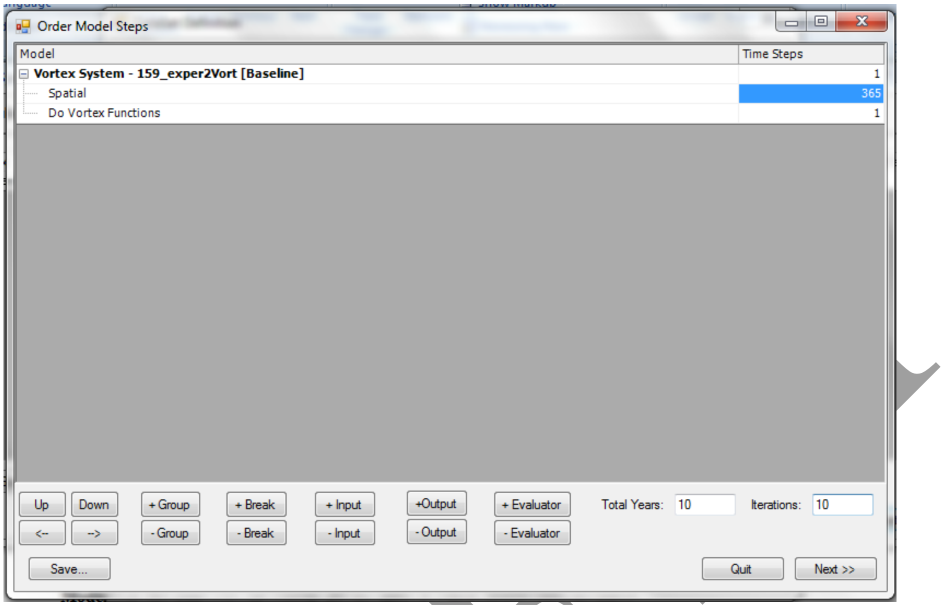
Global variables will be shared across all applications; Population and Individual Variables will be shared within the System

Variable Name	Used By
Global Variables	
Population Variables	
Landscape Width	Spatial
Landscape Height	Spatial
Individual Variables	
Name	Vortex
Index	Vortex
Age	Vortex
Sex	Vortex
Alive	Vortex
sur1	Vortex
sur2	Vortex

Populations – Defined by System project - This lists a summary of the population(s) from the system level program specification files, and can be useful to help confirm you have selected the appropriate files. If the system level program is Vortex, the .vpj title appears first, followed by the vpj scenario name in brackets. Population sizes are imported from the system level model specification files.

Variables – You will see a list of global, population, and individual variables that may or may not be shared between linked-programs. The ‘Used By’ column indicates which program(s) use this information. This table shows which of the linked programs had told MMM within their specification files that they would be using (creating, modifying, or just reading) each shared variable. However, any linked program can access any of the variables made available by other programs and therefore included on this list.

Order Model Steps screen:



Model	Time Steps
Vortex System - 159_exper2Vort [Baseline]	1
Spatial	365
Do Vortex Functions	1

Total Years: 10 Iterations: 10

Model - on this page you can control for the order in which MMM runs the linked programs, and also specify the number of time steps each program will run before data are passed on to the next linked program. The user must decide the best order to run the linked-programs, and whether or not it makes sense to run programs on the same time-scale. Often times one program is better suited to assessing daily patterns (i.e. disease spread, which in Outbreak runs on a daily step without option to change), whereas another might be designed to assess patters over a season or a year (i.e. demographics).

The system level programs will be listed in bold, with their modifier programs listed underneath them.

Buttons on the bottom of the page control for the overall design of linked-programs, including ordering, grouping, data overrides, duration and number of iterations:

Up/Down- Buttons allow the user to move linked-programs up or down and help the user control the overall order of the metamodel.

