FE / FOR 459

Forest Management Planning And Design

Bogdan Strimbu

Assignment #1: Forest Plan Essay

Introduction

The objective of the first course in the Capstone series is to develop a strategic and tactical forest management plan. This first assignment will evaluate an example of an actual management plan in North America. The example will provide context for the Capstone project that you will be completing during this course.

Learning Objectives

- Introduce the resource management plan.
- Understand the focus, content, and presentation of a plan.
- Provide context and reference for developing the group project plan.

Tasks

- Write a critique in the form of an essay for one management plan from ***Forest Plans of North America*** (Siry et al. 2015)
- Select one chapter (an individual plan) from the folder located in the FERN. Record your name and chapter number on the Excel sheet linked to this assignment.

Essay Content

The essay should contain *your own* perspective on the forest plan that you have selected. Your view on the plan should include the following:

- A summary description of the area for which the plan was executed
- The objective(s) of the plan
- The constraints of the plan
- The limitations of the plan
- What did you like about the plan?
- What did you dislike about the plan?
- What would you do differently if you could re-do the plan?
- References

In addition, the essay should relay your main findings about the plan in a few sentences. The essay should be formally written and structured with a tone and vocabulary for a professional audience. The essay should be at least 1 ½ pages but no longer than 2 pages, excluding the references.

All of the sources used in the essay should be acknowledged inside the narrative and detailed in the References section. There is no strict requirement on the style or format used, as long as it is consistent. Where applicable, the author(s) name(s), journal/book title, chapter/article title, year of publication, volume/chapter number, and the article number or pagination must be included. In-text sources should be identified using the author's last name followed by the publication year (e.g., Garman 2004). In the case that two authors are present then the last name of both authors is mentioned followed by the year of publication (e.g., Innes and Hickey 2006). For three or more authors, only the last name of the first author is written, followed by

”et al.” and the year of publication (e.g., Dorren et al. 2004). The cited literature should be presented in-text immediately after the argument, and the full citation should reflect the following templates, depending on the type of referred work. It is common practice for citations to include a hanging indent.

Book:

Houghton, J.T., G.J. Jenkins, and J.J. Ephraums. 1990. *Climate change: The IPCC scientific assessment*. Cambridge University Press, Cambridge, United Kingdom. 365p.

Chapter in Book:

Brokaw, N.V.L. 1982. Treefalls: Frequency timing and consequences. P. 101-108 in *The ecology of a tropical forest: Seasonal rhythms and long term changes*, Leigh, E.G., Jr., A.S. Rand, and D.M. Windsor (eds.). Smithsonian Institution Press, Washington, DC.

Dataset:

Putney, J.D., 2019. *Douglas-fir Growth Response to Operational Nitrogen Fertilization in Western Oregon* [Dataset]. Oregon State University, Corvallis, OR. Last Accessed: 27 Feb. 2023.

Journal Article:

Jurgensen, M.F., J. Johnson, M.A. Wise, C.S. Williams, and R. Wilson. 1997. Impacts of timber harvesting on soil organic matter, nitrogen, productivity, and health of Inland Northwest forests. For. Sci. 43(2):234-251.

Proceedings:

Blake, J.I., G.L. Somers, and G.A. Ruark. 1990. Perspectives on process modeling of forest growth responses to environmental stress. P. 9-20 in *Proc. of conf. on Process modeling of forest growth responses to environmental stress*, Dixon, R.K. (ed.). Timber Press, Portland, OR.

Technical Report:

Mason, R.R., and H.G. Paul. 1994. *Monitoring larval populations of the Douglas-fir tussock moth and western spruce budworm on permanent plots: Sampling methods and statistical properties of data*. USDA For. Serv. Gen. Tech. Rep. PNW-GTR-333. 22p.

Thesis/Dissertation:

Korol, R.L. 1985. The soil and water regime of uneven-age interior Douglas-fir (*Pseudotsuga menziesii* var. *glauca*). M.Sc. Thesis, Univ. of British Columbia, Vancouver, B.C., Canada. 164p.

Web Publications:

USDA Forest Service. 2002. *The process predicament: How statutory, regulatory, and administrative factors affect national forest management*. Available online at www.fs.fed.us/publications.html; Last Accessed: Apr. 15, 2005.

Submission

Prepare essays in a word document, and once completed, convert to a .pdf file. Name the file **A1_YourLastName.pdf** and upload it to the Assignment #1 page on Canvas.

Assignment 2: West Oregon District GIS

Introduction

Geographic Information Systems (GIS) are a crucial tool for resource planning because they are an efficient way to collect, organize, manipulate, interpret, and display geographic information. GIS software, such as ArcGIS, is a powerful approach to represent and communicate geographic and spatial data and outputs from analyses.

This week's lab begins preparation of the spatial and non-spatial input data from the Oregon Department of Forestry (ODF) West Oregon District needed for forest management planning on the District. The data in this lab includes information needed to run the Woodstock Optimization Suite later in this course and much of the information needed to complete project management plan introductions.[1]

Learning Objectives

- Prepare a shapefile of the ODF West Oregon District which supports the next assignment and provides a starting point for Woodstock landscape theme development. This file will break ODF stands into polygons annotated with the forest inventory, riparian, marbled murrelet, ownership, and county attributes needed to create a strategic forest plan with Woodstock.
- Review primary concepts of GIS and essential functions of ArcGIS Pro or ArcMap.

Files

This lab requires approximately 1 GB of storage space, about 800 MB for ODF's geodatabase and other files along with 40–200 GB for working files. Because Z: (\\onid-fs) has a 10 GB limit, consider creating a working folder for the lab on OneDrive to take advantage of OneDrive's 5 TB limit. For example, your working folder for the lab might be C:\Users\<your onid>\OneDrive - Oregon State University\FEFOR 459\Assignment 2.[2] Putting the working folder on a USB drive is also an option. However, be careful not to remove the USB drive while ArcGIS Pro, ArcMap, or ArcCatalog is running as doing so can corrupt the default geodatabase or other components of the lab. Separating files in different locations, such as placing some on OneDrive and some on Z:, is not recommended due to the number of ArcMap issues which appear to result.

Files for this lab are in

- Canvas: Files > ODF > GIS
- T:\Teach\Classes\FE459_FOR459\2024\ODF\GIS

Always copy files to your own location. Do not alter or work with these files in their original location. The files are

[1] In its 2018 and 2019 releases, Woodstock Optimization Suite was named Remsoft Spatial Planning System (RSPS). It's commonly referred to as just Woodstock and is sometimes still called RSPS.

[2] If files are saved elsewhere on the C: drive they are not on OneDrive and therefore won't sync between lab machines. They can, however, often still be temporarily accessed if you go back to the same lab machine.

- ODF_GISData_2023.gdb, which is compressed as a .zip. Spatial and non-spatial information for the West Oregon District is in this geodatabase.
- ODF_SLI_Tables_metadata.xlsx contains the definitions and descriptions for each field associated with the geodatabase.
- OR_Counties.zip, which contains a shapefile with the boundaries of all Oregon counties.

ODF's geodatabase has six spatial layers and six data tables. The ones most relevant to FE/FOR 459 are used in this lab. Since the spatial layers follow the OGIC (Oregon Geospatial Information Council) Oregon coordinate reference system standard used by state agencies they are in the NAD83 / Oregon Lambert (ft) coordinate system (EPSG:2992).

Tasks

Read through the following descriptions. Complete the associated tasks and questions related to GIS analysis of ODF's West Oregon District and connecting Woodstock with the District's spatial data. While this lab can be performed in most GIS tools, it's written for ArcGIS Pro and ArcMap. A supplementary section at the end of this document compares both variants of Arc and also QGIS. Task 4.4 covers retention of assignment files so, if you need stop partway through to start again later, checking Task 4.4 is suggested.

Task 1: Create an ArcGIS Pro or ArcMap map with West Oregon District layers

Task 1.1: Map setup

An ArcGIS (or QGIS) project or ArcMap map needs to be created with four starting layers from ODF's geodatabase: the two polyline layers for streams (*Hydrography_Statewide_Streams_FP_20221213*) and roads (*Transportation Roads Statewide ODF RT*) and the two polygon layers for stand (*Vegetation_SLIPoly*) and land ownership (*Ownership_ManagedLands_20230109*).

1. Start your choice of ArcGIS Pro or ArcMap. Create a new blank map with its default geodatabase set to be in the working folder you've chosen. Use File > Save or the save button to save the ArcGIS Pro project (the .aprx) or ArcMap map (.mxd) in the working folder as well.
2. Copy the ODF geodatabase and counties .zip files from Canvas or the T: drive. Unzip them to your working folder.
3. Add the streams, ownership, roads, and stand (*Vegetation_SLIPoly*) layers from the ODF geodatabase (.gdb) to the project. Since the coordinate system for these layers is EPSG:2992, adding them defaults Arc's data frame to EPGS:2992.[3] If you need to do so in order to access the ODF geodatabase, add a folder connection to your working folder (Insert > Add Folder in ArcGIS Pro, Catalog Tree > Folder Connections > right click > Connect to folder… in ArcCatalog if using ArcMap).
4. Optional: since the statewide roads and streams layers are slow to draw, uncheck them in the Table of Contents so that they're not displayed.

[3] The most current Oregon statewide coordinate systems are EPSG:6556 (m) and EPSG:6557 (ft), which use NAD83(2011) rather than NAD83(HARN). Adjustments between HARN and 2011 are mostly in the centimeter to millimeter range (NOAA NOS NGS 65) and thus unimportant to this course.

Task 1.2: Create ODF West Oregon District stand, boundary, and ownership layers

While ODF's West Oregon District is the capstone project planning area, the layers in ODF's geodatabase are statewide. It's easier to work with files just for the District, particularly since the streams and roads layers (*Hydrography_Statewide_Streams_FP_20221213* and *Transportation Roads Statewide ODF RT*) are large and slow to work with.

To start, create a layer with just the stands in the West Oregon District.

1. From the ribbon at the top of ArcGIS Pro choose Map → Select By Attributes. In ArcMap this is Selection → Select By Attributes in the menu. Choose *Vegetation_SLIPoly* as the layer (input rows) and in
    a. ArcGIS Pro: leave new selection as the selection type, select DISTRICT as the field, leave is equal as the operator, and select West Oregon in the dropdown.
    b. ArcMap: double click DISTRICT, and use the equals button and Get Unique Values to set up the selection as DISTRICT = 'West Oregon'.
2. Press OK. The West Oregon District's stands should now be shown as selected in the map. Stands in other ODF districts should not be selected.
3. Right click on *Vegetation_SLIPoly* and choose Data → Export Features (ArcGIS Pro) or Data → Export Data (ArcMap) and change the Output Feature Class to ODF_Stands in the default geodatabase (in ArcMap, change just the last part of the path after the default geodatabase or use the folder button to the right of the path). Press OK and (in ArcMap) say Yes to adding the exported data as a layer.

Next, dissolve the stands layer to get a boundary for the District.

4. At the righthand side, choose Search (in ArcGIS Pro you may need to first change to the Geoprocessing tab at bottom right). Type dissolve and click on Dissolve (Data Management).
5. Choose ODF_Stands as the Input Features and change the Output Feature Class to ODF_Boundary. Click on Run (ArcGIS Pro) or OK (ArcMap). Wait for the dissolve to complete and the layer to be added to the project. The new layer will saved to your default .gdb.
6. Right click on *ODF_Boundary* → Zoom to Layer.
7. If you are using ArcMap, remember to save the map file (.mxd) periodically using the save button since it doesn't autosave.

Task 1.3: Create layers for the ODF West Oregon District's streams and roads

1. Choose Select by Location and *Hydrography_Statewide_Streams_FP* as the input features (ArcGIS Pro) or target layer (ArcMap).
2. Select *ODF_Boundary* as the selecting features (ArcGIS Pro) or source layer (ArcMap).
3. Leave Intersect as the relationship (ArcGIS Pro) or "intersect the source layer feature" (ArcMap) as the spatial selection method.
4. Press OK. The streams within the West Oregon District's stands should show as selected (turn display of the streams layer back on if you need to).

