

Tutorial:

Introduction to Alternative Future Scenario Analysis and Envision

Contents

Tutorial:.....	1
Introduction to Alternative Future Scenario Analysis and Envision.....	1
Lesson 1: Basics of the Envision Interface	2
A. Getting to Know IDUs and the Envision Interface.....	2
B. Running a Scenario	10
Lesson 2. Adding Vegetation Dynamics.....	13
Lesson 3. Introducing Actors and Policies	14
Edit Policies - The “Policy Editor”	15
Defining Scenarios.....	17
Adding Actors.....	17
Additional Simulation Setup Options	18
Lesson 4. Adding Fire to the Landscape	20
Glossary	24

Lesson 1: Basics of the Envision Interface

In this lesson, you will learn the basics of working with the Envision interface by exploring the landscape data for the Blue River Watershed (the sub-basin of the McKenzie River Basin that contains the Andrews Experimental Forest). Along the way, we will define key terms and concepts used in Envision.

A. Getting to Know IDUs and the Envision Interface

Step 1: Installing Envision

If Envision has not already been installed on your computer, the first step is to get the installation package from the website and run it. The best option is to install the software in a directory called [d:]\\Envision. The drive letter (C, D, etc) does not matter. The 64-bit version of the software can be found at: <http://envision.bioe.orst.edu/Downloads/Setup-x64.msi>. A 32-bit version of the software can be found at: <http://envision.bioe.orst.edu/Downloads/Setup.msi>.

If you are running a 64-bit version of Windows, you should use the 64-bit version of Envision. Otherwise, install the 32-bit version. This installs the main Envision executable files and a standard set of plug-ins.

The next step is to get an example Envision study area project from the Envision website.

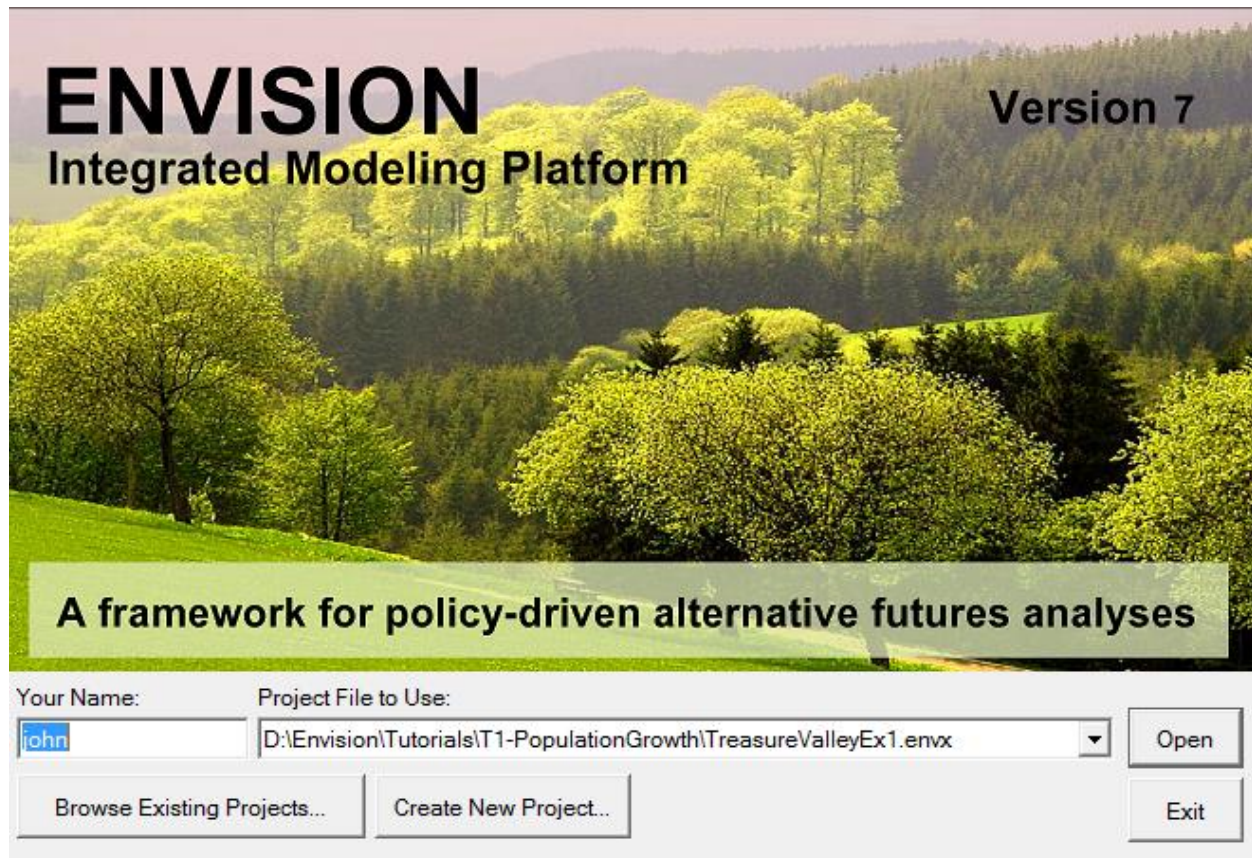
The datasets used in this tutorial are at <http://envision.bioe.orst.edu/Tutorials/Tutorials.msi>. Using the default install directory will allow the files to run unmodified. When you do that, you should end up with a directory [drive:]\\Envision\\Tutorials\\that includes a couple of subdirectories and a number of different files.

At this point, you should have installed both the Envision executable files and a set of input files (termed a “Project” in Envision terminology), and are ready to run Envision.

Step 2: Running Envision

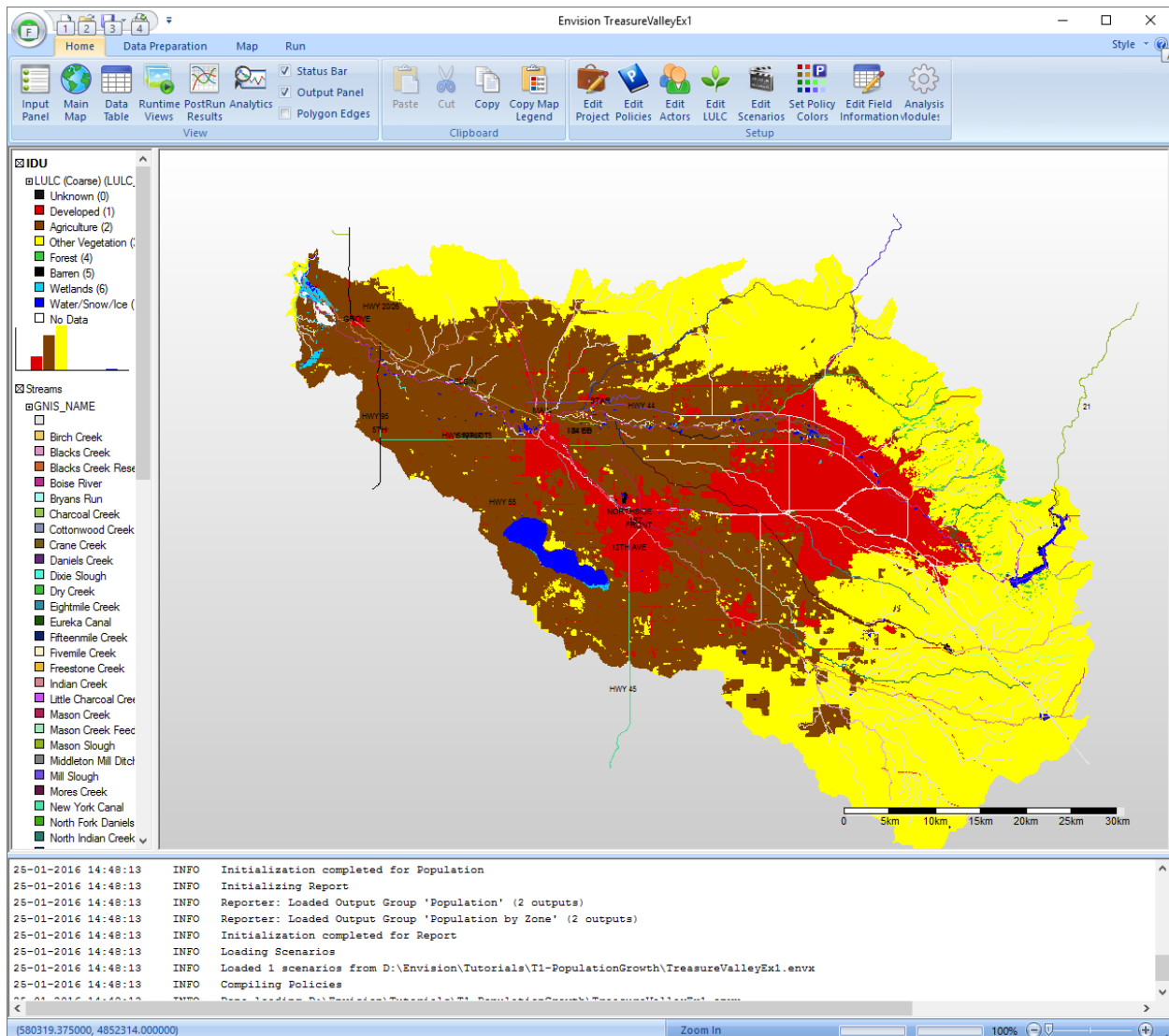
To run Envision, following the normal steps for the version of Windows you are running. You should have an Envision icon that will launch Envision, or run Envision from your Start menu.