+Group/-Group- Group is used for managing the time steps. With this function, it is possible to group two linked programs to run between themselves for a specified amount of time steps, before they pass information back to a third program. This is often necessary for inter-functionality of metamodels that run more than two linked programs. For example, Spatial and Outbreak could be run 365 times each for every time Vortex is run. Without such grouping, the model would look like this:

Vortex		1
	Outbreak	365
	Spatial	365

Meaning, that you would run 100 V years, for each of which you would run 365 daily time steps of Outbreak and Spatial. But, in this case, Outbreak and Spatial wouldn't really be working together, but rather Outbreak would run for its 365 days of the year and then Spatial would simulate its 365 days. To get Outbreak and Spatial to alternate (one day of Outbreak then one day of Spatial, etc.) they would need to be grouped together so that they as they each run once, they pass data to the other, then run again. Therefore the model should be set up as:

Vortex			1
	Group		365
		Outbreak	1
		Spatial	1

Here, within each of the Vortex time steps there would be 365 daily time steps of the pair of programs – Outbreak and Spatial.

+Break/-Break - The user may wish to impose breaks during the simulation process in order to see, change and/or save the interim results. The breaks can occur by years or multiple of years. The breaks can also occur according to rules regarding a global variable. You may use the up and down arrows to specify at what stage in the MMM process the break should occur. Once the program has paused, the user must select the **OK** button for the simulations to resume. The breaks always begin at year 0. By selecting **Show Population**, a table will be generated that displays the entire population with corresponding traits (i.e. Name, Index, Age, Sex, Alive for Vortex and the Disease state, StatePermanant, DaysInState, and IsVaccinated for Outbreak). The user can change any value in the table. Reset changes will turn the values back into those which had previously been generated in the model. **Write to a File** makes a text file containing the information from the table.

+Input/-Input - The input function allows the user to import values to override those generated in the modeling process. This might be relevant in a situation where the user wants more control over the impact of temporally occurring events on population characteristics.

The text file must be set up according to the following rules:

1. Any rows in the file that begin with "#" are ignored, so if you want file info, a header, etc., start it with #.
2. Each row of the file represents a new year.
3. Within each row, separate the variables that you are changing (order matters, specified in the MMM interface) by a semi-colon.
4. If the variable is an array, separate the elements of the array with a comma.

The following are examples of simple input files. The first represents a file setting two variables for 5 years:

```
#First Example, 2 variables for 5yrs
0.1; 0.9
0.2; 0.8
0.3; 0.7
0.4; 0.6
0.5; 0.5
```

The second, an array with 4 elements for 5 years:

```
#Second example, 1 array 4 elements for 5yrs
0.1, 0.2, 0.3, 0.4
0.2, 0.3, 0.4, 0.5
0.3, 0.4, 0.5, 0.6
0.4, 0.5, 0.6, 0.7
0.5, 0.6, 0.7, 0.8
```

And finally, a regular variable, then an array of 4 elements:

```
#Third example, 1 normal variable, 1 array 4 elements 5yrs
0.9; 0.1, 0.2, 0.3, 0.4
0.8; 0.2, 0.3, 0.4, 0.5
0.7; 0.3, 0.4, 0.5, 0.6
0.6; 0.4, 0.5, 0.6, 0.7
0.5; 0.5, 0.6, 0.7, 0.8
```

The Input Step Setup will allow user to choose the text file. Below that, the user must specify which variables will be input. If more than one variable is being input, order matters. The order must follow the order that is indicated in the text file that was created. Thus the up and down buttons can be used to regulate order.

+/- Output- This option lets you output the population data as stored in MMM, rather than any data output specific to a component model. This will create a dump of all data on the population and individuals each time step – which can be a very useful way to see just what is happening in your simulation.

+/- Evaluator -- *Evaluator* was developed as a utility to provide to MMM (and other software) the ability to evaluate user-defined functions, to which are passed the sets of parameter values. *Macro* is an extension of *Evaluator*, allowing evaluation of short programs ("methods" or macros) that consist of assignments of variables using *Evaluator* functions, within simple (Basic language-like) program flow. Both *Evaluator* and *MMMacro* can be used to transform state variables that describe the system, populations, or individuals. See www.vortex10.org/Evaluator.doc for more information.