5. Right click on *Hydrography_Statewide_Streams_FP* → Data → Export Features (ArcGIS Pro) or Export Data (ArcMap). Export the selected features to ODF_Streams in the map's default geodatabase and add the layer.
6. Repeat steps 1–5 with *Transportation Roads Statewide ODF RP* to create an *ODF_Roads* layer.
7. Repeat steps 1–5 with *Ownership_ManagedLands* to create an *ODF_Ownership* layer.

Both *ODF_Streams* and *ODF_Roads* retain a Shape_Length field from the ODF layers they were created from. If these layers have been saved in a geodatabase ArcMap will update these fields automatically. If they're saved in other formats it may be necessary to recalculate road and stream lengths by 1) right clicking the layer → Edit Features → Start Editing, 2) right clicking the layer → Open Attribute Table → Shape_Length → right click field header → Calculate Geometry, 3) Editor → Save Edits and Editor → Stop Editing.

- If Shape_Length is being automatically updated by ArcMap Calculate Geometry will stay greyed out when the layer is open for edit.
- If the layer is saved as a shapefile, Shape_Length will be truncated to Shape_Leng.

***Q1** Produce screenshots showing your roads, streams, boundary, and stands layers.*

Task 1.4: Simplify the ownership layer

Since only the ownership information in *ODF_Ownership*'s TYPE field is needed its other fields can be removed to simplify the final shapefile which will be presented to Woodstock.

1. Right click *ODF_Ownership* and choose Attribute Table (ArcGIS Pro) or Open Attribute Table (ArcMap).
2. Skipping TYPE, right click on each field's header to delete it. The OBJECTID *, GlobalID *, Shape *, Shape_Length, and Shape_Area are internally maintained ArcMap fields which cannot be deleted. That's OK. You can use ctrl+click to select multiple fields to delete.
3. Right click TYPE's field header, choose Fields (ArcGIS Pro) or Properties (ArcMap), and change the alias of the TYPE field to Owner. Close the tab and save the change (ArcGIS Pro) or press OK (ArcMap).
4. Since *ODF_Ownership*'s Owner field has values containing spaces, such as start an edit session (ArcMap), open the attribute table, and use Find and Replace (in the dropdown menu at the upper right (ArcGIS Pro) or left (ArcMap) of the attribute table—select the Owner field first and ensure Search only selected fields is checked) remove the spaces. Save the changes (Edit → Save in ArcGIS Pro, Editor → Save in ArcMap) and end the edit session (ArcMap). This isn't important now but it avoids problems with Woodstock landscape themes later in the course.
    a. In ArcGIS Pro, find and replace is restricted by the field's coded value domain. To allow free editing of the Owner field, first open the fields tab, select blank in the Domain dropdown for owner, close the tab, and save the change.
    b. Typical changes are "Board of Forestry Lands" to "BOF" and "Admin Site" to "Admin".
5. Color *ODF_Ownership* by county.
    a. ArcPro: Right click *ODF_Ownership* → Symbology.

b. ArcMap: Right click *ODF_StandsCounty* → Properties → Symbology tab → Show: Categories
c. Select Unique values and then "Owner" in the Value Field drop-down.
d. Click "Add All Values" (in ArcGIS Pro this is the button).
e. Pick a color scheme from the color ramp drop-down.
f. Click Apply to view the symbology change, then OK when you're satisfied.

Q2 *Produce a good quality screenshot clearly showing your ownership layer.*
Q3 *Create a table listing how many features are in each of these five West Oregon District layers (stands, streams, roads, boundary, and ownership).*

Task 2: Create layers for riparian buffers, murrelets, ownership, and counties

Depending on the type of data and your objective, there are a number of spatial manipulations that may be necessary. Following are several key operations on vector data that will be useful for starting your capstone project.

Task 2.1: Trim roads and streams layers to stands

In some applications it is useful to know certain attributes, such as size, that are common to difference processes or properties. For example, the length of the roads within a managed area or the allocation of stands within certain Counties. These spatial analyses can be completed using the intersection tool. This operation can be executed between a line feature and a polygon feature, as well as between two polygons. It is useful to have the data for all streams and roads associated with the West Orgon District, but it is useful to also have layers with only the stream and road segments fully enclosed within the District's stands.

1. Use search → intersect to access the Intersect (Analysis) tool. Add the *ODF_Roads* and *ODF_Boundary* layers as Input Features.
2. Change the Output Feature Class from ODF_Roads_Intersect to ODF_Roads_IN.
3. Click Run or OK and wait for the intersection to complete. The roads in *ODF_Roads_IN* should stop at the stands' edges.
4. Repeat 1–3 with *ODF_Streams* to create *ODF_Streams_IN*.

If needed, recalculate the geometry of road and stream lengths before answering all three parts of Question 5.

Q4 *Produce a snapshot displaying* ODF_Roads_IN *and* ODF_Streams_IN.
Q5 *Create a table of the road and stream lengths in* ODF_Roads, ODF_Roads_IN, ODF_Streams, *and* ODF_Streams_IN *using reasonable units and precision. Why do you think the lengths differ between* ODF_Roads *and* ODF_Roads_IN *and between* ODF_Streams *and* ODF_Streams_IN*?*

Task 2.2: Set streams' riparian buffer widths and create a riparian buffer layer

The Oregon Forest Practices Act (OFPA) requires stream protection buffers whose width varies depending on the size of the stream, whether fish are present, and distance from salmon, steelhead, and bull trout (SSBT) streams. ODF typically implements wider buffers than required by OFPA but the ODF GIS data available for FE/FOR 459 in Winter 2024 has not been updated with the SSBT, seasonality, or reach length reclassification needed to implement the OFPA rules effective January 1st, 2024. The steps below therefore use an approximation of the legally required buffer widths (Table 1).

In GIS, variable-width riparian buffers can be created by supplying widths to ArcMap's buffer tool using a field that's added to *ODF_Streams_IN*.

1. Open *ODF_Streams_IN*'s attribute table. In ArcMap, click Edit Features → Start Editing
2. Add BufferWidth as a new field (the Add button in ArcGIS Pro, Add Field in ArcMap's upper left dropdown). Since buffer widths are an integer number of feet and the layer's coordinate system (EPSG:2992) has units of feet, any numeric data type (short or long integer, float, or double) is fine.
3. Use a small bit of Python code to set *BufferWidth* to the values in Table 1 (how Python works for field calculations is discussed in more detail in the supporting materials for the FVS lab). Right click on BufferWidth's field header and choose Calculate Field (ArcGIS Pro) or Field Calculator (ArcMap).
   a. Leave the Expression Type as Python 3 (ArcGIS Pro) or change Parser to Python and check Show Codeblock (ArcMap).
   b. In the Code Block (ArcGIS Pro) or Pre-Logic Script Code box (ArcMap) paste
```
def getBufferWidth(fishpres, fpasize):
  if fishpres == "Fish":
    return 115
  elif fpasize == "Large":
    return 75
  elif fpasize == "Medium":
    return 75
  elif fpasize == "Small":
    return 75
  else:
    return 50
```
   c. In the BufferWidth = box paste
```
getBufferWidth(!Fishpres!, !Fpasize!)
```
   d. Click OK. BufferWidth should be filled with only the values from Table 1 and have no blanks.

Table 1: Conservative defaults for riparian buffer width requirements from the Oregon Forest Practices Act (OFPA) and Oregon Department of Forestry (ODF) policies for standard practice vegetation retention in western Oregon, as available from current ODF GIS data.

	ODF riparian buffer width, ft		
stream size	**fish streams (F, SSBT)**	**nonfish streams (N, Np, Ns)**	**fish unknown**
large	115	75	75
medium	115	75	75
small	115	75 for 500–600 ft	75
unknown	(no such streams)	50	50

Nonfish stream buffer widths can drop to 50 feet for 500–650 feet upstream from an SSBT stream and then to 20 foot at 1150 feet upstream. See [ORS 629-643-0100(1) Table 1](). ODF's minimum buffer width is 35 feet, however.

Now that buffer widths are established, buffer all streams within the ODF West Oregon District using the buffer tool.

4. Search for buffer and choose Buffer (Analysis) or go to Catalog → Analysis Tools → Proximity → Buffer.
5. Select *ODF_Streams_IN* as the Input Feature and set the Output Feature Class to ODF_Streams_Buffer in the default geodatabase.
6. For Distance select Field (ArcGIS Pro) or check Field (ArcMap) and then select BufferWidth as the field.
7. Side Type should be Full. Dissolve Type should be Dissolve all (ArcGIS Pro) or All (ArcMap) to combine all output features into a single feature.
8. Click OK. When buffering completes you should see a single buffer feature that's narrower around smaller streams and wider around larger ones. (You can also check the widths for correctness using the measure tool.)

The buffer will include areas outside of the West Oregon District boundary, so clip it to the boundary:

9. Search for clip and choose Clip (Analysis) or go to Analysis Toolbox → Extract → Clip.
10. Select *ODF_Streams_Buffer* as the Input Feature and *ODF_Boundary* as the Clip Feature.
11. Set the output features to ODF_Streams_Buffer_IN in the default geodatabase.
12. Click OK. When clipping completes the stream buffers should end at the edges of the stand polygons.

Since it will be necessary to distinguish riparian areas from upland areas later, a field needs to be added to *ODF_Steams_Buffer_IN* to track this.

13. Add SlopePosition to *ODF_Steams_Buffer_IN* as a text field.
14. Start editing *ODF_Steams_Buff_IN* (ArcMap) and enter "riparian" for the SlopePosition field.
15. Save the edit (ArcGIS Pro and ArcMap) and stop editing (ArcMap).

Q6 *Produce a snapshot of* ODF_Streams_Buffer_IN *displaying the riparian buffers. Choose the area displayed such that both the landscape context of the streams and the variation in buffer width is clear.*

Task 2.3: Extract marbled murrelet areas for the West Oregon District

In some situations, you may need to work with only portions of existing features, with the rest of the layer remaining unchanged. In these instances, the spatial information you are interested in can be separated using the clip tool. This information can then be added to existing features using the union tool. It will be important to have the marbled murrelet nesting site information included within the ODF stands data. To add this information, clip the marbled murrelet nesting areas to the West Oregon District.

1. Add the *Biology_MurreletManagementAreas* layer from the ODF .gdb
2. Search for clip or go to Analysis Toolbox → Extract → Clip.
3. Select *Biology_MurreletManagementAreas_20221213* as the input features.
4. Select *ODF_Boundary* as the clip features.
5. Change the Output Feature Class to ODF_MMMA in the default geodatabase and click OK.
6. Ensure the new *ODF_MMMA* layer includes only murrelet zones within the West Oregon District boundary.

It is always important to check the geometric accuracy of spatial files. In this case, there are polygons within *ODF_MMMA* that overlap because this feature includes past nesting sites that have been updated as well as buffers. As a simplifying assumption that all past, present, and planned murrelet areas should be managed as murrelet areas, dissolve all the *ODF_MMMA* polygons into one feature.

7. Search for dissolve or go to Data Management Tools → Generalization → Dissolve.
8. Select *ODF_MMMA* as the Input Feature.
9. Change the output feature class to ODF_MMMA_dissolve in the default geodatabase and click OK.

Similar to riparian areas, add a field indicate marbled murrelet restrictions.

10. Open the *ODF_MMMA_dissolve* attribute table and add Wildlife as a text field.
11. Edit *ODF_MMMA_dissolve*, enter "murrelets" for Wildlife. Save the edits and stop editing.

Q7 *Produce a screenshot of the murrelet areas on the District along with Wildlife = murrelets in the* ODF_MMMA_dissolve *attribute table.*

Task 2.4: Create a simplified county layer in the data frame's coordinate system

1. Add the orcntypoly.shp shapefile to the map.

The West Oregon District is in Lincoln, Polk, and Benton Counties. Since only the altName field of *orcntypoly* is needed later the layer needs to be copied and its attribute table reduced. This is also a convenient opportunity to convert the counties from EPSG:4269 into the map data frame's coordinate system.