When you first launch Envision, you should see a window like the one below:

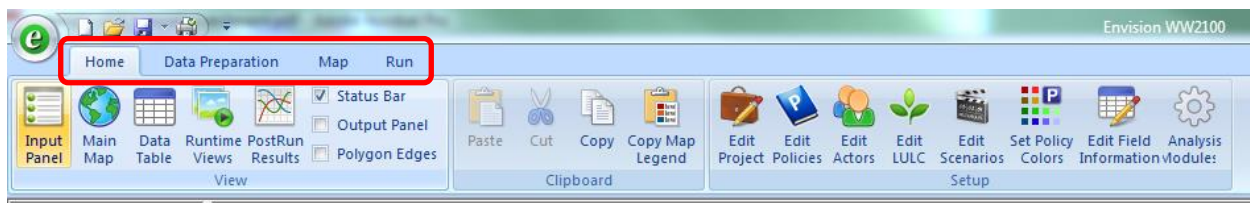


Click on “Find Existing Project” and map to the directory that contains the sample project. Navigate to **[drive:]\Envision\Tutorials\T1-PopulationGrowth** (or wherever you installed the tutorial) and select the **“TreasureValleyEx1.envx”** file. Click on “Open” to open this project.

When you open the project, there will be three map layers displayed. One is the polygon coverage of Independent Decision Units (IDUs). The second is a Stream network coverage, and the third is a Roads coverage. In addition, a plug-in module, *Target*, was loaded –it is used to model the spatial distribution of new population growth. A second plug-in, *Reporter*, was also loaded –it provides an easy way to get outputs out of Envision into comma-delimited text files – CSVs. We will discuss more about how these plugins are developed and specified later on. The software should now look something like this:



The project includes three shapefiles, one called IDU and the others called Streams and Rivers. The map is initially classified based on a particular attribute in the IDU shapefile, called LULC_A.

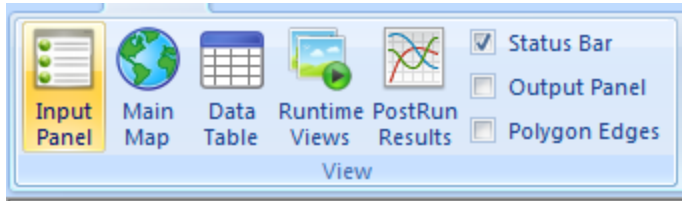


1. Look at the tabs at the top of the interface. These ribbon tabs let you flip between groups of operations in Envision. The four tabs are:

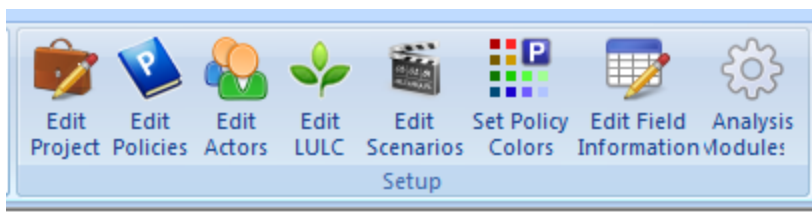
- a. **Home** – basic functions to edit projects and policies.

- b. **Data Preparation** – GIS tools to edit the attributes of the IDUs. For example, to add or remove fields or calculate a field based on other attributes.
- c. **Map** – GIS tools to view and query the data sets.
- d. **Run** – Tools to set up policies and run Envision.

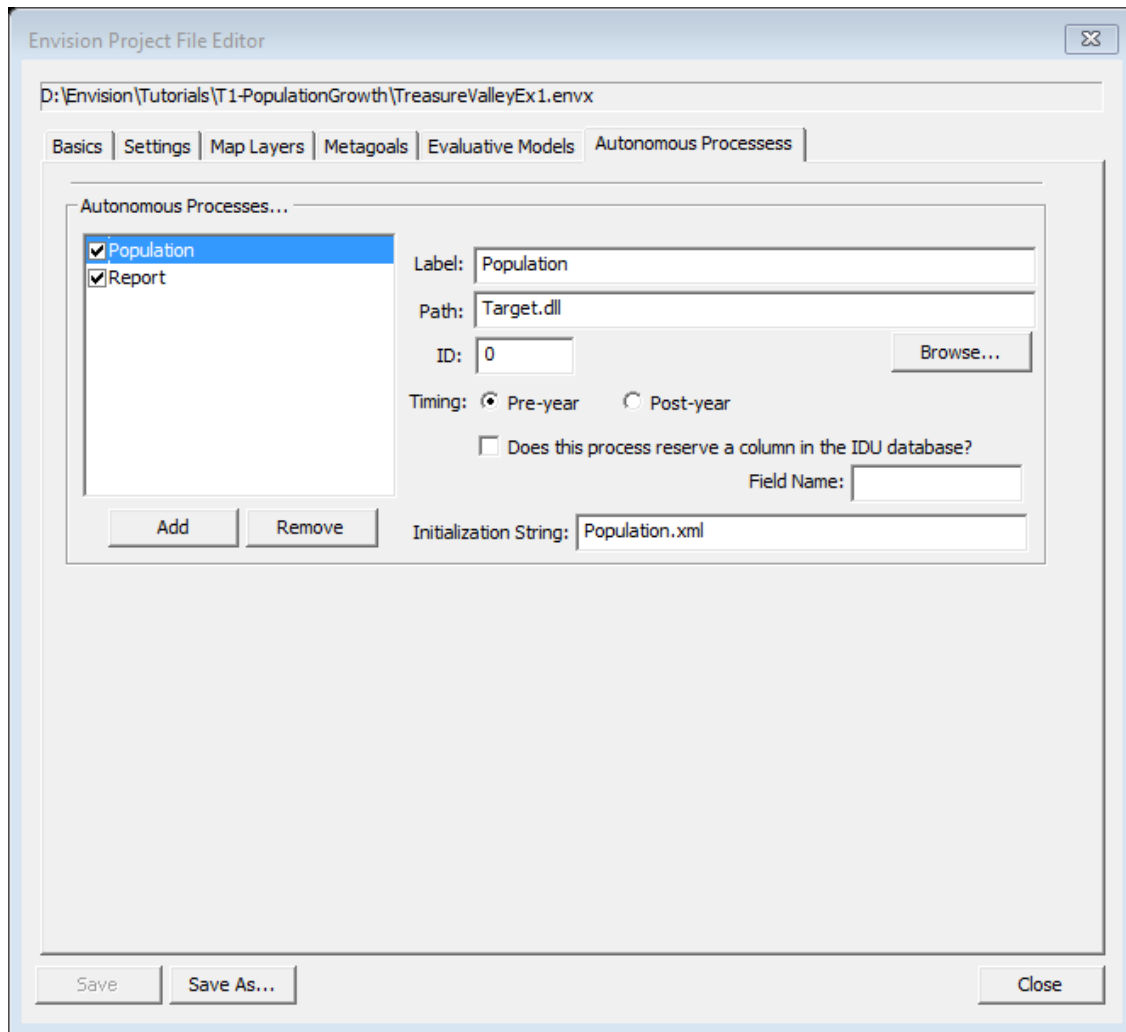
The tutorial sections below will walk you through some of the functionality on each tab.