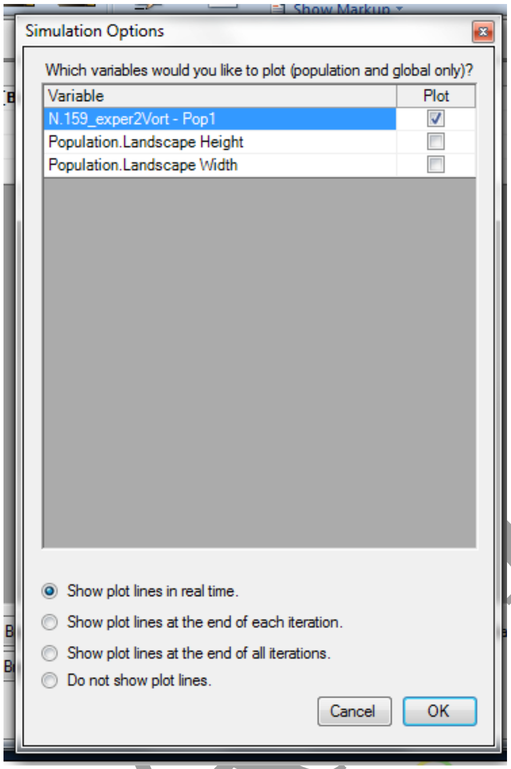
Total Years - is the length of time (number of cycles) you want full sequence of models to run for. The default is 10 years.

Iterations- is how many times you want MMM to generate a complete round of simulation modeling. The default is one iteration.

When you select **Next**, you will be given an option to save the MMM project. It is not necessary to save your project in order to continue with the simulation, but may be useful to you for future purposes.

Simulate Options screen:

After selecting **Next**, a screen comes up that looks like this:



Simulation Options

Which variables would you like to plot (population and global only)?

Variable	Plot
N.159_exper2Vort - Pop1	☑
Population.Landscape Height	□
Population.Landscape Width	□

● Show plot lines in real time.

○ Show plot lines at the end of each iteration.

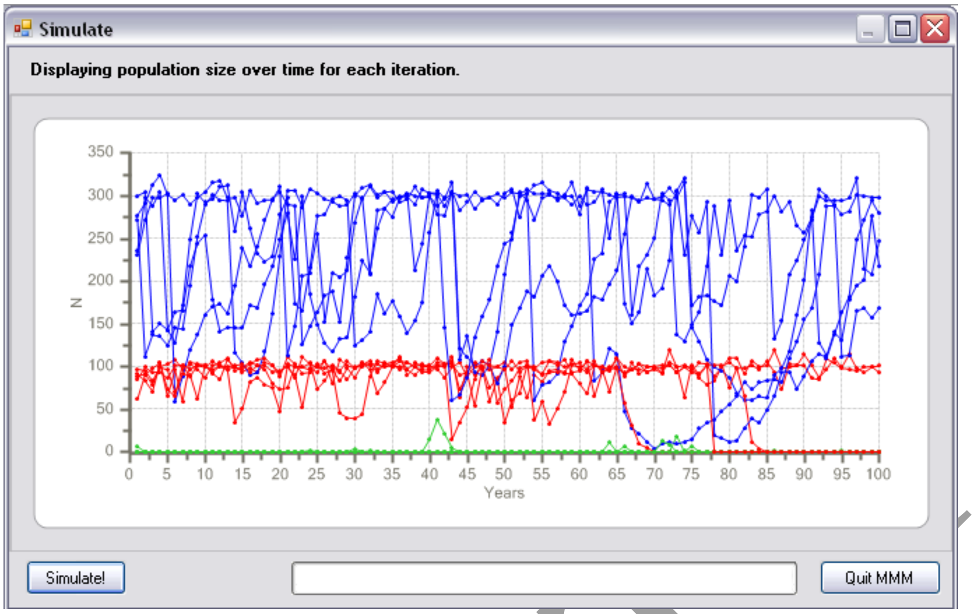
○ Show plot lines at the end of all iterations.

○ Do not show plot lines.

The default on this page is to see N for each population, but you can turn off the display of N (because sometimes the scale needed to show N makes it impossible to see variables that are on, say, a 0 to 1 scale), and you can add any global or population state variables to the graph.

On this same screen, you can also specify if you want to see the graph updated each year, or after each iteration, or after all iterations are completed, or no graph at all. The graphing is slow, so if you are running lots of iterations (in which case the graph will slowly turn into a solid blob of ink that usually cannot be read), it is faster to show the graph at the end of each iteration, and even faster to run it without showing the graph at all.