2. Search for project and choose Project (Data Management).
3. Choose *orcntypoly* as the input dataset and change the output feature class to Oregon_counties in the default geodatabase.
4. Click the globe icon (ArcGIS Pro) or hand icon (ArcMap) to the right of Output Coordinate System. Search for 2992, select NAD 1983 Oregon Statewide Lambert (Intl Feet) (ArcMap calls it NAD_1983_Oregon_Statewide_Lambert_Feet_Intl), and click OK.
5. Click Run (ArcGIS Pro) or OK (ArcMap) to reproject the county layer to EPSG:2992.
6. Open *Oregon_counties*' attribute table and
    a. ArcGIS Pro: choose Fields View from the dropdown at upper right and select everything which is not a read only field, geometry field, or the altName field by clicking on the boxes at the far left of the field rows. Right click, again on the field headers, and choose delete.
    b. ArcMap: right click on each field's header to delete it.
7. Use the fields view in ArcGIS Pro to change the alias of the altName field to County. In ArcMap, right click the field header, choose properties, and change the alias to County.
8. Optional: *orcntypoly* is no longer needed and can be removed from the map. Removing it reduces ArcMap warnings caused by the mismatch between its coordinate system and the data frame's coordinate system.

Task 3: Intersect and union counties, ownership, murrelets, and riparian areas into stands

Recall from FE/FOR 457 that Woodstock schedules harvests on areas, that areas are identified by landscape themes, that an area's yields are determined by finding a yield table which matches its themes, and that management objectives and constraints are written in terms of themes and yields. To run Woodstock on the West Oregon District, Woodstock therefore needs to know all of the areas on the district, how big they are, and what their themes are. This task produces a layer containing a basic set of themes—county, land ownership, marbled murrelet management areas, and riparian buffers—which will be extended with more information in the next task in order to support creation of additional themes in later assignments.

The intersection and union tools are useful here because they preserve all the attribute fields within their input and output files, allowing information from the other layers prepared in the previous task to be added to ODF's stand layer. An intersection retains only the areas which are present in both layers while a union adds the areas from both layers together. Since the West Oregon District's stands are embedded in counties, an intersection reduces the counties layer to just the stands. Unions are used for murrelet areas and riparian buffers as these three layers have been clipped to the stand boundary. The order in which counties, ownership, murrelet management areas, and riparian buffers are combined with stands is arbitrary but, since some capstone groups may consider more complex murrelet management options or adjust riparian buffer widths, those two are brought in after counties and ownership. This simplifies the process when those groups redo the unions with modified murrelet or riparian layers.

Task 3.1: Intersect counties

Because ODF has obligations to transfer a portion of its timber harvest income to counties, harvest revenue needs to be tracked on a per county basis. Using the intersection tool, create a stand layer which includes the counties within which the ODF West Oregon District is located.

1. Search for intersect and choose the Intersect (Analysis) tool.
2. Enter *Oregon_counties* and *ODF_Stands* as the input features.
3. Change the Output Feature Class to ODF_StandsCounty in the default geodatabase and click OK.
4. Color *ODF_StandsCounty* by county using the same classification by unique values process as in Task 1.4. The unique values for the county names (in the altName field) should be Benton, Lincoln, and Polk.

***Q8** Produce a snapshot showing the ODF stands colored according to the county in which they are located in.*

Task 3.2: Intersect ownership

ODF also needs to track whether timber harvest income should be directed to counties or the Common School Fund. Lands owned by the Board of Forestry provide county revenue and lands owned by the Fund provide school revenue .

1. Start the Intersect (Analysis) tool again and enter *ODF_Ownership* and *ODF_StandsCounty* as the input features.
2. Set the Output Feature Class to *ODF_StandsCountyOwner* in the default geodatabase.
3. Press Save and then OK to execute the union.

4. Color *ODF_StandsCountyOwner* by owner by changing its symbology to Owner categories.

***Q9** Produce a screenshot of ownership distribution in* ODF_StandsCountyOwner*.*

Task 3.3: Union murrelet management areas

The marbled murrelet layer can now be added to *ODF_StandsCountyOwner* using the union tool. The resulting layer will then contain the ODF stand information, county information, and marbled murrelet areas. For non-murrelet areas the Wildlife field will be changed from blank to NoListedSpecies. The assumption behind this is no species listed as threatened, endangered, sensitive, or an Oregon conservation strategy species makes sufficient use of areas marked as NoListedSpecies that they require management approaches distinct from managing for the many unlisted wildlife species present also across the West Oregon District.

1. Search for union and choose the Union (Analysis) tool.
2. Enter *ODF_StandsCountyOwner* and *ODF_MMMA_dissolve* as Input Features.
3. Change the OutputFeature class to ODF_StandsCountyOwnerMurrelet in the default geodatabase and click OK.

To update the attribute table:

4. Start editing *ODF_StandsCountyOwnerMurrelet* and open the attribute table.
5. Right click the Wildlife field header and choose Calculate Field (ArcGIS Pro) or Field Calculator (ArcMap). In ArcMap, also change the parser to Python.
6. Enter the one line
   ```
   !Wildlife! if !Wildlife! else "noListedSpecies"
   ```
   in the box below Wildife = .
7. Press OK. All of the blank cells should change to noListedSpecies. The Murrelets cells should remain unchanged.
8. If everything looks good close the attribute table, save edits, and (in ArcMap) stop editing.
9. Display murrelet and non-murrelet areas in different colors using the same unique values symbology as in previous tasks.

***Q10** Produce a snapshot showing areas colored according to whether they have marbled murrelet restrictions.*

Task 3.4: Repair polygon intersections

One of the most common errors with Check Geometry is self-intersection which identifies the polygons that intersect themselves. For polygon layers, self-intersection indicates a polygon is not well formed as it crosses over itself. In some cases, the geometry can be repaired using the Repair Geometry tool.

1. Search for check geometry and choose Check Geometry (Data Processing).
2. Select *ODF_StandsCountyOwnerMurrelet* as the Input Feature and click OK.
3. Review the resulting warnings.

Q11 *Run the Check Geometry tool on* ODF_StandsCountyOwnerMurrelet *and produce a snapshot of the results.*

4. Search for repair geometry and select Repair Geometry (Data Management).
5. Select *ODF_StandsCountyOwnerMurrelet* as for Input Features and click OK.[4]
6. Once the repair completes, re-check the geometry using Check Geometry again.
7. Confirm the geometry issues have been resolved.

Q12 *Produce a screenshot of the check geometry output showing the errors in* ODF_StandsCountyOwnerMurrelet *have been removed.*

Task 3.5: Union riparian buffers into stands and update stand areas

As with marbled murrelet management areas, Woodstock needs to be aware of where riparian buffers are so that it can be told not to clearcut them. A second union further splits up ODF's stand polygons to provide this information. Both a desirable feature and a limitation of this approach is riparian areas receive the ID of the stand they're part of. This means the OneLiner table join performed in Task 4.1 most likely assigns ODF inventory data from upland cruise plots to riparian areas. However, correcting this riparian bias is beyond the scope of this course.[5]

1. Open the union tool and select *ODF_StandsCountyOwnerMurrelet* and *ODF_Streams_Buffer_IN* as the Input Features.
2. Change the output feature to ODF_StandsCountyOwnerMurreletRiparian in the default geodatabase.
3. After executing the union, you should see the stand polygons are now split into upland and riparian buffer sections. Check that 1) the Wildlife field from is showing noListedSpecies or murrelet and 2) the SlopePosition field is showing blank or riparian values. If either field is missing or looks incorrect, check the two input layers.
4. Start editing *ODF_StandsCountyOwnerMurreletRiparian* and use the Field Calculator change empty values in SlopePosition to Upland. The Python one liner for the box under SlopePosition = is
   ```
   !SlopePosition! if !SlopePosition! else "upland"
   ```
   Similar to the Wildlife field, riparian entries in SlopePosition should remain unchanged and the blanks should change to upland.
5. Once satisfied, save the edits and (in ArcMap) stop editing.

Task 3.6: Merge slivers into larger polygons

When intersections or unions are executed, processing artifacts can result. These are often features with very small sizes, which are commonly called slivers as they are frequently small and narrow. Most

[4] If *ODF_StandsCountyOwnerMurrelet* is stored in a shapefile which hasn't been spatially indexed using the Add Spatial Index (Data Management) tool will give a warning about the lack of a spatial index is expected.
[5] It's likely the longer a riparian buffer has been boundaried out of surrounding upland clearcuts the greater its divergence from upland inventory and thus the larger this inventory bias becomes. Possibly the most efficient bias reduction option that's currently operational would be an enhanced forest inventory (EFI) developing a network of ground cruise plots from DOGAMI's aerial LiDAR data within no harvest streambank zones, inner RMAs, and outer RMAs. The plots would then be cruised and the ground cruise data integrated with a LiDAR area based approach (ABA) to estimate riparian inventory from the point clouds' statistics.

commonly, slivers occur because feature boundaries don't quite align between layers. They can also be processing artifacts or legitimate fragments of a stand. There are many methods to remove slivers, but a simple iterative procedure uses the Eliminate function from the Data Management toolbox. Since eliminate operates by merging the features selected in a layer into adjacent, larger ones a method is needed to select slivers without also selecting other small and narrow but important areas, such as riparian buffers for type N streams. ACRES or other area based fields can be used to find small polygons and various indicators can be used to identify polygons with high aspect ratios. In this case a shape index will be used (Demetriou et al. 2013, Eq. 1), which has a minimum value of one for a circle and increases as a polygon becomes longer and skinnier.

1. Update *ODF_StandsCountyOwnerMurreletRiparian*'s ACRES field by right clicking its header in the attribute table and selecting Calculate Geometry. Set the Property and Area Unit (ArcGIS Pro, US Survey Acres) or Units (ArcMap) and click OK.
2. Open *ODF_StandsCountyOwnerMurreletRiparian*'s attribute table and add ShapeIndex as a floating point field (float or double). Start the Field Calculator and calculate the ShapeIndex as
    a. ArcGIS Pro: `!Shape_Length! / (2 * math.sqrt(3.1415 * !Shape_Area!))`
    b. ArcMap: `[Shape_Length] / (2 * Sqr(3.1415 * [Shape_Area]))`
3. Use Select by Attribute and `((ShapeIndex > 3.1) AND (ACRES < 0.8)) OR (ACRES < 0.02)` to select *ODF_StandsCountyOwnerMurreletRiparian* features which are 1) small enough to be less than 0.8 acres in size with aspect ratios high enough their shape index exceeds 3.1 or 2) too small to be worth tracking individually.
    a. In ArcGISP Pro, use Map → Select by Attribute on the ribbon and enable the SQL toggle.
    b. In ArcMap, choose Select by Attribute from the upper left dropdown menu.
4. Review the selected areas for reasonableness. Nearly all of them should be the sort of fragments which aren't meaningful to forest management. If you're dissatisfied, adjust the thresholds (3.1, 0.8, and 0.02) or, if it's quicker and easier, manually deselect polygons.
5. Search for eliminate and select Eliminate (Data Management).
    a. Select *ODF_StandsCountyOwnerMurreletRiparian* as the Input Feature.
    b. In ArcGIS Pro the Use the selected records toggle should default to on.
    c. Change the Output Feature Class to *ODF_StandsCountyOwnerMurreletRiparian_Sliv1* in the default geodatabase.
    d. Press Run (ArcGIS Pro) or OK (ArcMap) to run the eliminate tool.
6. Repeat steps 1–5 using *Sliv1*, *Sliv2*, and so on until the sliver selection finds no more polygons to eliminate. This process can get stuck if a small polygon has no adjacent polygon to merge it into. If that happens, just delete the polygon manually. Right clicking on the number (ArcGIS Pro) or the ▶ (ArcMap) at the far left of selected features in the attribute table and choosing Zoom To (ArcGIS Pro) or Zoom to Selected (ArcMap) can be helpful.
7. Save (or rename) the final layer as *ODF_StandsCountyOwnerMurreletRiparian_SlivFinal*.
8. Change the symbology and verify county, ownership, murrelet areas, and riparian buffers have remained substantially intact.

Q13 *Produce a screenshot of* ODF_StandsCountyOwnerMurreletRiparian_SlivFinal *showing your work. What are the total number of features within the initial* ODF_StandsCountyOwnerMurreletRiparian *layer compared to the* SlivFinal *version? How many slivers were eliminated?*

Task 4: Create a shapefile for Woodstock with planning polygons and forest inventory

In many instances, spatial decision making includes nonspatial information. In GIS, nonspatial information is commonly included by joining a table of non-spatial data with a spatial layer's attribute table. This join adds columns from the nonspatial table to the layer's attribute table by matching up the rows of the nonspatial table with the rows in the label attribute table.[6] In this case, ODF's stand IDs are used for the matching.