2. When you are on the “**Home**” tab, the buttons in the “**View**” box change what is displayed in the main content area. These buttons include:
 - a. **Input Panel** – this is in development and we will not be using it for our work
 - b. **Main Map** – Displays a navigable map of the map layers in the main window. In this part of the tutorial, we will work with the IDU coverage. IDUs are the attributed polygons that Envision uses to represent the landscape and are the basic processing units of Envision. Any landscape specific input data that will be used by plug-in models or analyzed in scenarios is included as an attribute of the IDUs.
 - c. **Data Table** – Displays a view of the data tables underlying the map layers. For example, look at the table for the IDU coverage (IDU.dbf). Each row contains the data for one IDU. The columns or “fields” are the associated attributes of the IDU. The data table view includes tools to find and query IDUs with specific characteristics.
 - d. **Runtime Views** – these are in development
 - e. **PostRun Results** – Displays run results. This will be blank until you have carried out some model runs.
 - f. **Status Bar** – Displays the status of a run, e.g. how much is completed in bar form and percent (right side). On the left side it also shows the coordinates of the mouse location in the coordinate system of the data file (in this case UTM).
 - g. **Output Panel** – Displays the sequence of processes as they run. This can be useful for problem solving—for example to track when errors occur or to look up the path of where data layers are stored.
 - h. **Polygon Edges** – Displays the perimeters of polygons. Helpful if you want to zoom in to the boundaries of particular IDUs.

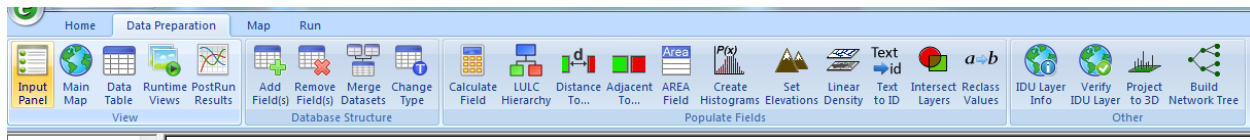


3. Also on the “**Home**” tab, the “Setup” box includes tools to edit the project and its policies and scenarios.
 - a. **Edit Project** – Displays a pop-up window that allows you to edit basic properties of the project, such as the input map layers. It also allows you to import elements from other projects. If you look at the Autonomous Processes tab, you can see models that have been loaded, whether or not they are being used for the particular run, and also what some of the inputs/parameters related to the models are. For this example project, we are only running to plug-in modules, named Population and Report; each is checked to indicate it is active.

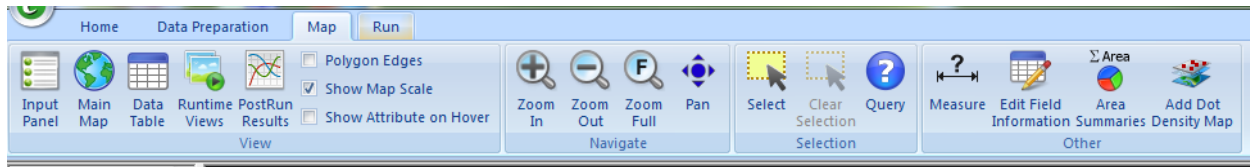


We will talk about some of the other elements here as the course continues, but for now you can ignore them.

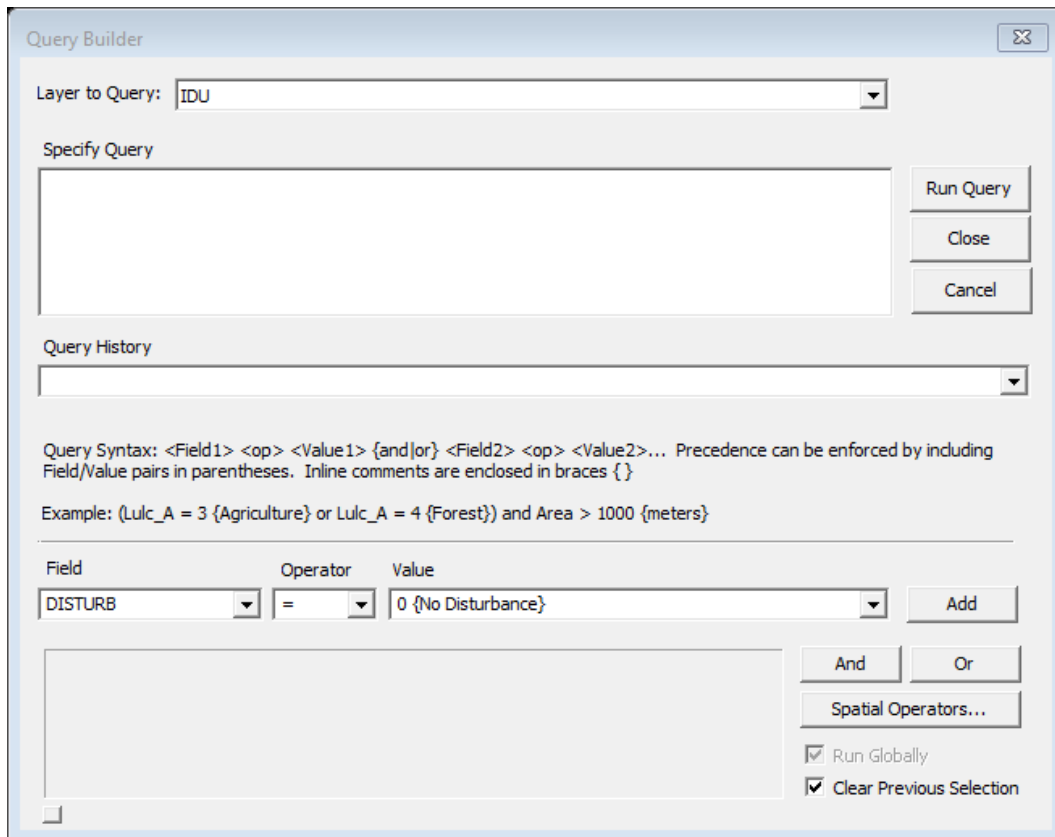
- b. We will work with the other buttons in this box (“Edit Policies”, etc.) in a later section of this tutorial.



4. The “**Data Preparation**” tab provides tools for editing the IDU polygon coverage. We will not work with these tools in this exercise.

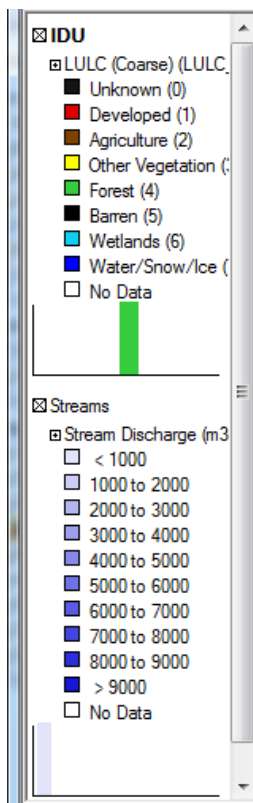


5. The “**Map**” tab includes tools for moving around and displaying information about the landscape:
 - a. **Pan and zoom tools** (in the Map Ribbon Tab).
 - b. **Selection and query tools** (in the Map Ribbon Tab). Tools to select particular IDUs so that you can summarize information or perform operations only on a defined set of polygons.
 - c. **Area Summaries** – A tool to make measurements on the map and calculate total areas for specific attributes. It allows you to specify queries, which can be entered directly or developed with a query builder. To get a query builder, click on Query, and see description below.
 - d. **Add Dot Density Map** – this functionality won’t be used during the class
 - e. **Choosing the Attribute to Display on the Map** – If you right click in the grey area surrounding the map, you can select different attributes of the IDU polygons to display. For example, try displaying a more detailed version of the land use/land cover (called LULC - Fine) or the land ownership type (e.g. federal, private etc.). Note that each one of these attributes corresponds to a “field” in the IDU’s data table. The information that shows up in the popup box is outlined in the Field Information interface, which you can explore by clicking Edit Field Information. We won’t go into more detail about this today, but it will be an important component of some elements of the class project. Feel free to explore the editor if you wish.
 - f. **Displaying Information for a Specific Polygon** – If you zoom in and right click on a polygon, you can display attributes or take actions specific to that polygon. For example, you can display all of the attributes for that polygon (the “Properties of the Site”).
 - g. **Query** – This dialog allows you to build a query, select polygons that satisfy that query, and it provides a brief statistical summary of the results. It looks like:



In this case, you can click Add to add the suggested query to the Query Builder, and after that, click Run Query to make use of it.