Simulate:



Simulate - enables MMM to begin running the linked-programs A bar across the bottom indicates the percentage complete of the entire process.

When MMM is finished a small box saying "**Complete!**" will appear.

The graphic shows the changes in population size (or whatever variable you asked to be shown, on the previous specification screen) from the populations associated with the linked models.

Tip -- Colors represent the different plots chosen in the Simulate Options screen. Refer to the legend below the graph to see which line displays the data for which variable of each population.

After the run is complete you can edit the graph (double-click on the graph to pop open a graphics editor), print it, or save the graph (in any of various formats) and data (as an excel file).

After a simulation is done, you can hit Simulate again to just run it all over again. Or you can close the simulate box to go back to the previous screen, change options, and then re-run. Sometimes, backing up in screens causes MMM to get confused. If that seems to be happening, close MMM, restart the program, load your project (if you saved it before!), and resume your analyses.

Quit MMM- This button provides a way to exit the program after the simulation has completed.

To see output, you need to quit MMM and open up the individual programs and refer to their normal protocol for viewing output.

Running the MMM program will generate a set of output files. The files that are created will depend on the programs that are invoked in the metamodel. Unless the user specifies an "Output" step in the "Order Model Steps" screen, MMM itself won't produce any output files. That is left entirely to the programs running within the MM.

Chapter 3. Example.

This chapter reviews an example using the MetaModel Manager to model black footed ferrets and prairie dogs with sylvatic plague. The data files for this example are included in the installation of MetaModel Manager, placed into a C:\MetaModelManager\Projects\Samples folder. Additional sample MMM projects are available on the MMM website, at www.vortex10.org/MetapopDzDemo.zip and www.vortex10.org/PredatorPreyDemo.zip.

Background: This example models the demographic responses to dynamic predator-prey relationships in the presence of disease. Black footed ferrets are perhaps one of the most endangered species in the United States. They are specialized on eating prairie dogs, which make up over 90% of the food ingested by the ferrets. The prairie dogs are subject to the sylvatic plague, which can reduce population size dramatically. Please note that in many cases, parameters are based on fictional information. The example presented below should be interpreted for educational experience only, and not cited as factual.

BFF-PD TUTORIAL:

Opening screen

1) Select ‘**Begin a New Project’** from the opening screen

“Setting System Level and Modifier Programs” screen

1) From the Population/ System Models first drop down box, select ‘**Vortex**’
2) Set you Vortex (.vpj) project by opening the ‘**choose**’ link, and selecting ’**MMM_PDog.vpj**’. You will need to navigate to this file in the C:\MetaModelManager\Projects\Samples folder (or wherever you may have stored it after the MMM installation).
3) Next choose **‘PD MM ReproEffect’** scenario from the dropdown box.
4) Click on “Add a Modifier Model”, and then choose ‘**Outbreak’** from the drop down box.
5) Under Project File, click on the “…” icon to open a file dialog and select the PDPlague.xml Outbreak specification from the C:\MetaModelManager\Projects\Samples folder.
6) Now click on “Add a System Model”, and then select ‘**Vortex**’ from the Model drop down box.
7) Set your Vortex (.vpj) project by opening the ‘**choose**’ link and selecting ‘**MMM_BFFerrets.vpj’**
8) Next choose ‘**BFF MM Worse’** scenario from the dropdown box.
9) There is no other linked model to set for Black Footed Ferrets, so now select ’**Next**‘ to move to the next screen.

“DataSet Definition” screen

1) Everything should come up by default, showing the GSvars and PSvars used by theVortex and Outbreak projects. Select ‘**Next**’

“Order Model Steps” screen

1) Change Outbreak to have 365 time steps, so that it will run a daily simulation. Vortex should remain at the default timestep (1) for 1 year.

2) Change total years to 100, and Iterations to 50. (Note that you will want to run more iterations for actual scientific analyses – here we suggest a small amount just for the sake of time, since running more iterations takes significantly longer time to execute. Hit **'Next'.**
3) At the Save this project before continuing prompt, select '**Yes**', and name it 'MyFirstTutorial'.

"Simulate Options" screen

1) Note that the population sizes will be displayed (the default). Select **'Show plot lines in real time'**. Hit **'OK'**.