Task 4.1: Join ODF inventory and planning tables

This task joins the *Vegetation_SLI_ROOTS_OneLiner* and *Vegetation_SLI_ROOTS_Structure* tables in ODF's .gdb with *ODF_StandsCountyOwnerMurreletRiparian_SlivFinal*. *Vegetation_SLI_ROOTS_OneLiner* provides inventory data, such as species basal areas, DBH, or TPA. *Vegetation_SLI_Structure* table contains additional data on vegetation structure as well as ODF's desired future condition (DFC) for the stand. The DFC is critical information as it specifies stand level management objectives.

1. Add *Vegetation_SLI_ROOTS_OneLiner* and *Vegetation_SLI_Structure* from the ODF .gdb. (Be careful to get *SLI_Structure* and not the other structure table without a DFC field.)
2. Right-click *ODF_StandsCountyOwnerMurreletRiparian_SlivFinal* and choose Joins and Relates → Add Join (ArcGIS Pro) or Join… (ArcMap).
3. ArcMap: Ensure Join Attributes from a Table is selected.
4. Select STD_ID as the input field in *ODF_StandsCountyOwnerMurreletRiparian_SlivFinal* which the join will be based on. STD_ID is short for stand ID.
5. Select *Vegetation_SLI_ROOTS_OneLiner* as the table to join. ArcGIS Pro may populated by default.
6. Select std_id as the join field. std_id is also short for stand ID.
7. In this case, non-matching records aren't of interest, so the default behavior of keeping all records is not needed.
    a. ArcGIS Pro: Uncheck Keep All Target Features.
    b. ArcMap: Choose Keep Only Matching Records.
8. Click Validate Join to check whether the join will be successful.
9. If issues are reported, you should assess their importance and either execute the join anyway or adjust it such that the results would meet your expectations.
10. To execute the join, click OK. You should see the fields from *OneLiner* have been appended to *ODF_StandsCountyOwnerMurreletRiparian_SlivFinal*'s attribute table.
11. Repeat steps 1–10 to also join *Vegetation_SLI_Structure*. In *Vegetation_SLI_Structure* the stand ID field is STD_ID.

Once both joins are completed, open *ODF_StandsCountyOwnerMurreletRiparian_SlivFinal*'s attribute table and verify the fields from *SLI_ROOTS_OneLiner* and *Vegetation_SLI_Structure* have been appended to *SlivFinal*'s attribute table.

---

[6] Arc also has a relate option, which creates a link between the spatial layer and the table with the additional information.

Task 4.2: Create Woodstock compatible theme fields

Woodstock places two important requirements on shapefiles it imports. First, management areas must have an age of at least one planning period. This is incompatible with ODF's inventory practices, which include age zero. The simplest solution is to increase these areas' ages from zero to one. Second, landscape theme information must appear in fields named Theme1, Theme2, and so on. This means the county, owner, murrelet, and slope position fields need to be copied into matching theme fields. Themes for stand ID and desired future condition themes are also useful and silvicultural themes will be added in later assignments. Woodstock also **requires themes appear in their numbered order** in the shapefile's attribute table.

1. Create an integer age1 field (either short or long integers are fine) in *ODF_StandsCountyOwnerMurreletRiparian_SlivFinal* and use the field calculator to make age1 a copy of the existing age column with a minimum age of one rather than zero. The Python for this is
   ```
   def getAge1(age):
     if age == 0:
       return 1
     else:
       return age
     end
   ```
   Set the age1 = box to `getAge1(!Vegetation_SLI_Structure.AGE!)` and press OK to populate the age1 column from the AGE in *Vegetation_SLI_Structure*. AGE and age1 should be same except for age1 having ones where AGE has zeros.
2. Create Theme1, Theme2, …, Theme6 as text fields (ArcMap's default length of 50 characters is plenty).
3. Start editing *ODF_StandsCountyOwnerMurreletRiparian_SlivFinal* (ArcMap). Use the field calculator to fill Theme1 with a copy of the county field by picking the County (ArcGIS Pro) or altName (ArcMap) field from the Fields box. `!ODF_StandsCountyOwnerMurreletRiparian_SlivFinal.altName!` (ArcMap) will appear in the code box. Press OK to run the field calculator.
4. Repeat step 3 to fill
   a. Theme2 with ownership information from the Owner field (TYPE in ArcMap).
   b. Theme3 with murrelet information from Wildlife.
   c. Theme4 with riparian information from SlopePosition.
   d. Theme5 with stand IDs from STD_ID. This one differs from the others since 1) Woodstock requires theme attributes have some text as well as just numbers and 2) Python's `str()` function is needed to convert STD_ID from an integer to a text string. Therefore, enter `'stand' + str(!ODF_StandsCountyOwnerMurreletRiparian_SlivFinal.STD_ID!)` in the field calculator rather than just `!STD_ID!`.
   e. Theme6 with the desired future conditions from DFC.
5. When satisfied, save the changes and (ArcMap) end the edit session.

If you make a mistake and end up with out of order theme fields the Feature Class to Feature Class (Conversion) tool can be used to create a new layer with the theme fields permanently reordered.

Task 4.3: Export a shapefile for use by Woodstock

Since Woodstock does not read the ESRI file geodatabases (.gdb files) ArcMap (and ArcGIS Pro) use by default it is necessary to export *ODF_StandsCountyOwnerMurreletRiparian_SlivFinal* in a format which Woodstock does support. While increasingly perceived as a legacy file format, shapefiles remain in common use for exchange of vector GIS data between programs and are mutually supported by Arc and Woodstock. Hence this final task of the lab.

To export *ODF_StandsCountyOwnerMurreletRiparian_SlivFinal* as a shapefile

1. Right click the layer and choose Data → Export Features (ArcGIS Pro) or Data → Export Data (ArcMap).
2. Export all features in the layer's source data.
3. Click the folder icon to the right of Output feature class and navigate out of the map's default .gdb to the directory where you want to save the shape file. Enter ODF_FinalData.shp as the name and, in ArcMap, change Save as type to Shapefile.
4. Click Save (ArcMap) and then OK to export the shapefile.
5. Once export completes and *ODF_FinalData* is added to the map it should appear identical to *ODF_StandsCountyOwnerMurreletRiparian_SlivFinal*. Verify you can also locate the individual ODF_FinalData.{cpg, dbf, prj, sbn, sbx, shp, shx} files which form the shapefile within your working folder.

The intersections, unions, and joins accumulate several fields starting with FID_ and OBJECTID_ that don't need to be retained. There are also duplicate fields starting with DISTRICT_, STD_ID_, and AREA_RPT_ which don't need to be kept either. If you wish, these can be removed with the Delete Field (Data Management) tool.

Q14 *Produce a screenshot displaying a few of the joined fields and the Woodstock theme fields at the end of ODF_FinalData.shp's attribute table.*

Task 4.4: Ensure assignment files are retained for future use

The layers created in this assignment are reused throughout the rest of the course. It's difficult to do this if you've lost your work.

1. If you haven't been doing so, consider using the Catalog tab (ArcGIS Pro) or ArcCatalog (separate from ArcMap) to remove unneeded working or temporary layers from the map's default geodatabase. Layers need to be removed from ArcGIS or ArcMap's contents before they can be deleted from the catalog.
2. Quit ArcGIS Pro or ArcMap, saving any outstanding changes.
3. If using OneDrive, check that your working folder is synced (along with any other needed files) in case you need to access your work from another machine. This also protects against cases where a lab machine's C:\Users directory is emptied (part of routine space maintenance) or the machine is reimaged. If you log off a lab machine before OneDrive finishes syncing the files remain stuck on that one machine until you log back in and syncing completes.

4. If not using OneDrive, back up your working folder (and any other needed files) to Box, a USB flash drive, or another safe location (for example, one option is to copy all of the files to Z:, possibly skipping the ODF .gdb to save space).

Submission

Prepare answers for all the questions above in a Word document. Submit a .docx or .pdf on Canvas.[7]

References

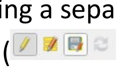
Demetriou D, See L, Stillwell J. 2013. A Parcel Shape Index for Use in Land Consolidation Planning. Transactions in GIS 17(6):861–882. https://doi.org/10.1111/j.1467-9671.2012.01371.x

Supplementary Material: QGIS

QGIS (https://qgis.org/) is an open source GIS solution which builds on other open source GIS tools from the OSGeo project (https://www.osgeo.org/), primarily GDAL and GRASS GIS. QGIS is generally more reliable than ESRI's Arc products, runs faster, is free to download, and provides broadly the same functionality. While there are differences in what Arc and QGIS support, they lie in details of QGIS plugins and ESRI toolboxes and are typically important only to niche use cases. While this lab is written for ArcGIS Pro and ArcMap because Arc is often used in other OSU courses, it can also be performed entirely in QGIS.

Task S1: Create a QGIS Project with West Oregon District layers

Set up the working folder as described above, start QGIS, create a new project, and save the project file in the working folder. Rather than ESRI's proprietary geodatabase format (which can be read from open source tools), QGIS defaults to storing layers in GeoPackage files, which is an open format (.gpkg, https://www.geopackage.org/).

1. Use Layer → Add Layer → Add Vector Layer to add the ODF geodatabase. Select Source Type = Directory, choose Source = OpenFileGDB in the dropdown, and select the ODF .gdb as the folder. Press Add and select the layers from the .gdb.
2. Follow the rest of the steps in Task 1 using their QGIS equivalents (Table S1). QGIS uses the same start editing, save edits, stop editing for layer changes as ArcMap with the difference that, rather than having a separate editor toolbar, edit toggle and save buttons on the attribute table toolbar are used ().
3. QGIS lacks Arc's distinction between calculating fields and calculating geometry, using its field calculator for both. Rather than automatically maintaining Shape_Length and Shape_Area fields as Arc does, fields can be set to length or area using `$length` or `$area` in QGIS's field calculator. A quirk of QGIS is `$length` follows the units of a layer's CRS but `$area` always returns square meters. Therefore, the calculation for acres is `$area / 4046.86`.
4. What Arc calls unique values symbology QGIS calls categorized symbology. All three have the same process for classification and changing color ramps, just with somewhat different user interface layouts. The add all values button in Arc is called classify in QGIS.

[7] Reminder for Mac users: neither Canvas or Word supports .pages files, which prevents opening them for grading.

Task S2: Create layers for riparian buffers, murrelets, ownership, and counties

1. The largest difference between Arc and QGIS for these tasks is the QGIS field calculator uses its own scripting language rather than Python. In QGIS, Python run using PyQGIS in the Python Console. PyQGIS is beyond the scope of this document.
2. Specifying a field for variable width buffering is done using the dropdown at the far right of the distance box of QGIS's buffer tool.

The QGIS field calculator equivalent of `getBufferWidth()` is

```
if("Fishpres" = 'Fish', 115,
case
when "Fpasize" = 'Large' then 75
when "Fpasize" = 'Medium' then 75
else 50
end
)
```

Task S3: Create layers for riparian buffers, murrelets, ownership, and counties

Differences in these tasks are just in the field calculator syntax. Replacing empty values in the QGIS field calculator can be done with

```
if("Wildlife" is not "", "Wildlife", 'noListedSpecies')
if("SlopePosition" is not "", "SlopePosition", 'upland')
```

The calculation for shape index becomes

```
$perimeter / (2 * sqrt(3.1415 * $area))
```

Table S1: Names and locations of QGIS equivalents of the Arc tools described in the main section of this document.