6. The **“Map Legend” panel** includes additional tools for viewing and managing information about the currently loaded map layers. It shows, for each map layers, the currently active field (the field currently being displayed on the map) and the legend associated with the active field. The checkbox next to the layer name controls the visibility of the layer. The order of the layer, if multiple layers are defined, determines the drawing order. The “top” layer is drawn first, with subsequent layers drawn as one moves down the list: layers lower in the list are drawn “on top” of layers higher in the list. In general, user interaction with this panel is via the mouse. Some specific functionalities include:



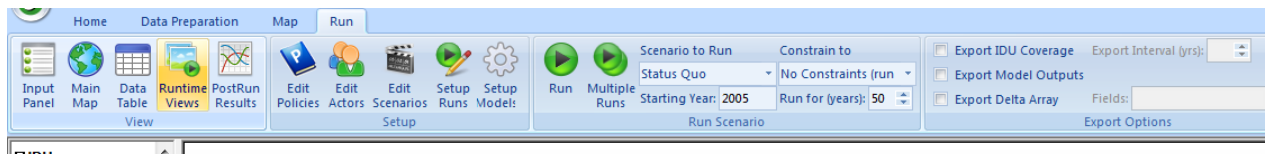
Layer Properties	Layer properties can be viewed/specified by right-clicking on the layer name and selecting “Properties” from the context menu. A number of properties, including layer type, number of fields, number of rows, IDU area statistics, labeling information, and field-specific information is available in the Properties view
Layer Transparency	Layer transparency can be set by right-clicking on the layer name and selecting “Transparency”
Legend Colors	Colors used to depict specific attributes can be modified by double-clicking on the color box associated the attribute

Adding “Overlays”	An Overlay allows multiple fields to be drawn using a single coverage. Overlay layers are generally drawn “on top” of the original layer using transparency to allow both layers to be visible
----------------------	--

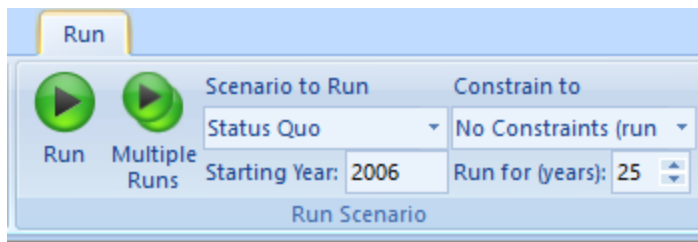
Practice Questions for Section 1A

1. Display the medium resolution land use/land cover data (LULC_B). What percentage of the watershed is covered by Cultivated Crops? (Hint: Use Envision’s query mechanism)
2. What portion of these Cultivated Crops areas are within a Rural Residential Zone (field ZONE in the coverage) (Hint: Again, use Envision query mechanism)
3. How many IDUs are there in the watershed? (Hint: Find the “Layer Properties”)
4. What is the size range of the IDUs? (Hint: Find the “Layer Properties”)
5. Is all of the area zoned for the reservoir, actually covered by water?
6. How much land has tree stands older than 100 years at the start of a simulation?

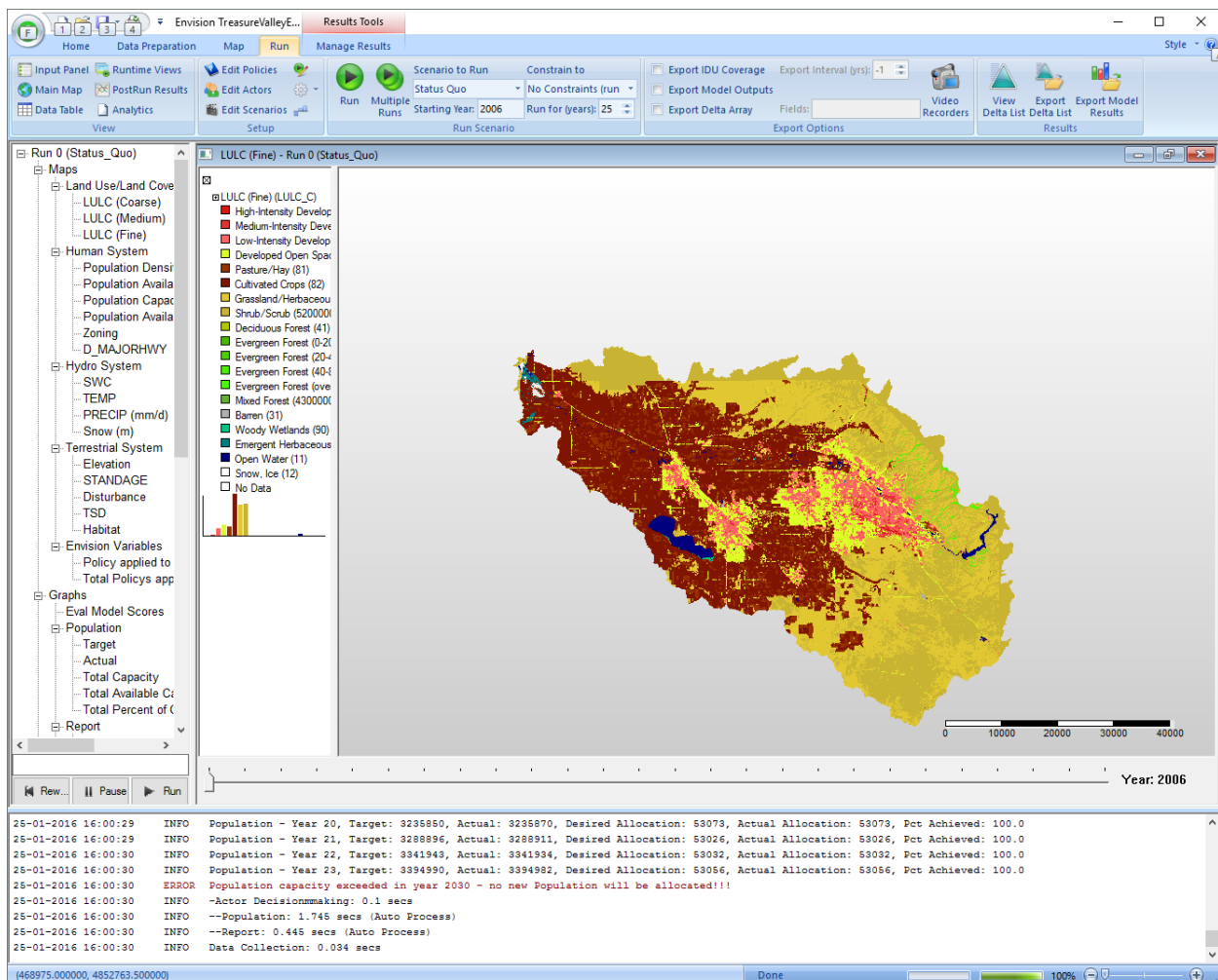
B. Running a Scenario



1. The “Run” ribbon tab includes tools for setting up and running scenarios in Envision. Scenarios are the projections of future change created by Envision. They consist of a coherent collection of “policies”, model settings, assumptions and preferences. As we go through the buttons and functions on the “Run” ribbon tab, we will define the elements of a scenario.