"Simulate" screen

1) Select '**Simulate**'. The simulation will start. Do not be surprised if the screen fails to show the graph correctly. The graphics in some cases are not able to keep up with the simulation processing. Also, if you click off the screen (to move control to a different window), the simulation screen will probably stop being updated during the iterations, and may display a message that Vortex is "not responding". Do not worry, as the simulations are continuing to work away in the background. The final graph will be displayed correctly when the simulation is finished.
2) When the processing is finished, you will see a '**Complete!**' box pop up. You can select okay to make it go away and see your graph again.

Congratulations! You have completed your first metamodel analysis of a dynamic, two-species system in which the prey species is subject to epidemics of disease. The final figure generated should show prairie dog population size fluctuating quite a bit because of disease, sometimes leading to the decline or local extinction of the black footed ferrets. The prairie dogs themselves rarely go extinct, although drop significantly in numbers on some iterations.

To learn more about the options available in MMM, select Quit MMM, and start over again this time calling the saved 'MyFirstTutorial' MMM project –

Opening Screen

1) Select '**Open an Exisiting Project'** from the opening screen
2) Select '**MyFirstTutorial**'

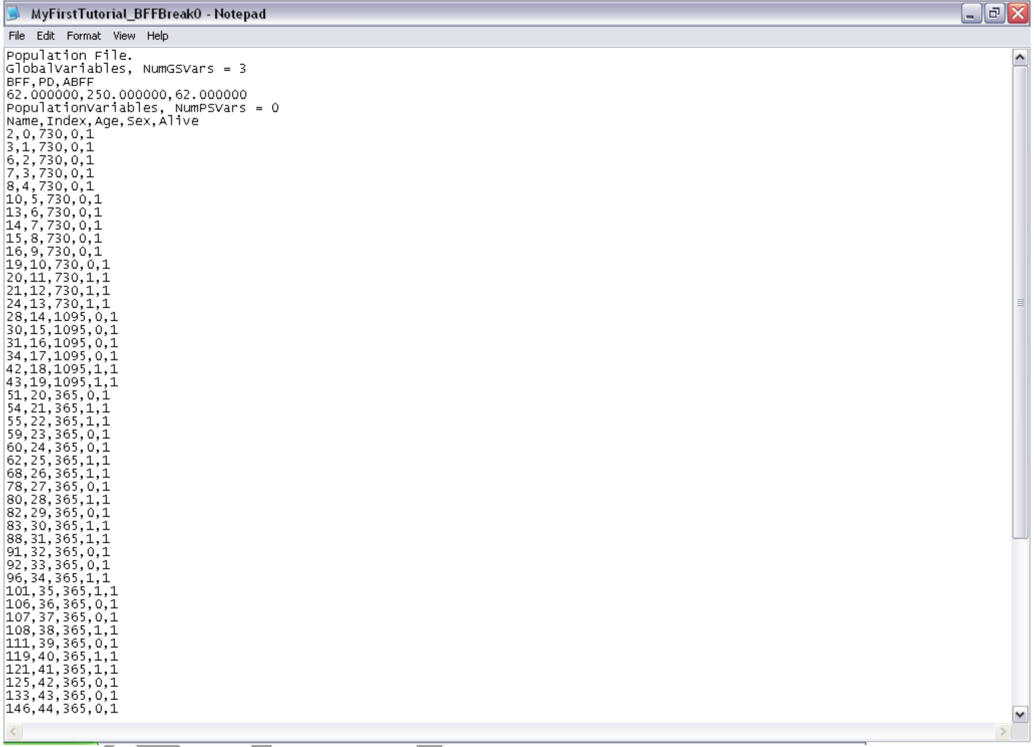
Order Model Steps

1) Create a break at the end of the second system level run by selecting '**+ Break**' and making sure it is the last item on the model list. Choose for the break to occur every 10 years
2) Also, change Total Years to 50, and iterations to 1, so the metamodel will run faster

Simulate Screen

1) Rather than launching into the summary graph, a screen will pop up showing statistics of the populations at the desired intervals.