ArcGIS Pro and ArcMap	QGIS	QGIS location(s)
select by attributes	select features by expression	main toolbar button dropdown
export features	export → save features as	layer right click
zoom to layer	zoom to layer	layer right click
dissolve	Dissolve	processing toolbox search Vector → Geoprocessing → Dissolve
open attribute table	open attribute table	layer right click
select by location	select features by polygon	main toolbar button dropdown
delete field	delete field	attribute table toolbar
change field alias or name	copy to new field and delete old	attribute table toolbar: new field, delete field, field calculator
field calculator	field calculator	attribute table toolbar
calculate geometry	field calculator	attribute table toolbar
find and replace	field calculator	attribute table toolbar
symbology	properties → symbology	layer right click

intersect	intersect	processing toolbox search Vector → Geoprocessing → Intersection
Python	field calculator script PyQGIS	attribute table toolbar Plugins → Python Console
buffer	buffer	processing toolbox search Vector → Geoprocessing → Buffer
clip	clip	processing toolbox search Vector → Geoprocessing → Clip
dissolve	dissolve	processing toolbox search Vector → Geoprocessing → Dissolve
check geometry	check validity	processing toolbox search Vector → Geometry → Check Validity
repair geometry	fix geometries	processing toolbox search
project	reproject layer	processing toolbox search
intersect	intersect	processing toolbox search Vector > Geoprocessing > Intersection
union	union	processing toolbox search Vector > Geoprocessing > Union
eliminate	eliminate selected polygons	processing toolbox search
join	properties > join	layer right click

with `$perimeter` always returning meters and `$area` m² in QGIS. Then

```
(("ShapeIndex" > 3.1) and ("ACRES" < 0.8)) or ("ACRES" < 0.2)
```

Task S4: Create Woodstock shapefile

Joins are a standard operation on SQL databases and therefore work the same way across many tools beyond Arc and QGIS. QGIS's terminology is slightly different from Arc's, referring to the field in the table being joined as the join field and the field in the layer as the target field.

The field calculation for age1 field simplifies to `if("age" = 0, 1, "age")` and, to export a shapefile, select ESRI Shapefile in the Format dropdown of Export > Save Features As.

Assignment 3: Forest Management Plan Introduction

Introduction

In this assignment your group will decide the specific goals and objectives for your management plan projects and draft the introduction of your plan. Much of the information needed for this assignment is available from the GIS assignment and, similarly, all subsequent assignments in this course build on the management plan introduction. Well-reasoned choices of objectives with good planning support in the introduction therefore make quality work easier throughout the rest of the course. Conversely, poor execution in the introduction tends to increase the difficulty of subsequent assignments.

A forest management plan may contain different elements depending on the ownership, district, or forest for which it is written, as well as the company, organization, agency, or other entity that prepares it. However, in general, a plan contains:

- Introduction
- Goals, Objectives, Constraints, and Issues
- Overview and Description of Resources
- Intended Management
- Landscape (Spatial) Planning
- Recommendations
- Conclusions

A forest management plan introduction begins with a statement of purpose. This purpose is the motivation for developing the plan and what guides the planning process and development. It is followed by a description of the goals of the forest, the associated objectives, and how these will be achieved and measured. Some objectives are well supported by operational forest management tools (e.g. revenue or volume) while others are not (e.g. wildlife habitat, climate adaptation). Regardless, how objectives will be measured must be clearly stated and supported within the introduction. The introductory section also includes a description the planning area including location, ownership, classification, history, physical conditions, biological conditions, and human components. These sections should provide readers with a good understanding of the planning area and all details relevant to plan development.

Maps, charts, and summary tables are also included in a forest management plan introduction. Maps often present information such as the forest's location, land use classifications and reserve areas, soils, site productivity, stand types, and age class distributions.

Learning Objectives

- Understand goals, objectives, constraints, and issues related to forest management planning.
- Understand the information, data, and other contents that should be included in the introduction of a forest management plan.

Tasks

- Develop and state the goals and supporting objectives of your group's forest management plan.
- State the constraints and issues that must be considered during planning.
- Write the introductory chapter of a plan for managing the Oregon Department of Forestry's West Oregon District under your group's objectives. This will become the first chapter of your final group project plan.

Forest Management Plan Introduction Chapter

Below are the *minimum* elements your management plan introduction should contain. Your group should include additional information and display maps, tables, charts, and figures that best align with the chosen objectives.

Introduction

- Statement of management plan purpose

Goals, Objectives, Constraints, and Issues

- ODF specific
- Regulatory (i.e., Oregon Forest Practices Act)
- Group selected

Location and Ownership

- Description of the West Oregon District location, ownership, classification, and brief history
- Include a table of area distribution by ownership type and county
- Include a map of the planning area

Physical Conditions

- Description of the geology, soils, topography (slope, elevation, etc.), hydrology, and climate on the District
- Include a table soil and topographic distribution by area
- Include a soils map (topography map could also be included)

Biological Conditions

- *Vegetation*
  - Description of main forest types and plant associations
  - Include a table of the area distribution of forest types and associated measures such as age, DBH, height, volume, etc.
- *Forest Health*
  - Description of the primary forest health concerns present on the District (insects, disease, abiotic factors, other disturbances)
- *Fish and Wildlife*
  - Description of the primary species that utilize the District, as well as fish-bearing streams and associated fish species
  - Include a table of species present that are threatened, endangered, or of concern

Human Components

- Description of human activities/uses on the District, including recreation, aesthetic values, and cultural resources

Your report should be formally written for a professional audience, including appropriate citations and references. Much of the information needed for this assignment can be found within "*ODF – Northwest Oregon State Forests Management Plan,*" "*ODF – West Oregon District Implementation Plan*," and other information provided by Blake McKinley for this course. These resources are on Canvas as well as the T: drive.

Forest Management (Davis et al. 2001) and *Forest Plans of North America* (Siry et al. 2015) may be helpful in writing.

Submission

Name the file for the chapter "A3 <GroupName> Introduction" and upload it to the Assignment 3 page on Canvas as a .docx or .pdf. Only one submission is needed per group.

Assignment 4: Management Plan Prescriptions

Introduction

A forest management plan depends on the size of the forest to be managed. In small forests, it is feasible for management plans to consider individual silvicultural prescriptions and management options for each stand. In large forests, there are too many stands for it to be practical to include such detailed planning in strategic forest plans. Stands are therefore typically classified by species composition, productivity, and age. Planning tools such as Woodstock are then given a set of silvicultural prescriptions which can be applied to different stands and asked to find the combination and timing of prescriptions which best meets the management goals. In general, this optimization process becomes more precise if it is given more stand classifications and silvicultural options. However, the resulting plan cannot be more accurate than the classifications and prescriptions it work with. A speed-accuracy tradeoff therefore applies.

Learning Objectives

- Classify stands into landscape level themes for applying silvicultural prescriptions.
- Develop silvicultural prescriptions that meet forest management and stand level objectives.
- Create growth and yield tables for relevant combinations of classifications and prescriptions.

Tasks

Each group should create multiple silvicultural prescriptions that meet the forest management objectives and simulate them using the US Forest Service's FVS (Forest Vegetation Simulator). Because each prescription must be transferred from FVS to Woodstock, prescriptions should be developed with Woodstock integration in mind. The most common way of presenting stand management options to Woodstock is tables in its yield section. Each yield table begins with a *Y keyword listing a combination of landscape themes to which the table applies. These link each yield table to stands with the same combination of themes as well as to Woodstock actions and transitions which can be applied to those themes. The six themes defined in the GIS assignment (county, ownership, wildlife, slope position, stand, and desired future condition) therefore need to be extended with additional themes.

At least three additional themes are required in this assignment.

1. a species theme with minimum of six species compositions
2. a productivity theme with at least seven productivity classes
3. a stocking theme containing information about stand density.

Each group should develop and support their choice of themes according to their management objectives. Additional themes may be required depending on specific management objectives.

These themes and associated silvicultural prescriptions constitute the second chapter of the forest management plan. Once written, this chapter should clearly and concisely present the silviculture and proposed activities used to manage the forest. Prescriptions' defining properties (species composition, productivity, planting density, thinning age and intensity, rotation age, and any other important considerations) should be described. Motivations for selecting prescriptions should also be described, along with how the prescriptions support the forest management plan's objectives. Figures, tables, flowcharts, or other presentation tools should be used communicate each prescription's stand development and how prescriptions will be used as management tools.

This assignment also includes preparing FVS yield tables in Excel that are ready to be transferred to a Woodstock yield section. Since these tables control the accuracy of the rest of the management plan, the FVS run which produces them should be of good quality. A yield table is required for every combination of silvicultural themes present on the landscape, making it important to track each table such that Woodstock gets the correct yields for each combination of themes. **Thinning and regeneration harvest (clearcutting) do not need to be included within FVS runs** for yield tabling as both can be acceptably approximated with Woodstock actions. However, FVS runs including thinning are needed to assess prescriptions' silvicultural results and are very likely to be needed in the next assignment to determine how many years to `_SHIFT` stands by when modeling thinning. It may also be helpful to produce Stand Visualization System (SVS) figures in FVS (Figure 1).

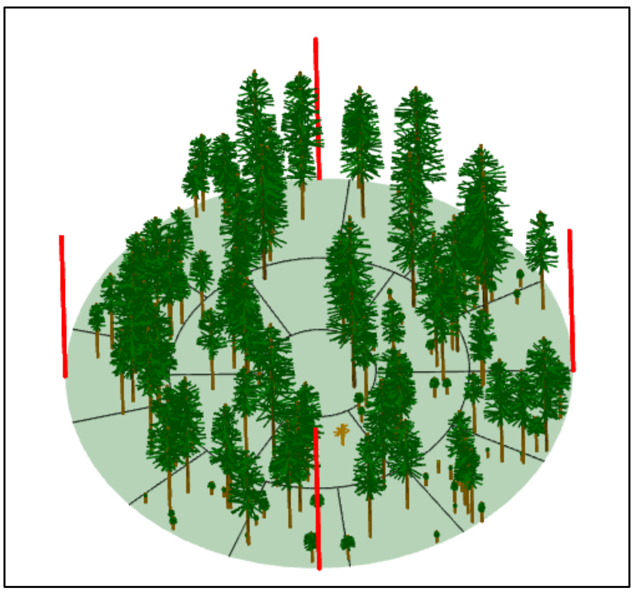


Figure 1: Example of a Stand Visualization System figure, output from FVS, portraying stand structure at a given simulation timestep.

Management Plan Chapter and Yield Tables

There are two elements required to be submitted for this assignment: the second chapter of your forest management plans and the prepared FVS yield tables in Excel. The following are the minimum elements required for each; however, your group may choose to include additional information and/or tables, charts, and figures that best align with the purpose/objective of your plan:

Forest Management Plan Chapter 2: Management Activities (65% of grade)

Overview of Current Conditions

- Description of silvicultural themes and associated management activities.
- Relationship of themes to desired future conditions under ODF requirements and group objectives.
- Description of how the theme values chosen allow the planning objectives to be met.
- Current area distribution of silvicultural theme classifications.

<u>Silvicultural Prescriptions</u>

- Describe the defining properties of each prescription, why prescriptions were selected, and how they are appropriate to the stand and planning objectives and desired future conditions.
- Summarize the silvicultural properties of each prescription (stand entry timing, harvest intensities, reforestation choices, growth and yield, and any other characteristics of interest to the management objectives).

Excel File Containing Yield Tables (35% of grade)
A longform spreadsheet ready to be converted to a Woodstock .yld file as described in this assignment's lab material. Each yield table should be preceded by a *Y keyword whose masks match the themes chosen.

The management plan chapter should be formally written for a professional audience. Refer to the Oregon Department of Forestry's management plans, as well as the provided ODF Capstone Constraints document, for information on the current and desired future conditions—as well as your group's own management objectives. When used well (which often requires reworking of FVS output) figures are often efficient for communicating differences among prescriptions by allowing readers to easily compare and contrast differences between prescriptions. Additionally, *Silviculture and Ecology of Western U.S. Forests* (Tappeneiner et al. 2015) is a great resource for information on developing silvicultural prescriptions.

Submission

Name the file for the chapter "A4 <GroupName> Silviculture" and upload it to the Assignment 4 page on Canvas as a .docx or .pdf. Only one submission is needed per group.

Introduction to FVS

1 Introduction

The Forest Vegetation Simulator (FVS) is the United States Forest Service's nationwide tool for forest growth and yield modeling. While FVS has many regional variants and settings specific to individual national forests it must be configured correctly to produce acceptably accurate results.

Reference material used for development of the present document includes:

- Essential FVS: A User's Guide to the Forest Vegetation Simulator (Dixon 2022)
- Keyword Reference Guide for the Forest Vegetation Simulator (Van Dyck and Smith-Mateja 2022)
- Users Guide to the Database Extension of the Forest Vegetation Simulator Version 3.0 (FVS with SQLite) (Crookston et al. 2021)
- Pacific Northwest Coast (PN) Variant Overview of the Forest Vegetation Simulator (FVS Staff 2021)

These guides and other documentation are available from the FVS website. The NVEL (National Volume Estimator Library) and NSVB (National Scale Volume and Biomass Estimators) documentation is also helpful. The US Forest Service makes FVS freely downloadable for use on machines running Windows, so FVS simulations need not be restricted to lab machines.