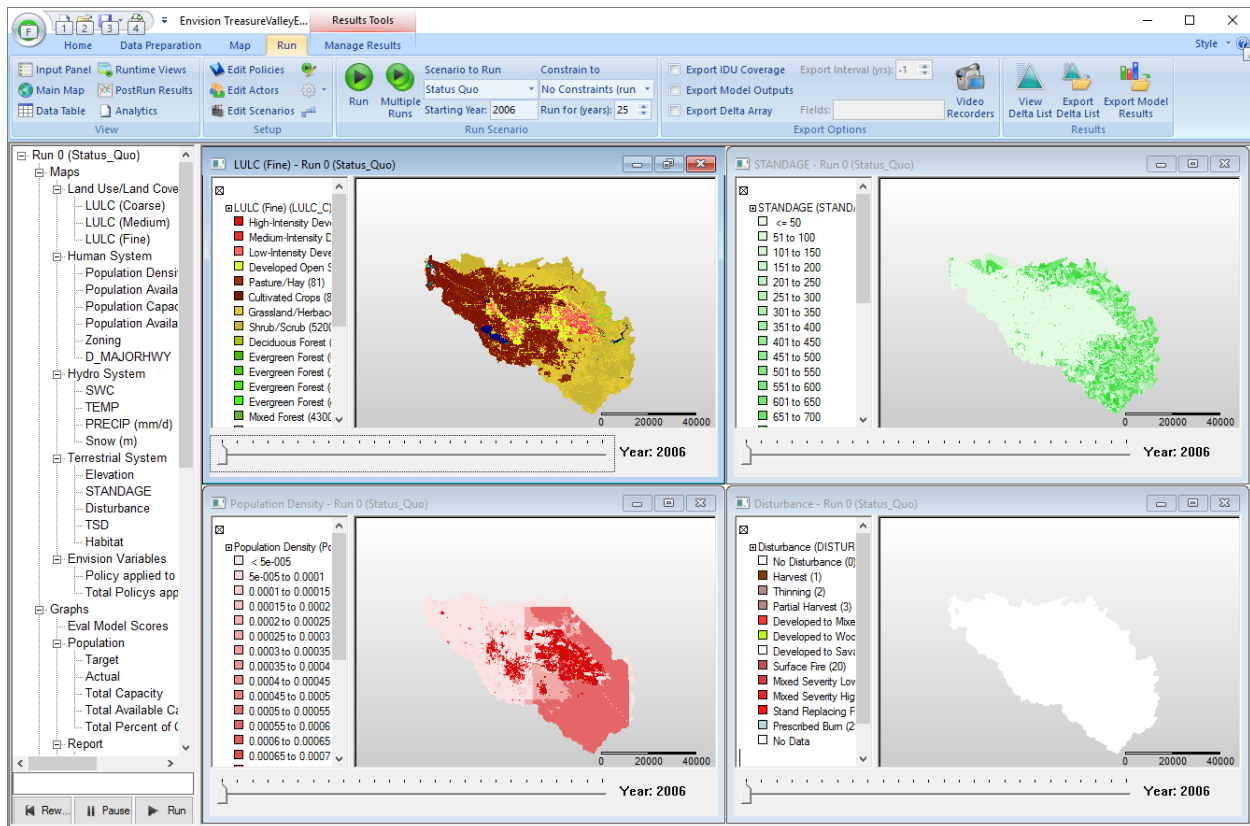


1. **Run Scenario** –The “Run Scenario” bar allows you to choose the scenario to run and to constrain where it runs (for example to a particular set of IDUs that you have selected with the query tools). It also allows you to select the starting year and duration of your run. There are no constraints on the starting year you choose, but it makes sense to choose a year consistent with the base data sets that make up the IDUs. In this example, the land use/land cover is from about 2006, so we start from there.
 - a. Try running the “Status Quo” Scenario (for the whole basin) with a starting year of 2006 and run time of 25 years. It should take 10 seconds or so. As it runs, right click to pull up the classification popup window, and switch between different landcovers, and perhaps other attributes. As the simulation runs, the plug-ins are updating the map as needed.
2. **Viewing Results** – Click on the “PostRun Results” button on the “View box”. The expandable Tree (Run 0 above) allows you to view results in a variety of formats. After running the “Status Quo” Scenario, you can explore the types of maps and graphs that can be displayed. The screen should look something like the following if you expand the hierarchy on the left and double click on “LULC (Coarse)”.



The Run buttons at the bottom of the screen allow you to view the results dynamically and watch how they change through the years of the model run.

Assuming you already double clicked LULC (Coarse), do the same for the other 2 LULC categories and also Stand Age. The screen now should look something like the following. These are depictions of the initial conditions for each of the 4 related elements of the system.



- Click on the “run” button in the lower left hand corner to watch how this attribute changes with time.

Practice Questions for 1B:

- How do you interpret the patterns that develop? Is this a good representation of the Status Quo?
- Examine the Envision project file (TreasureValleyEx1.envx) file located in the Tutorials\T1-PopulationGrowth folder in a text editor. The project file defines a number of settings, plug-in parameters, and many other aspects of an Envision application. It is an XML document with a specific structure. You will notice it is divided into a number of primary sections. Many of the items we will be discussing in this tutorial reflecting settings and inputs specified in this file.

Lesson 2. Adding Vegetation Dynamics

At this point, we have explored some of the input data associated with our sample project, set up a simple population allocation model, defined a set of reported outputs, and run the model through time. We used a plug-in (*Target*) to change the data in our maps through simulated time to represent the spatial allocation of new population entering our study area. Next, we will add an additional landscape model, this one for representing vegetation dynamics. We will employ a simple State Transition Model plugin (SSTM.dll) that allows use to define probabilistic and deterministic transitions between various application defined states, where a “state” is the value of a field attributed stored in an IDU coverage, and a “state transition” is a probability that the IDU will transition from its current state (say, “A”) to some new state (say, “B”) over one time step, generally considered to be a year.

Examine the Lesson 2 Project file, `Tutorials\T2-PopAndVegTrans\TreasureValleyEx2.envx`, in a text editor. Navigate to the `<autonomous_processes>` section, then continue to the `<autonomous_process>` defining the ‘Simple State Transition Model’, our name for this model. Notice the ‘init_info’ attribute – this specifies the input file configuring this model. This is a typical way of adding a new plug-in to an Envision application, and configuring that plug-in. Next, open the plug-in input file, located at `Tutorials\TreasureValley\sstm.xml`. Without going into the nitty-gritty details, you hopefully can get a sense of what this input file is specifying for “legal” transitions. The SSTM plug-in uses this information to implement a state transition model reflecting the transitions defined in the input file.

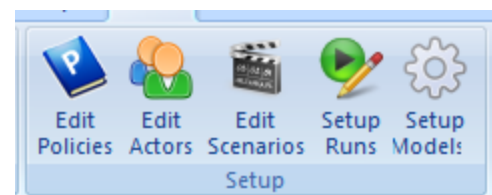
Also note that we added an additional Reporter entry for this lesson. Examine the `Reporter.xml` file in the `Tutorials\T2-PopAndVegTrans` folder to identify this entry. Details for Reporter inputs files are documented in the Envision Developer’s Guide.

Finally, restart Envision, load the `TreasureValleyEx2.envx` project file, and run a 25 year simulation. Look at the post-run results to view the dynamic maps and outputs associated with this lesson.

Lesson 3. Introducing Actors and Policies

At this point, we have explored some of the input data associated with our sample project, and we have run the model through time, using two landscape process models (*Target* and *SSTM*) for representing population growth and vegetation state transitions. Next, we will add actors and policies to the application. Specifically, we will introduce two policies allowing for conversion of agricultural lands and forest lands to rural residential zoning, and define a scenario in which this policy is active. The Project file we will use for this tutorial is at Tutorials\T3-AddingPolicies\TreasureValleyEx3.envx. In this tutorial, we've updated the Reporter.xml file with additional outputs, and introduce a new input file, Policies.xml, that contains the policy definition used in this tutorial.

The “Set Up” bar lets you adjust the suite of settings that will affect a scenario. We will go through each of the buttons and their function individually. The first of these is behind the button labeled Edit Policies. Click it and you will see the following dialog box:



Policy Editor

Policies

Ag Conversion to Rural Res

≤ ≥ Show Less

Basic Properties | Site Constraints | Resource Constraints | Outcomes | Effectiveness Scores | Scenarios

☐ This effect of this policy persists for a period after application
Minimum (yrs): 0 Maximum (yrs): 0

☐ This policy is mandatory
Percent Compliance: %

☐ This policy is a scheduled policy (must specify a start date)
Use Start Date: -1

☐ This policy excludes other policies from being applied at the same site
Use End Date: -1

Color: Set Color

Access Control: ☒ Share this policy with other users ☒ Allow other users to modify this policy

Narrative

This policy looks for lands that are current in an agricultural use that are next to rural residential zone areas, and converts some of these to RR zoning

Add New... Copy... Delete Edit Field Information... Import... Save... Cancel OK

Edit Policies - The “Policy Editor”

The Policy Editor dialog box allows you to view and define the characteristics of each policy that is part of the project. Policies capture rules, regulations, incentives, management actions and other strategies that will modify the landscape when the scenario runs. This will become clearer as you step through the setting for a policy:

Open the policy editor and use the drop down box at the top to view the “**Ag Conversion to Rural Res**” policy. The tabs in the box allow you to control when, how, where and what will happen when the policy is implemented. Click through the tabs to get a sense of the variety of functions that are available. We will only work with some of the tabs in this tutorial and will work with others next week.

Click on the “**Site Constraints**” tab. The expression in the “Site Attribute Query” defines how Envision will select IDUs where the policy will be implemented. For example, in this policy, the query selects IDUs that are in an agricultural land use (LULC_A = 2 {Agriculture}) and are adjacent to an area zoned Rural Residential (NextTo(ZONE=3 {RR})). You can type in these queries or you can use the drop down boxes at the bottom to build the queries.