2) With ‘Vortex(1) – Population 1’ highlighted, Select ‘Show Population’ and the population at time step 0 will appear for PDs including their Outbreak disease state characterisitics.
3) With Vortex(2) – Population 1’ highlighted, Select ‘Show Population’ and the population at time step 0 will appear for BFFs. There is no disease state information to show because Vortex was run alone for this population without Outbreak as a modifier. Let save the statistics of BFFs to a file. Slect ‘Write to File’ MyFirstTutorial_BFFBreak0


MyFirstTutorial_BFFBreak0 - Notepad
File Edit Format View Help

```
Population File.
GlobalVariables, NumGSVars = 3
BFF,PD,ABFF
62.000000,250.000000,62.000000
PopulationVariables, NumPSVars = 0
Name,Index,Age,Sex,Alive
2,0,730,0,1
3,1,730,0,1
6,2,730,0,1
7,3,730,0,1
8,4,730,0,1
10,5,730,0,1
13,6,730,0,1
14,7,730,0,1
15,8,730,0,1
16,9,730,0,1
19,10,730,0,1
20,11,730,1,1
21,12,730,1,1
24,13,730,1,1
28,14,1095,0,1
30,15,1095,0,1
31,16,1095,0,1
34,17,1095,0,1
42,18,1095,1,1
43,19,1095,1,1
51,20,365,0,1
54,21,365,1,1
55,22,365,1,1
59,23,365,0,1
60,24,365,0,1
62,25,365,1,1
68,26,365,1,1
78,27,365,0,1
80,28,365,1,1
82,29,365,0,1
83,30,365,1,1
88,31,365,1,1
91,32,365,0,1
92,33,365,0,1
96,34,365,1,1
101,35,365,1,1
106,36,365,0,1
107,37,365,0,1
108,38,365,1,1
111,39,365,0,1
119,40,365,1,1
121,41,365,1,1
125,42,365,0,1
133,43,365,0,1
146,44,365,0,1
```

4) Select OK and the simulation will proceed, but pause again at year 10. Show population of BFFs and save to file again, this time as MyFirstTutorial_BFFBreak10. Keep doing this through the first iteration.

Chapter 4. Template for open data exchange via MetaModel Manager

We hope that in addition to the demographic, animal movement, and disease spread program suite built by various members of the Biocomplexity Network, you will find opportunities to build your own programs in conjunction with MMM. If you choose to do so, your program must be developed in a way that is compatible with the MMM. You can use the information and template described below to help set up the coding for your own programs so that they are compatible with the MetaModel Manager and can share system, population, and individual data with other linked programs. We won't pretend that those who are new to programming will be able to write computer code for a model that is then linked to others via MetaModel Manager. You need to be experienced in a computer language such as C#, Visual Basic, Delphi, or Java to be able to write the functions necessary to have programs developed in these languages access data in MMM. (However, even if you have no expertise programming in such languages, you can write simple macros or scripts – as you might in Excel or for a statistical package – to carry out relatively simple data manipulations using the MMMacro or Evaluator utilities that are included with MMM. MMMacro handles all the data exchange with MMM for you.)

To add your own Modifier models to the toolbox available to MetaModelManager, you will need to create a .NET compatible dynamic link library (dll) with functions that provide the interface through which to access the Global, Population, and Individual data contained within class MData. The standard function calls that MetaModelManager uses to communicate with any linked model are Initialize(), Simulate(), and Close(), although these three functions can be given different names if the synonymous names are specified to MetaModelManager in the model specifications.

The structure of the metamodel data in MData is defined in the small library MMMCore. The C# code for the MMMCore.dll is provided in the MetaModelManager installation file.

Another small code file – MMLinkTemplate.cs – is included to provide a template for the three key functions for passing data between MetaModelManager and a user-provided model. In addition, the code for the MMMacro modifier program which provides a means to modify variables in MMData via macros or scripts is provided to show a more complete and complex example.

Your program will appear on the program list on your computer if you set it up to do so. It will only be resident on your computer. This is done via the Manage Loaded Apps option on the opening screen of MMM.

If you think your program may have broader utility to the conservation community, then we encourage you to make your program available for others to use. The metamodeling tools website (www.vortex10.org) will provide downloads for relevant software that developers are willing to distribute freely, and will provide links to other sites for programs that charge a fee or are otherwise restricted in their use.