2 Objectives

- Setup and run stand growth simulations using FVS version 20230518 or newer.
- Obtain growth and yield tables for input to Woodstock.

3 FVS database setup and stand configuration

3.1 Create a new project and empty the database

Because ODF's cruise records for the West Oregon District are not available we will grow stands from planting (stand age zero) and, therefore, work with an empty FVS database without measurements of existing trees. After starting FVS using the desktop shortcut or Start → FVS it will launch in a browser window.[1] Go to Manage Projects → Manage project and type in a suitable project name (Figure 1). Press Make new project, select the newly created project from the dropdown under Select project, and press Open selected project. A new instance of FVS will start in another browser tab and, at the top, you should see Project title: <the project name you just typed in>.

[1] If a lab machine is missing both shortcuts FVS can be started directly by running C:\FVS\FVSSoftware\FVS_Icon.VBS.

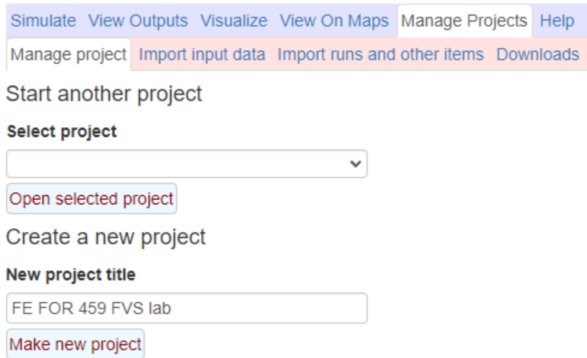


Figure 1: Location of Make new project and Open selected project in FVS menus.

To see the four tables stored inside FVS's database select Manage Projects → Import input data → View and edit existing tables, which will show the FVS_GroupAddFilesAndKeywords, FVS_PlotInit, FVS_StandInit (Figure 2), and FVS_TreeInit.

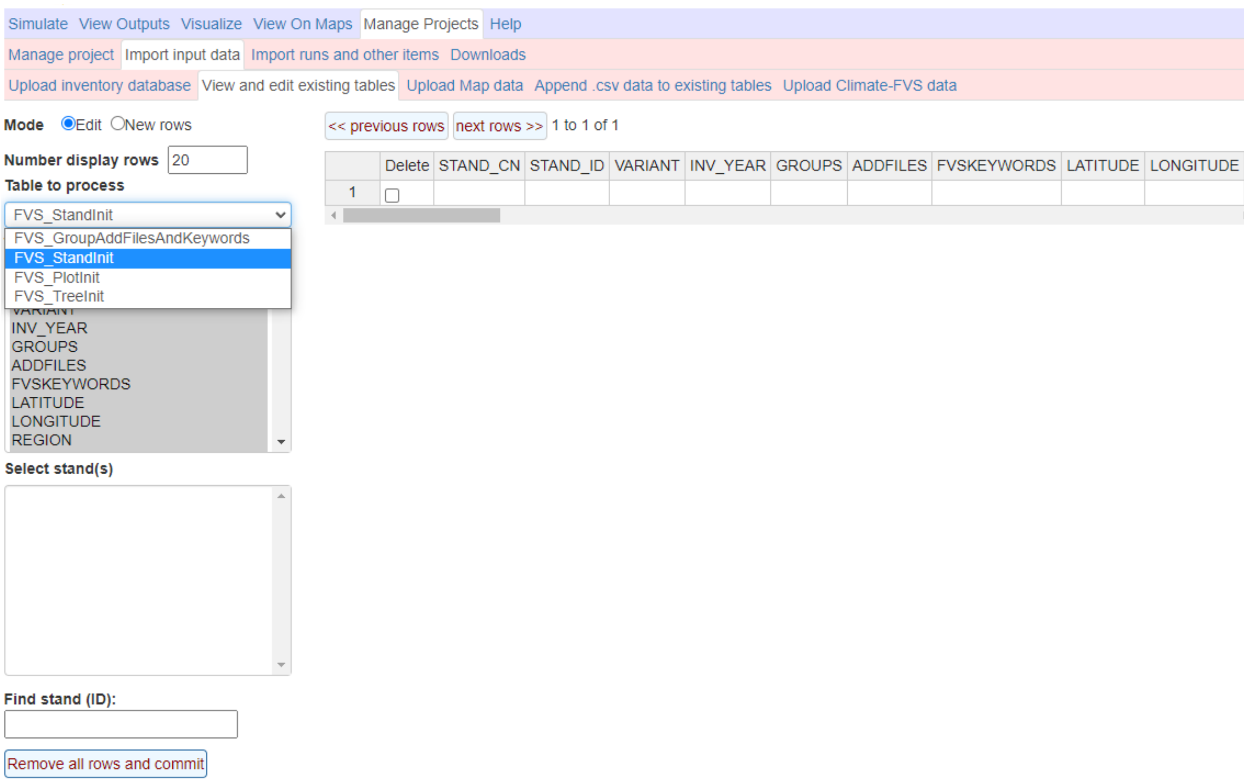


Figure 2: Dropdown menu for viewing contents of the FVS_GroupAddFilesAndKeywords, FVS_PlotInit, FVS_StandInit, and FVS_TreeInit tables and the location of the Remove all rows and commit button.

3.2 FVS_GroupAddFilesAndKeywords table

The FVS_GroupAddFilesAndKeywords table which connect the FVS to its underlying Access database. While it may eventually be deprecated out of FVS, problems with this table easily prevent FVS from seeing the other three tables' contents, causing simulations to provide no output. The table contains only three columns (GROUPS, ADDFILES, and FVSKEYWORDS) and no changes are required.

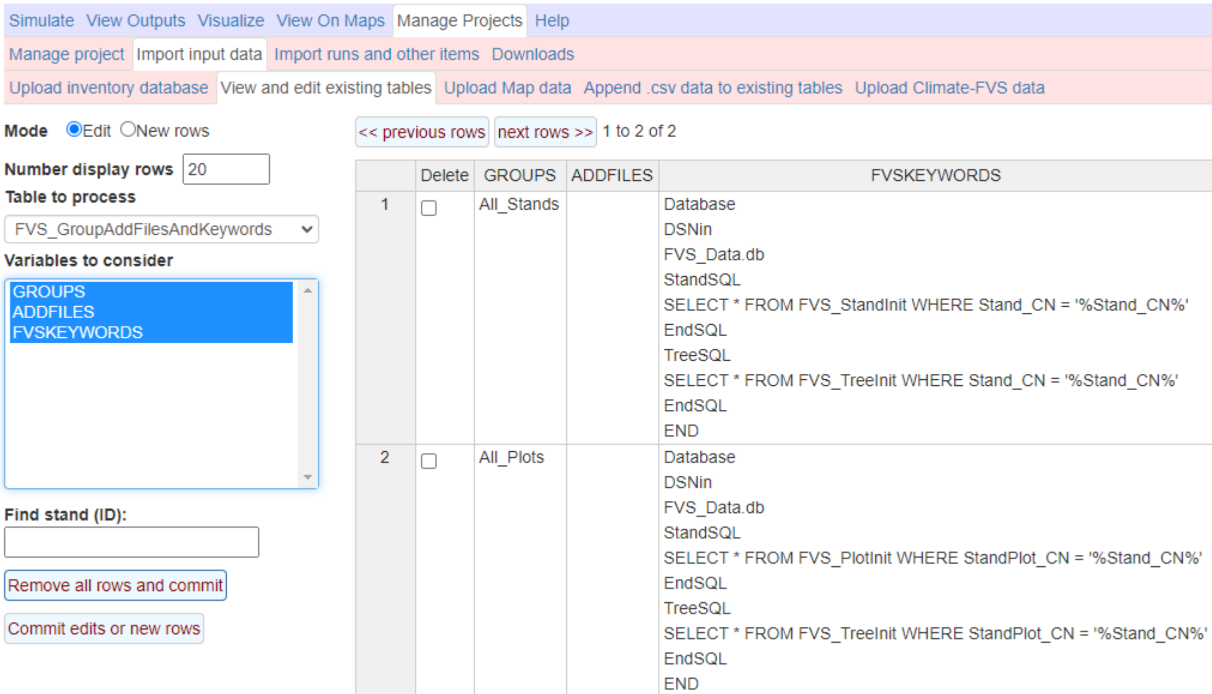


Figure 3: Expected state of FVS_GroupAddFilesAndKeywords table.

3.3 FVS_PlotInit

The FVS_PlotInit table is used to store subplot data. By default, it is populated with a set of plots. Since none of these plots are needed for this course, use the Remove all rows and commit button to empty the table (Figure 4). No further use of this table is needed.

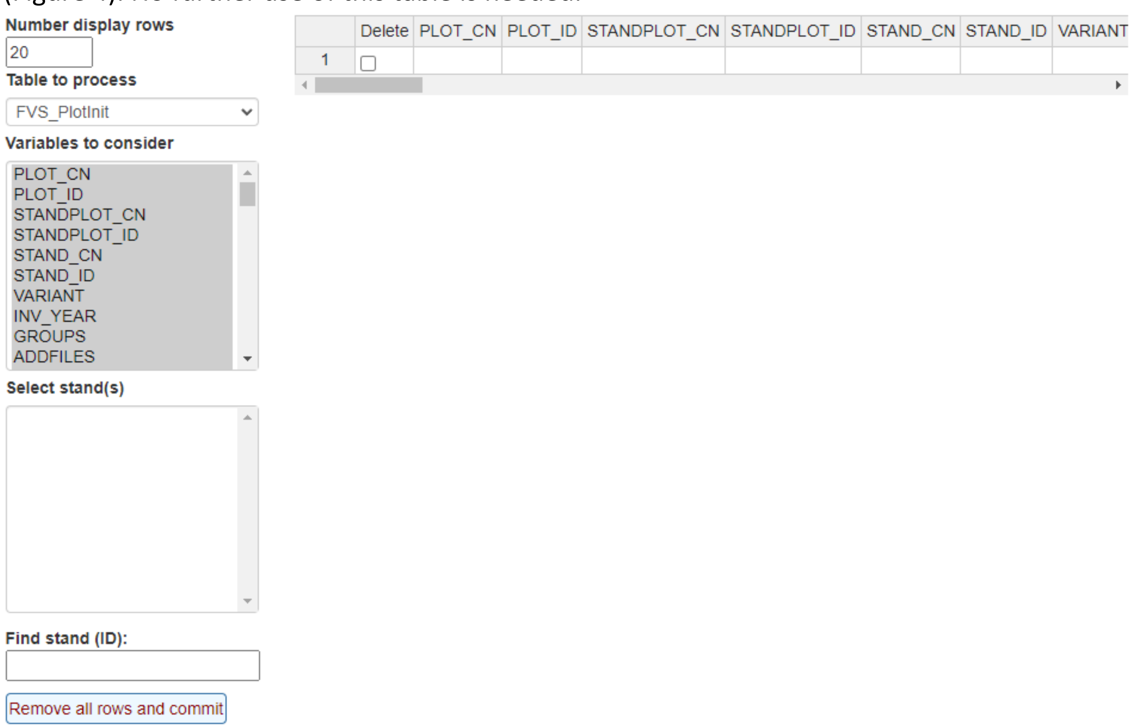


Figure 4: Empty FVS_PlotInit table.

3.4 FVS_StandInit table

The FVS_StandInit table contains stand level information, expressed through cruise plots. As with FVS_PlotInit, use the Remove all rows and commit button to empty the table.

There are four mandatory fields for each stand: STAND_CN, STAND_ID, VARIANT, and INV_YEAR. However, 11 other columns are important (Figure 5, Table 1). If one of these columns is not set, FVS will use a default value which is often unsuited to the stand being simulated. The STAND_CN and STAND_ID fields identify individual stands within a forest inventory as FVS can handle multiple stands. These two required fields connect the stand attributes in FVS_StandInit with the stand's plots and trees as defined in the stand and the plot tables. To add stands, choose Mode = New rows at upper left, click on a row, and add information by typing in each cell. Entering an informative name in STAND_CN is recommended. For example, stands for simulating Douglas-fir plantations might be named "DF high", "DF medium", and "DF low" depending on site productivity. Since FVS sometimes displays only STAND_ID it can also be helpful to choose informative stand ID numbers. For example, an ID of 2020140 could be chosen to indicate a predominantly Douglas-fir stand (FIA species code 202) with a 50-year site index of 140 feet.