The screenshot shows the 'Policy Editor' dialog box with the 'Site Constraints' tab selected. The 'Policies' dropdown at the top is set to 'Ag Conversion to Rural Res'. The 'Site Attribute Query' text area contains the query: 'LULC_A = 2 {Agriculture} and NextTo(ZONE=3 {RR})'. Below the text area, there is a 'Check Syntax' button and a syntax explanation: 'Syntax: <Field1> <op> <Value1> {and/or} <Field2> <op> <Value2>... Precedence can be enforced by including Field/Value pairs in parentheses. Inline comments are enclosed in braces {}'. An example query is provided: 'Example: (Lulc_A = 3 {Agriculture} or Lulc_A = 4 {Forest}) and Area > 1000 {meters}'. Below the example, there is a 'Map Layer' dropdown set to 'IDU' with a note '<<< Optionally specify layer (default is IDU layer)'. A table-like structure for building queries is shown with columns 'Field', 'Operator', and 'Value'. The first row has 'LULC_A' in the Field column, '=' in the Operator column, and '0 {Unknown}' in the Value column. Below this table is a 'LULC (Coarse)' dropdown. To the right of the table are buttons for 'Add', 'And', 'Or', 'Spatial Operators...', and 'Update Map'. At the bottom of the dialog are buttons for 'Add New...', 'Copy...', 'Delete', 'Edit Field Information...', 'Import...', 'Save...', 'Cancel', and 'OK'.

Policy Editor

Policies

Ag Conversion to Rural Res

Basic Properties | Site Constraints | Resource Constraints | Outcomes | Effectiveness Scores | Scenarios

Site Attribute Query

LULC_A = 2 {Agriculture} and NextTo(ZONE=3 {RR})

Syntax: <Field1> <op> <Value1> {and/or} <Field2> <op> <Value2>...
Precedence can be enforced by including Field/Value pairs in parentheses. Inline comments are enclosed in braces {}

Check Syntax

Example: (Lulc_A = 3 {Agriculture} or Lulc_A = 4 {Forest}) and Area > 1000 {meters}

Map Layer: IDU <<< Optionally specify layer (default is IDU layer)

Field	Operator	Value
LULC_A	=	0 {Unknown}

LULC (Coarse)

Add

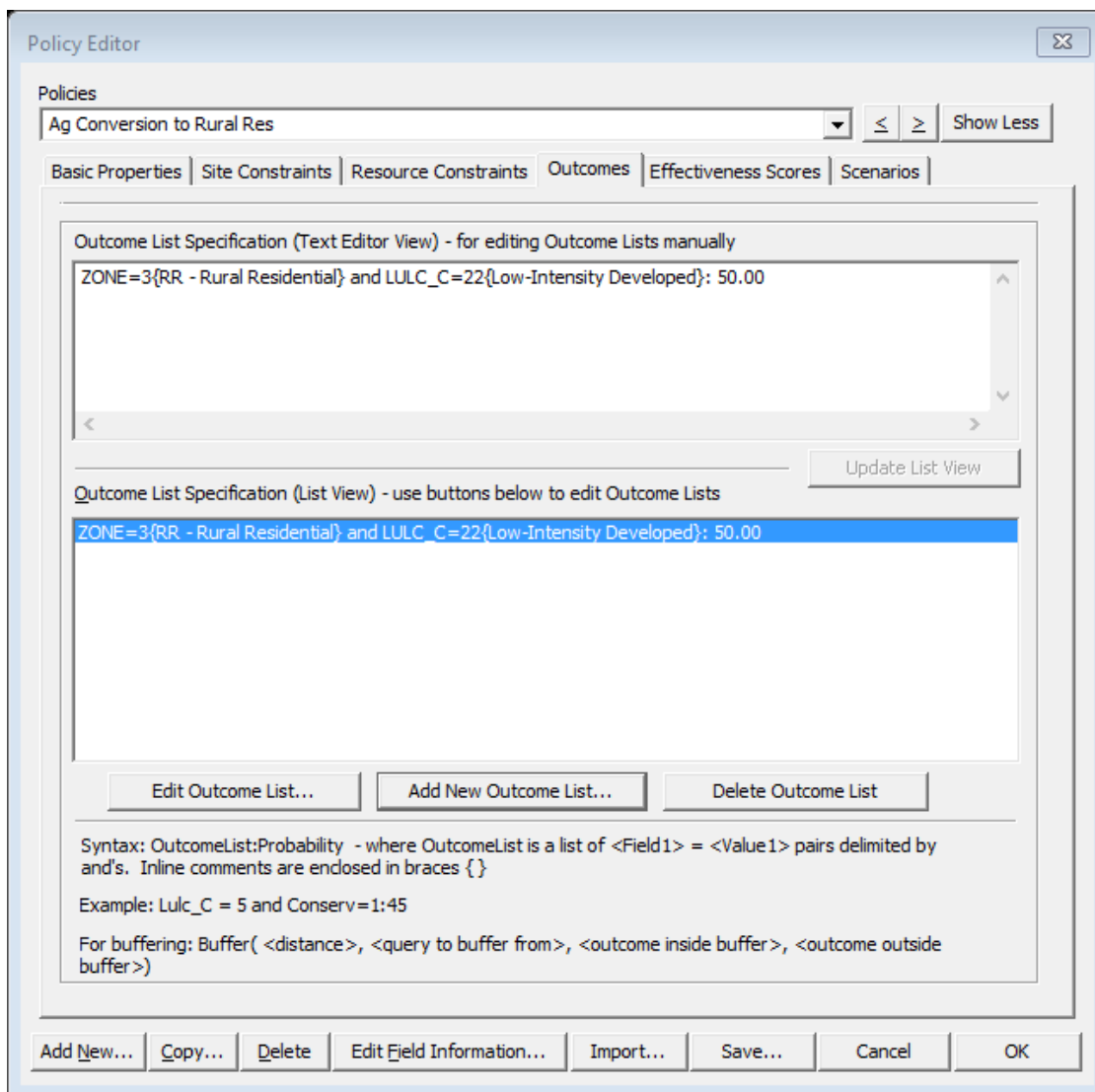
And Or

Spatial Operators...

Update Map

Add New... Copy... Delete Edit Field Information... Import... Save... Cancel OK

The “**Outcomes**” tab lets you control what will happen when the policy is implemented. The Outcome is written in a similar format to the “Site Constraints” query, in this example the expression is ZONE=3{RR - Rural Residential} and LULC_C=22{Low-Intensity Developed}: 50.00 meaning that when the policy runs, the ZONE field for IDUs selected for policy application will change to a value of “3”, the code for a Rural Residential zone, and the land use will be changed to Low-Intensity developed (LULC_C=22). The “50.00” in the expression controls the probability that the outcome will occur if an actor chooses to utilize this policy. In this case, the 50.00 indicates that each year, 50% of all the polygons that satisfy the Site Constraints and are selected by an actor for implementation will have this outcome. For the other 50%, nothing will happen as a result of this particular policy application..



The “**Resource Constraints**” tab allows you to add additional complexity to the policy implementation, for example to add an initial cost and annual maintenance cost for application of the policy. This feature uses a look up table and while we won’t work with it here, it’s use is documented in the Envision Developers Manual.

Similarly, we will not deal with the “**Effectiveness Scores**” in this tutorial. It is used to specify the actor values and landscape performance metrics this policy applies to.

The “**Scenarios**” tab allows you to view and edit the scenarios that use this particular policy.

Note that the buttons on the bottom of the policy editor allow you to create “new” or “copy” policies and to “save” any new or edited policies so that you can work with them again later.