Chapter 5. Understanding the use of State Variables to model biological links

Vortex offers the infrastructure to set up state variables in order to provide extra information regarding an individual, population, or metapopulations. Defining and using these variables is key to creating successful linkages between programs in order to express how the parameters of one population may be influenced by outcomes from other programs. Vortex has three levels of state variables: global, population and individual. These state variables must be numerical, can change over time. (See the Vortex manual for more information on how it creates and uses state variables.)

Other modeling programs may also provide the means to set up state variables, or may have pre-defined state variables that are shared through MMM, or may read (from MMM) the variables created by other linked programs. For example, Outbreak does not provide a way for the user to create his or her own state variables, but Outbreak always creates a population variable called "Prevalence", and individual variables called "DiseaseState", "DaysInState", "StatePermanent", and "IsVaccinated". In addition, Outbreak can use any other population individual state variables created by other programs and handed to it by MMM.

Any program linked to MMM can access the state variables defined by other linked programs, or can create new variables and add them to the data set (MData) that is shared with all linked programs. This allows each linked program to read and optionally change the population and individual data created by other programs.

Just as writing your own model to be linked in to others via MMM is not something that can be explained quickly in this manual, understanding how to make use of the data that do flow among linked programs is also not something that will be immediately obvious to a new user. Later versions of this manual will provide more detailed explanations and examples, but for now an example is very briefly described below.

Example: Linking Vortex-Vortex
In a two-species two-Vortex metamodel, information from each species may be coded into global state variables such as in the case of BFF-PD demonstrated in the earlier chapter. The following is a more detailed description of how to set up that linkage.

1) Creating the Operational Link:
A key to creating a link to share data between two Vortex models is to define the same (or at least overlapping) set of global or population state variables (GSvars and PSvars) that will be shared by both programs but, in most cases, modified only by one of the two models. In this example, GS1 is set to be the number of ferrets and GS2 is set to be the number of prairie dogs.

For BFFs in their own BFF model, the number of individuals (GS1) is set each year to "N", the number of individuals generated by the model. For BFFs in the PD model, the BFF population size (GS1) is first set to 50 and then assigned to remain to be GS1, which means that the PD model will not change this GS1 variable but will read the number of ferrets for use in specifying impacts (predation!) on the prairie dogs.

GS2, the number of PDs, is set in the PD model each year to "N", the number of individuals generated by the model. In the BFF model, the number of PDs (GS2) is first set to 250 and then assigned to stay as GS2, which serves to read the information from the PD model but not try to change the number of prairie dogs from within the ferret model.

Global State Vars	Label	Trophic Level			
		PD (Prey)		**BFF(Predator)**	
		Init In	Transition In	Init In	Transition In
GS1	BFF	50	=GS1	=N	=N
GS2	PD	=N	=N	250	=GS2

2) Creating the Ecological Link:
The ultimate ecological link between the two Vortex models occur when the global state variables are later used in equations to parameterize the demographics of each of the two populations. In our example, the mortality of PDs in the PD model for females from age 0 to 1 equals 45+((15*GS1)/N). The fewer PDs present in the population relative to BFFs, the more young PD mortality goes up. In the BFF model, first year mortality is specified as 46.6+((20*N)/(1+GS2)), so that a higher ratio of predators to prey reduces the survival of young ferrets.

Links between models may also be made through using individual state parameters, whereby specific information is tied to specific individuals in the population. In the BFF-PD-Plague metamodel, IS1 tracks the DiseaseState of each prairie dog. DiseaseState is determined within the Outbreak model of plague, and transferred to the Vortex model by MMM. Vortex then uses this IS1 variable to modify the reproductive rate of prairie dogs, with a function that specifies that currently sick (DiseaseState 'I': IS1=3) females cannot reproduce. Vortex ignores the disease status of each individual when setting mortality rates, because the death of prairie dogs from plague occurs within the Outbreak model (with 50% mortality of diseased PDs).

We do not want to imply that these few simple relationships describe accurately the functional response of predator to prey and the reverse; we just inserted some simple functions to show how such relationships can be specified and modeled within a 2-species metamodel, with disease impacting one of the species. A more complete and realistic metamodel for ferrets and prairie dogs might include reciprocal effects on adult survival of both species, effects on reproduction of ferrets, and possibly other functions defining the interactions between the species.