The VARIANT field is required to identify the equations and default values for a particular area throughout North America. Oregon is covered by seven variants. The variants for western Oregon are Pacific Northwest Coast, Westside Cascades, Organon NWO/SMC, and Organon SWO. Due to FVS limitations in how it uses Organon the preferred variant for Oregon's Coast Range is the Pacific Northwest Coast variant, which is abbreviated **pn** (Figure 5, Figure 6).

The INV_YEAR column identifies the year when inventory was carried out and marks the start of the simulation. Because we will choose to plant stands at INV_YEAR in Section 4.2, INV_YEAR corresponds to a stand age of zero. Since planting has been ongoing on the West Oregon District for decades, there is no one choice of INV_YEAR which makes dates in FVS simulations match up with stands' actual growth. However, it can be easier to keep track of things if the same choice of INV_YEAR is made for all stands in FVS and that choice happens to be within a few decades of common planting times on the District.

Figure 5: Entering data in the FVS_StandInit table after using the Remove all rows and commit button to clear the table. Note the Mode radio button at upper left; to later add second, third, fourth, and additional stands Mode must be set to New rows and both a STAND_CN and STAND_ID entered for each new stand being added.

Table 1: Required columns in FVS_StandInit.

field	value	notes
STAND_ID	stand number	Should be a number. FVS may not load a stand if letters are used.
STAND_CN	name of stand	An informative stand name (can contain letters).
VARIANT	pn	pn = Oregon and Washington coast, op = Organon NWO/SMC
INV_YEAR	semi-arbitrary	Forty years before rpt_yr in SLI_ROOTS_OneLiner is suggested.
GROUPS	All_Stands	Described below.
REGION	6	USFS Region 6 is Oregon and Washington.
FOREST	12	The Siuslaw National Forest is forest 12 in Region 6.
LOCATION	612	Combined region and forest code. Tells FVS to use cubic and Scribner volume equations applicable to the Siuslaw.
BASAL_AREA_FACTOR	see below	Prism factor (BAF) used with trees larger than BRK_DBH.
INV_PLOT_SIZE	see below	Inverse of the fixed area plot size used with trees smaller than BRK_DBH (*e.g.* 0.1 acre plots ⇒ INV_PLOT_SIZE = 10).
BRK_DBH	999	Breakpoint DBH, in inches, between BAF and fixed radius plots.
MAX_SDI	SDI_{max}	Reineke SDI_{max} for SITE_SPECIES in English units. If not specified, FVS defaults to the highest value it knows of, which is unlikely to be appropriate. Consider also SDI_{max} typically increases somewhat as site index increases.
SITE_SPECIES	main species	FVS species code for Douglas-fir, western hemlock, red alder, *etc*. depending on stand type.
SITE_INDEX	see below	The 20, 50, or 100 year site index for SITE_SPECIES in feet (refer to FVS variant documentation for which ages are used with which species). If not specified, FVS typically defaults to a low value.

Other fields can be left blank (ADDFILES, FVSKEYWORDS, GIS_LINK, PROJECT_NAME, LATITUDE, LONGITUDE, DATUM, DISTRICT, COMPARTMENT, ECOREGION, PV_CODE, PV_REF_CODE, AGE, ASPECT, SLOPE, ELEVATION, …).

All stands should have All_Stands in their GROUPS column. Multiple groups can be listed, each separated by a space. In large projects, using additional groups helps with identification of stands within a database with containing many stands.

For inventoried stands (where tree measurements are present in the FVS_TreeInit table), cruise information is required. The BASAL_AREA_FACTOR and INV_PLOT_SIZE fields indicate the variable and fixed radius plot sizes, respectively, used in nested plot cruising. For inventories using only variable or fixed radius plots BRK_DBH can be set to either 0 or 999 to exclude the other plot type. The final inventory field is the number of plots in the stand, which is entered in NUM_PLOTS. FVS calculates trees' expansion factors from these four fields as usual.

The last six required columns provide simulation accuracy. To grow a stand, information regarding site productivity and carrying capacity is needed, therefore, values for SITE_SPECIES, SITE_INDEX, and often MAX_SDI are needed. If no value is entered the default value for the respective variant is used, which is likely undesirable (Table 1). Refer to the FVS variant documentation for species codes to use in SITE_SPECIES (Section 1, see also Section 3.5). REGION, FOREST, and LOCATION control which NVEL

(National Volume Estimator Library) equations and log lengths FVS defaults to for estimation of merchantable stemwood yields. Use of the wrong equations can result in significant tree taper and Scribner board foot errors.

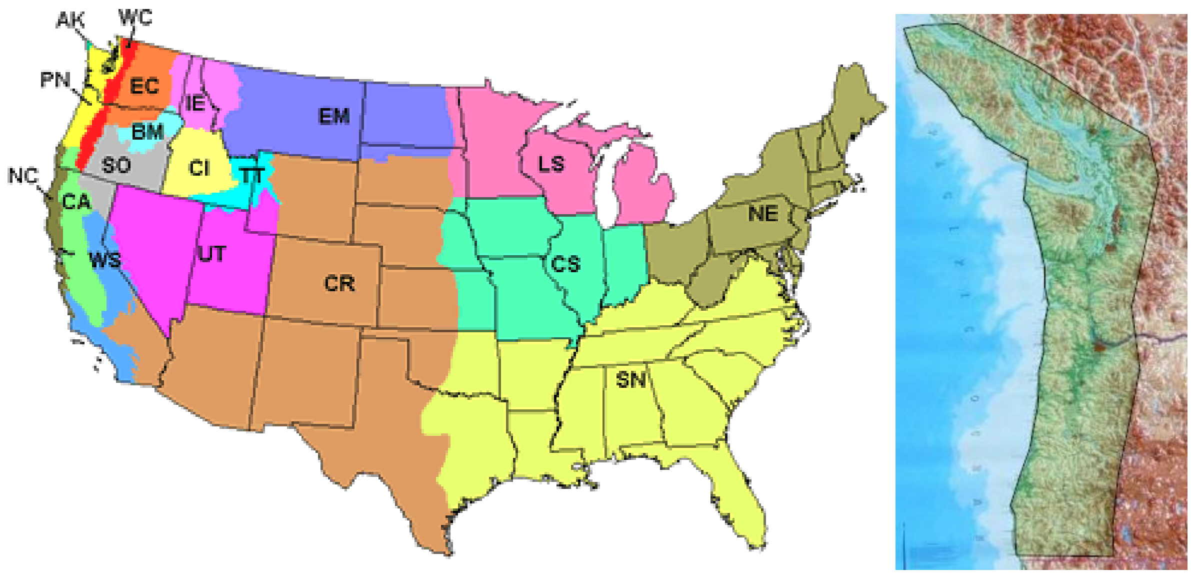


Figure 6: Geographic coverage of FVS variants for the contiguous United States (left, there is also an Alaska variant) and the FVS-OP ORGANON Pacific Northwest variant (right) (US Forest Service).

Based on the descriptions above and Table 1, enter values for the 13 primary fields describing at least one stand. Use the Commit edits or new rows button to save your changes to FVS_StandInit. (You can return to the table to edit stands or add more stands of interest at any time.)

3.5 FVS_TreeInit table

The FVS_TreeInit table contains individual tree information. Tree records must match FVS_StandInit. Each tree has to belong to one stand, identified by STAND_CN, and to one plot, identified by PLOT_ID. FVS also requires a SPECIES entry, which is an FVS, FIA, or USDA Plants species code, and the tree's DBH in inches. FVS species codes are usually two letters (*e.g*. DF for Douglas-fir and WH for western hemlock). FIA codes are three or four digits (202 for Douglas-fir and for 263 western hemlock). The USDA Plants database uses primarily four letter binomial codes (Douglas-fir is PSME and western hemlock TSHE). Refer to each FVS variant's documentation for the list of species it supports.[2]

Since, in FE/FOR 459 tree growth modeling, each stand will be grown from seedlings planted at stand age zero (in forest inventory, stand ages are often offset from the trees' age from germination by the seedlings' age at planting), use the Remove all rows and commit button once again to empty the FVS_TreeInit table.

---

[2] In general, growth models for temperate and boreal forests support only the most abundant tree species. Depending on the quality and nature of the model's input checking it may raise errors on unsupported species, silently reassign them to a supported species deemed equivalent, or just ignore the tree entirely. If in doubt, the most reliable check is to look at the model's tree list output to see how it's growing uncertain trees.

Figure 7: An empty FVS_TreeInit table.

Once you have completed the database preparation you can start running FVS growth and yield simulations.

4 Getting started with growth and yield prediction

4.1 Stand selection

To obtain the tree growth predictions needed for forest planning in Woodstock you should simulate FVS according to the prescriptions you have planned. First, the stand to be simulated should be selected under Simulate → Stands submenu. FVS_StandInit should be selected as the default Inventory Database Table, along with FVS-PN as the default variant. The All_Stands group should show in the Groups box and, after selecting All_Stands, you should see the stands available inside the All_Stands group. After selecting a stand to include in a simulation, press Add selected stands. The stands will be transferred to the Run Contents window (*e.g.* Figure 8). Check that Kwd: From: FVS_GroupAddFilesAndKeywords is included in the Run Contents pane. If this keyword is not present the run won't read stand information from the FVS_StandInit table and the results will not be as expected.[3]

The default title of an FVS simulation run is "Run" followed by an incrementing ID number. It's useful to give runs more descriptive names. Figure 8 illustrates one possibility for a high site index Douglas-fir stand planted at 450 seedlings ac^{-1}.

4.2 Simulation setup and execution

After selecting stands to simulate, choose the length of time for which the stand will be grown. Because growth and yield predictions should cover stand ages within the capstone project's planning horizon, it would be ideal to run FVS simulations to least the age of oldest stand in the Western Oregon District's inventory plus the project's 50-year horizon (see Remsoft's Lifespan section guidelines in the Woodstock documentation). However, FVS will report no more than 40 growth intervals from the stand inventory date. With five-year intervals the maximum simulation duration is thus 5 • 40 = 200 years (Figure 9).

[3] FVS should insert FVS_GroupAddFilesAndKeywords automatically but it doesn't always do so.