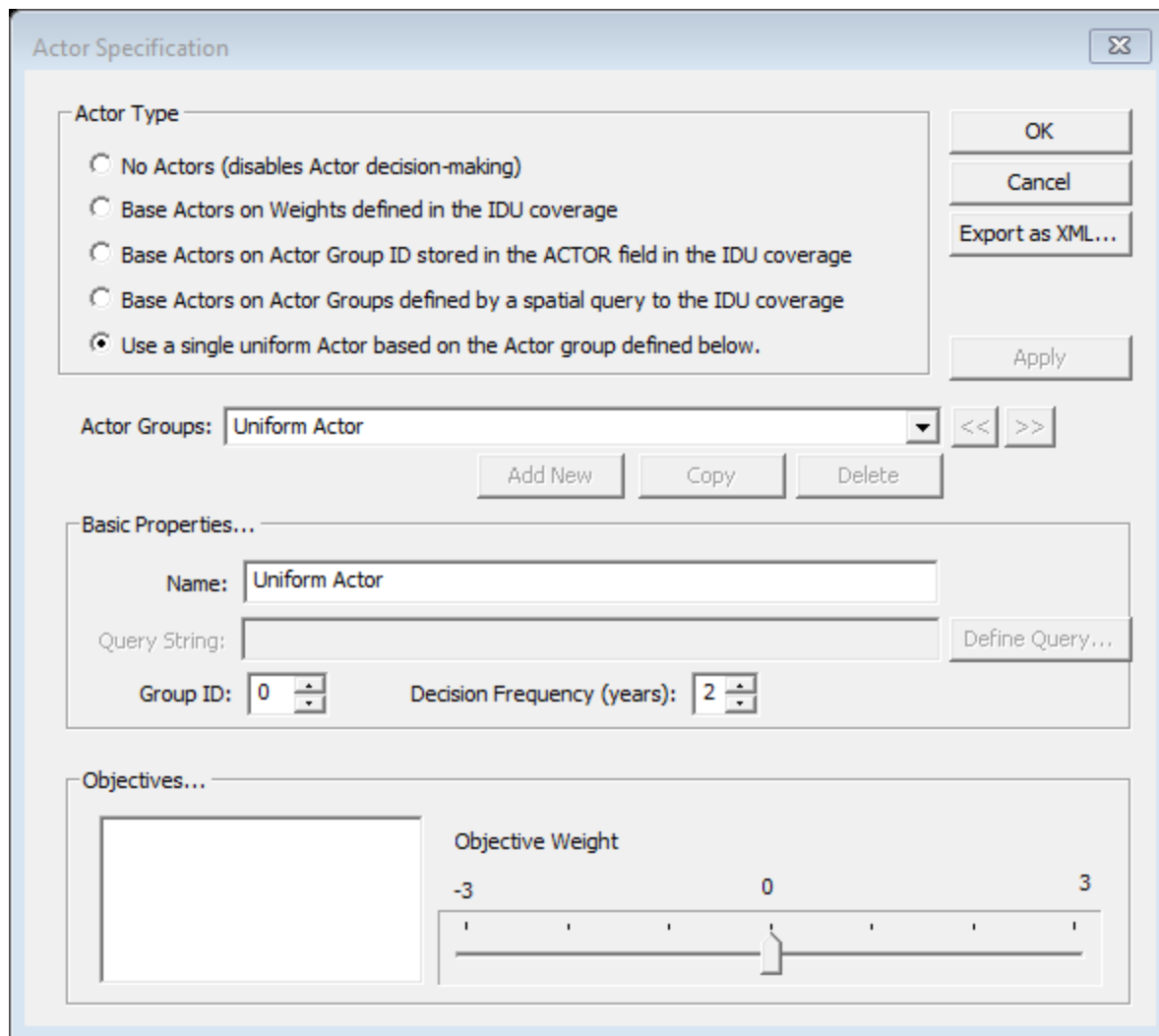
Defining Scenarios.

The Scenario Editor is the dialogue box that allows you to create scenarios and define which policies will be active in a particular scenario. In this tutorial, we have defined two scenarios, a “Status Quo” scenario, and an “Ag Conversion” scenario that allows agricultural lands adjacent to area zoned Rural Residential to be converted to residential use. As shown, in the “Status Quo” policy (shown above) the example policy is not enabled, but it is in the “Ag Conversion” scenario.

The screenshot shows the 'Scenario Editor' dialog box. At the top, there's a 'Scenarios:' dropdown menu with 'Ag Conversion' selected, and buttons for 'Add', 'Delete', 'Grid View...', 'Save...', and 'Close'. Below this, there are fields for 'Name:' (containing 'Ag Conversion') and 'Originator:', along with a checkbox for 'Default Scenario?'. A 'Description:' text area is also present. The main section is divided into two panes. The left pane, titled 'Scenario Parameter Settings', contains a 'Decision Weights' section with a note about policy preference scores and a 'Policy Preference Scores' input field set to '0.3'. Below this is a 'Landscape Model Evaluation Frequency' set to '1' year. The right pane, titled 'Policies To Include...', has a list box containing 'Ag Conversion to Rural Res' which is checked. At the bottom, there's a 'Model Settings' section with a tree view showing 'Process Variables' (Population, Simple State Transition Model) and 'Decision Elements'. To the right of this is a 'Model Variable Values' section with radio buttons for 'Use Default Value' and 'Use Specified Value'.

Adding Actors.

When you click on this “Edit Actors” button it opens the “Actor Specifications” dialogue box. One of the features of Envision is that it allows you to model the ability of different individuals or “actors” to make different decisions about the management of their land based on their own “values”. This dialogue box allows you to control whether there will be multiple actors with different “values” affecting their decisions. We will work more with this feature in a different exercise. For now, we will leave this box set as shown above, with a single uniform actor.



The image shows a software dialog box titled "Actor Specification". It contains several sections: "Actor Type" with five radio button options, "Actor Groups" with a dropdown menu and three buttons, "Basic Properties..." with text fields and spinners, and "Objectives..." with a list box and a slider.

Actor Type

- ☐ No Actors (disables Actor decision-making)
- ☐ Base Actors on Weights defined in the IDU coverage
- ☐ Base Actors on Actor Group ID stored in the ACTOR field in the IDU coverage
- ☐ Base Actors on Actor Groups defined by a spatial query to the IDU coverage
- ☒ Use a single uniform Actor based on the Actor group defined below.

Actor Groups: Uniform Actor

Buttons: Add New, Copy, Delete

Basic Properties...

Name: Uniform Actor

Query String:

Group ID: 0 Decision Frequency (years): 2

Button: Define Query...

Objectives...

Objective Weight

Slider: -3, 0, 3

Buttons: OK, Cancel, Export as XML..., Apply

For this tutorial, we want to create a single, uniform actor that will control the entire landscape.

Additional Simulation Setup Options

Setup Runs – This dialogue box adjusts settings for a run. For example, you can run multiple iterations, to look at the range of outcomes that might result from the combinations of policies and characteristics in a particular scenario. You can also alter how the results will be displayed and exported. For now, we will not use these elements.

Setup Models – This button provides access to any user interfaces that the plugins might define. For now, our plugins do not provide interfaces, so the button is non-functional.

Practice Set for Lesson 3 (explore on your own)

1. Examine the <scenarios> section of the
2. Run the two different scenarios, Status Quo and Ag Conversion, defined for this tutorial. After running the scenario, switch to the **PostRun Results** view. How much agricultural land was lost in each scenario?
3. Consider a policy to allow harvest of Evergreen Forests between 40-80 years old, converting them to young forests (age 0-20 years). You will find these attributes represented in the LULC_C field in the IDU coverage. You might take some of the following steps:
 - a. Use the Policy Editor to create a new policy.
 - i. To help with the syntax, try using the buttons at the bottom of the policy editor to start out by creating a copy of one of the existing harvest policies and renaming it to your new policy.
 - ii. Use the “Site Constraints” tab to set up the attribute query that will make the policy active on all of the IDUs with an evergreen stand between 40-80 years.
 - iii. Use the “Outcomes” tab to specify what will happen when the policy takes effect. In this case, you will change the LULC_C code to 4210000 (the code for age class 0-20years) each time the policy is applied. You can further control the frequency of application through the “Result Probability” setting for the policy.
 - b. Use the Scenario Editor to create a new scenario and turn off all of the existing policies and to turn on your new policy.
 - c. Try viewing your results (after running the simulation) using the “PostRun Results” button, and note that you can compare the results from different runs (or scenarios) directly.

Lesson 4. Adding Fire to the Landscape

In this lesson, we will add a representation of a landscape disturbance process, wildfire, again relying on existing Envision plug-ins to provide this functionality. More specifically, we will use a standard plug-in called *Spatial Allocator* to “allocate” fires to the landscape, specifying information about fire frequency, distributions of fire sizes, and fire severity. Additionally, we will track, in the IDU coverage, the locations and severity of these fires, and the “time since disturbance”, tracking for each IDU through time, so that we will know, at any location, how long it has been since that IDU experienced fire.

Examine the Project file at `Tutorials\T4-AddingFire\TreasureValleyEx4.envx` with a text or XML editor. You will notice in the **<autonomous_processes>** section, we have added three new processes to the set of modules used in this project. The most critical of these is the process named “Fire” that utilizes the Spatial Allocator plug-in. As is generally the case, this plugin is configured with an XML input file named (somewhat unimaginatively) “SpatialAllocator.xml”, located in the same directory as the project file. Open this file with a text or XML editor as well. This file defines two “allocations”, one reflecting low severity fires, the other stand replacing fires. Additional attributes specify the area burned as a function of time, probabilistic information about area burned in a fire, and various constraints and “hints” on where fires are likely to ignite, based on underlying landscape attributes. At runtime, the plug-in will generate fires on the landscape through time that conform to the attributes specified above.

In this example, we also want to track “time since disturbance” (TSD), which involves adding a process that runs each year, and for each IDU checks to see if it has been disturbed in that year (meaning, burned), sets a field in the IDU called “TSD” to a value of 0 (zero) to signal that we need to start tracking TSD for that IDU in successive time steps. This is accomplished with a plug-in called **Sync** that allows fields for an IDU to be updated whenever certain conditions are observed in that IDU. Looking at the project file, you will see the entry for the **Sync** autonomous process. It references a file called “sync.xml” as its input file. That is accomplished by the second entry in “sync.xml” (located in the same directory as the above files).

The first entry in the “sync.xml” input file serves a different function. Whenever a fire occurs in a time step, the Spatial Allocator process set the corresponding IDU’s “DISTURB” field to either 21 for a low severity fire, or 23 for stand replacing fire. At the beginning of the next time step, we would like to have those DISTURB codes flipped to their negative values. That allows us to track (and see) in the IDU coverage not just current fires, but also the historical legacy of fire for a given IDU at a given time.

Finally, we need a process that increments the Time Since Disturbance field (TSD) at each time step whenever an IDU is in a post-disturbance state. We will use another standard Envision plug-in, *Trigger*. *Trigger* is in some ways similar to *Sync*, in that it updates IDU fields based on some specified set of conditions being met, but is significantly more flexible than *Sync*. It is also more computationally expensive. The input file for *Trigger* for this project is “trigger.xml” in the project directory. It specifies that TSD values should be updated at the end of each time step by adding one to the prior value

whenever the DISTURB code is less than 0 (remember, this will be true whenever the IDU is in a post-disturbance state).

The timing for which Envision executes these process is critical. Envision can run processes in series or in parallel. By default, processes execute in series based on two things. Within the **<autonomous_model>** entries in the project file, there is an attribute named 'timing'. This controls when in the Envision execution pipeline the process runs. It will have a value of either '0' or '1'. '0' indicates that the process should run *before* actor decision-making is invoked, allowing any process that set up the IDU's field information relevant for policies to run before the Actor(s) start making policy choices. For timing values of '1', those processes will be run *after* Actor decision-making is invoked, allowing for any processes that needs to respond to a policy decision *in that timestep* to run with appropriately updated IDU information.

Within each of these two periods in the execution pipeline, the order of plug-in execution follows the order specified in the Project (.envx) file.

After examining the various input files for this project, load the project in Envision. At this point, we will examine another capability of Envision plug-ins – the ability for the plug-in to “expose” a user interface (UI) to allow creating and modifying plug-in input files. While not all plug-ins expose a user interface, many of the “standard” Envision plug-ins do. In this project, Target and Spatial Allocator both expose an interface. To access these, go to the **Run** tab shown below and select the **Setup Models** option. You will see a list of all plug-in that expose a user interface. Select the plug-in of interest, and the GUI will be presented. Generally, the UI's will closely mirror their corresponding XML input files in terms of information presented; however, this depends on what the designer of the plug-in choses to expose.



The UI's for Trigger and Spatial Allocator are shown below. Both allow you to edit and save the inputs for their respective plug-ins. Generally, after saving a new set of inputs, you will need to shut down and restart Envision for the new inputs to take effect.

Target Setup

Basic Setup

Growth Rate: Linear Rate: 0.025 Percent

Density Field: POPDENS

OK

Availability Field: POP_AVAIL Capacity Field: POP_CAP % Availability Field: P_POP_AVAI

Cancel

Reports

Name	Query

Add Report Delete Report

Constants

Name	Value
peoplePerDU	2.3
kSQ	1.0
kCG	2.0
kUG	0.5

Add Constant Delete Constant

Allocations specify the capacity available at specific locations. Those locations are determined with a spatial query. Preference factors bias the capacities in particular circumstances.

Allocation Set: Status Quo Add Remove

Allocation Set ID: 0

Allocations

Name	Query	Value
Residential High Density	ZONE=10	kSQ*20*peoplePerDU
Agriculture	ZONE=54 or ZONE_B=...	kSQ*0.5*peoplePerDU
No Zone/Agricultural	ZONE=1 and LULC_B=...	kSQ*0.5*peoplePerDU
No Zone/Developed-Low	ZONE=1 and LULC_B=...	kSQ*4*peoplePerDU
No Zone/Developed-Med	ZONE=1 and LULC_B=...	kSQ*9*peoplePerDU
No Zone/Developed-High	ZONE=1 and LULC_B=...	kSQ*16*peoplePerDU

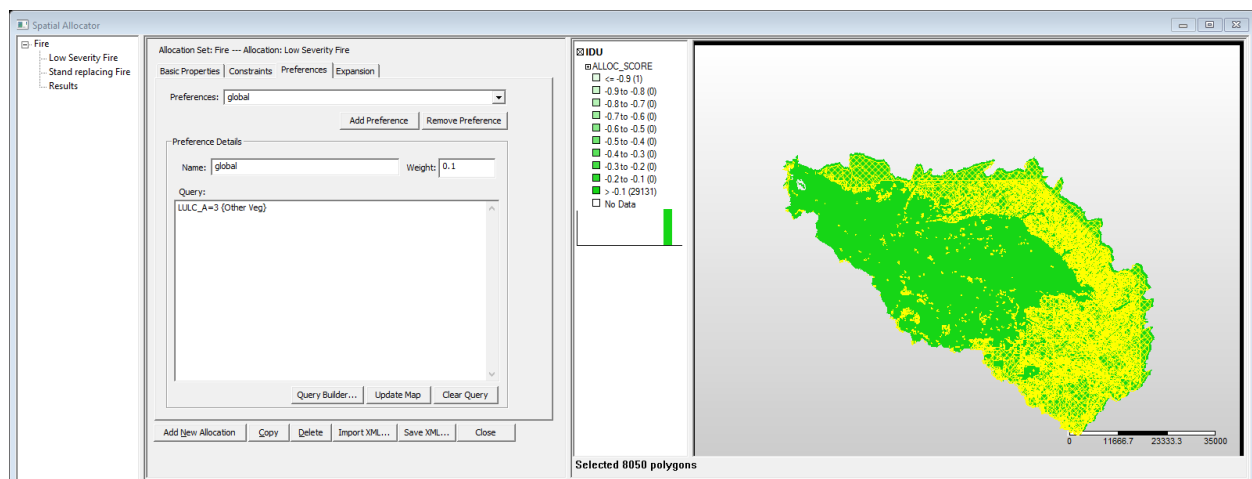
Add Allocation Delete Allocation

Preference Factors

Name	Query	Value

Add Preference Delete Preference

Trigger user interface shown above, Spatial Allocator user interface shown below.



Next, try running a simulation. Before starting the run, right-click on the main map and select “Fields” -> “Terrestrial System” -> “Disturbance”. This will make the DISTURB field the “active” IDU field during the run, allowing you to see, as the model runs, where fires are occurring.

Finally, we note that in all examples above, the models employed are extremely simplified. However, the process followed is generally quite similar for all Envision plug-ins and is something like the following: 1) identify an process need and appropriate plug-in, 2) add the plug-in to the Project file, either directly by editing the project .envx file, or using Envision Project Editor, 3) configure the plug-in with an appropriate input file. More complicated plug-ins often have more complicated inputs, but that is just the nature of models!

Practice Questions for Lesson 4.

1. After running a simulation, how much area experienced stand-replacing fire in year 10? Low-severity fire? (Hint: look at the reporter outputs – an output reporting fire areas was included in the Reporter.xml file used in this project. Examine this file and identify this entry)
2. How would you add additional disturbances into the simulation? (Hint: look at the top of the SpatialAllocator.xml file used in this project – it lists a series of “standard” disturbance codes are defined there.)
3. How would you include a salvage harvest policy that allowed clear cuts on forested stands in the 40-80 year old class that have experienced fire with the last five years?

Glossary

Actors (agents) – entities that make decisions about the management of particular portions of the landscape for which they have management authority

IDUs (Independent Decision Units) – Attributed polygons that represent a portion of the landscape. IDUs are the basic processing units for Envision. The attributes within an IDU are homogenous – for example, all of the area within an IDU is a specific land cover type.

Landscape - a set of polygon-based geographic information system maps and associated information containing spatially explicit depictions of landscape attributes and patterns

Multiagent Decision-making - the process of selecting policies and generating land management decisions affecting the landscape pattern

Policies - Fundamental descriptors of constraints and actions defining land use management decisionmaking; Policies capture rules, regulations, incentives and other strategies promulgated by public agencies in response to demands for ecological and social goods, as well as considerations used by private landowners/land managers to make land and water use decisions.

Response Indicators – stakeholder relevant metrics for evaluating landscape change

Scenario - A primary mechanism for capturing a set of bundled concepts for managing water, growth, land use, etc. They consist of coherent collections of “policies”, model settings, assumptions and preferences related to future change. Envision uses these to assess future trajectories of landscape productions e.g. “things people care about”. They are a primary mechanism for capturing stakeholder input.