Running FVS simulations to at least 200 years is encouraged. Groups with old forest objectives will likely need to specify longer intervals to accommodate longer planning horizons.[4]

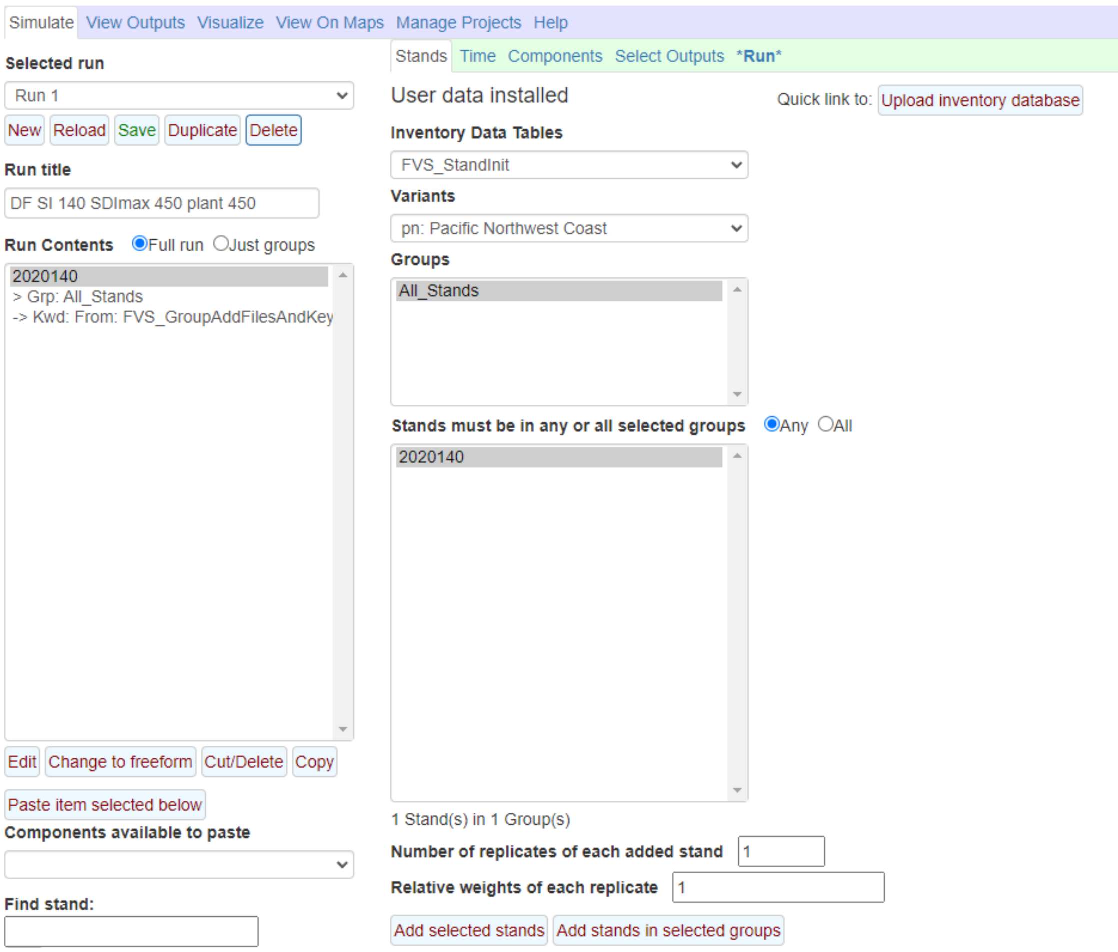


Figure 8: Population of Run Contents with Kwd: From: FVS_GroupAddFilesAndKeywords after selecting a stand to simulate and changing the run title.

The next step in modeling the growth and yield of the stand is selection of management actions through the Components tab. To plant trees, choose Components → Management → Categories → Planting & Natural regeneration. Under Planting & Natural regeneration category there is only one component: Plant/Natural with Partial Estab[lishment] Model. Because planting occurs at the start of simulation, set Years following disturbance for site preparation = 0. In the following boxes, indicate the planting details. As a default, assume 90% survival rate and, for spring germinants with late fall planting,

[4] As of FVS 20220118, FVS stops simulations whenever it reaches its 40th output, regardless of whether that output is a regular timestep year or an additional output reporting year requested through that box (Figure 9). If a stand's age goes beyond the range of _AGE in its Woodstock *Y yield table, Woodstock reuses the oldest row in the yield table until the _DEATH action triggers (see Yields section guidelines in Woodstock's documentation). Provided the stand is old enough to have more or less reached a steady state, relying on Woodstock repeating FVS's last row of output is very likely an acceptable approximation within the scope of this course.

planting at age 1.5 years when the height is about 1.5 feet. Once finished, press Save in run, which will add > Kwd: Plant/Natural with Estab Model to the Run Contents (Figure 10).

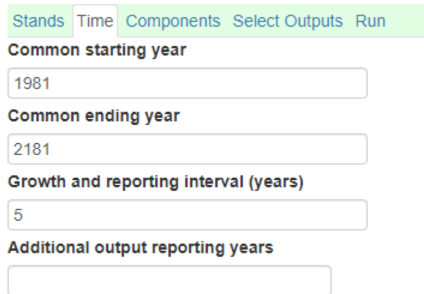


Figure 9: Configuration of a 200-year run with a five year timestep. The Common starting year should be whatever you chose for INV_YEAR when entering the stands.

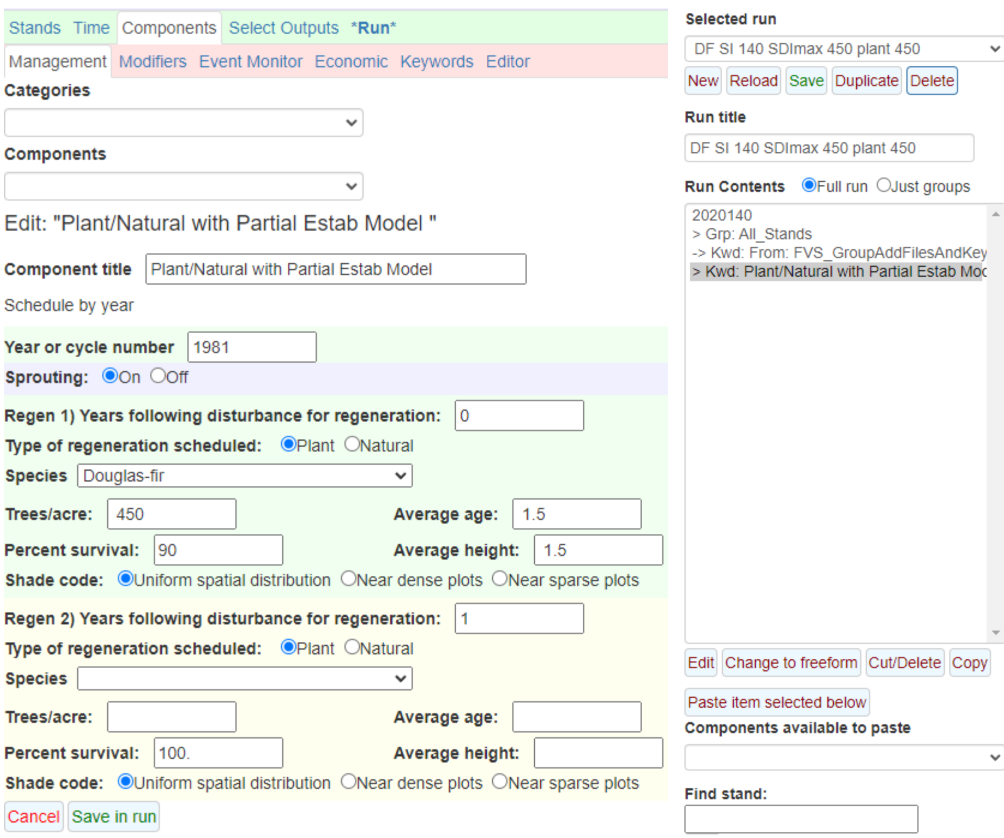


Figure 10: Setting up planting (left) and the resulting FVS keyword added to the run contents (right). To modify the planting or check the values entered, select the keyword as shown and use the Edit button.

Now everything is set for a successful run.[5] Choose the Run tab and press the Save and Run button (Figure 11). After you run FVS you should check for errors, which are displayed below the chart showing the increase in volume with age (Figure 12).

---

[5] SLI_Roots_OneLiner has rpt_yr, birthyr, and tot_age fields and SLI_Structure has a corresponding age column. birthyr is the year the stand was established, not the year the seedlings germinated (see

If no errors are found and the volume prediction appears reasonable given the species mixture, planting density, site productivity, SDI_{max}, and log rules chosen then you can export the FVS output in a permanent format. To save the run press the FVS Main Output File button (Figure 12), which copies FVS's default output to your Downloads directory. The name of this text file should be the name of the run followed by _FVSoutput.txt.

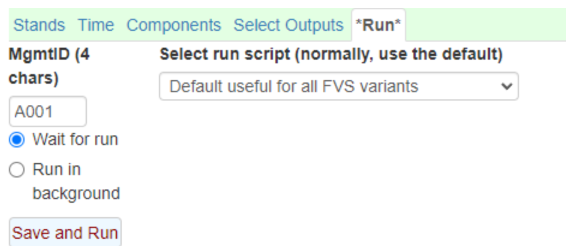


Figure 11: The Save and Run button for saving an FVS configuration to disk, invoking FVS, and displaying the results. (Note that Save and Run and Save button under the Selected run save different aspects of the run, so use of both is needed.)

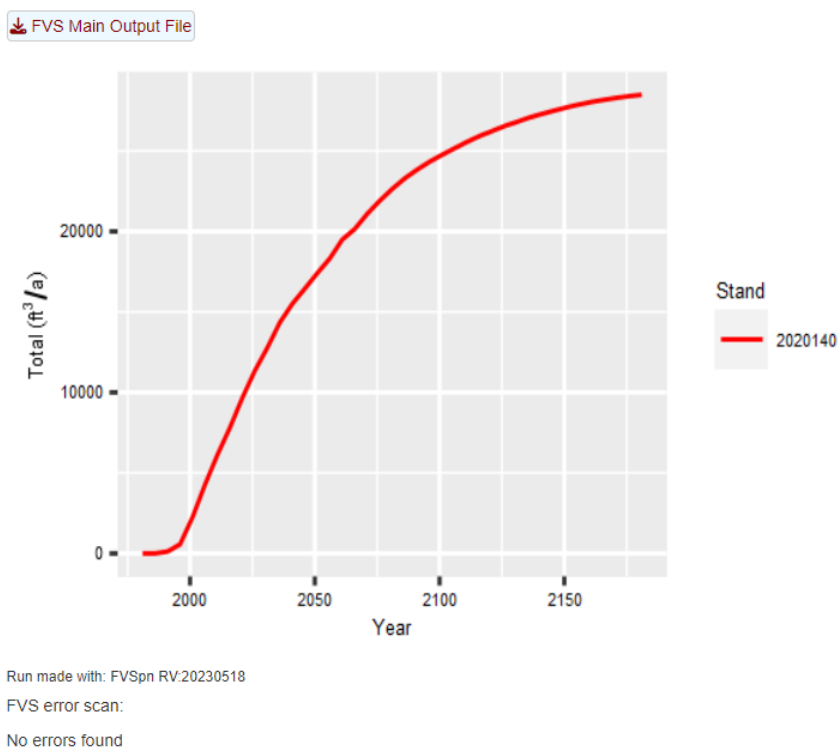


Figure 12: An example FVS run result with no errors found during the error scan. The FVS Main Output File button is at upper right.

ODF_SLI_Tables_metadata.xlsx). For the most part, age = tot_age = rpt_yr – birthyr, meaning stand ages in the GIS files do not include seedlings' age at planting. Similarly, FVS calls the year simulations start age zero, so ages in FVS also neglect seedlings' age at planting. Therefore, starting age bias in stand growth and yield estimates is minimized when plantings in FVS match the stock used by ODF.

4.3 Reading and error checking FVS's main output file

The FVS main output file contains information about the run itself, as well as summary statistics for the intervals specified in the Time tab. Several pieces of information are important to verifying FVS is correctly configured. If the STANDSQL keyword does not appear in the output file or the region, forest, BAF, plot size, break DBH, site index, or SDI_{max} does not match what was specified for the stand then the growth predictions made by FVS won't provide the desired result (Figures 13–15). The main output file also contains stand level summary statistics but usually it's more efficient to work with FVS's View Outputs menu and .xlsx export options.

```
STANDSQL    STANDSQL COMMAND FOR INPUT DATA BASE: FVS_Data.db
            SELECT * FROM FVS_StandInit WHERE Stand_CN = '%Stand_CN%'

            STAND-LEVEL DATA BASE READ:
            INV_YEAR:                 2021
            REGION:                      6
            FOREST:                     12
             COMPOSITE LOC:            612
            LOCATION:                  612
            BASAL_AREA_FACTOR:         0.0
            INV_PLOT_SIZE:              1.
            BRK_DBH:                 999.0
            NUM_PLOTS:                   1
            SITE_SPECIES:               DF MAPPED TO INTERNAL CODE: DF
            SITE_INDEX:              140.0 FOR SPECIES: DF
            MAX_SDI:                 450.0
            END OF DATA BASE READ.
```

Figure 13: Part 1 of checking an FVS run is reading from FVS_StandInit as expected.

```
DESIGN      BASAL AREA FACTOR=    0.0; INVERSE OF FIXED PLOT AREA=    1.0; BREAK DBH=  999.0
            NUMBER OF PLOTS=     1; NON-STOCKABLE PLOTS=     0; STAND SAMPLING WEIGHT=
1.00000
            PROPORTION OF STAND CONSIDERED STOCKABLE=  1.000

********    FVS14 WARNING:  HABITAT/PLANT ASSOCIATION/ECOREGION CODE WAS NOT RECOGNIZED;
HABITAT/PLANT ASSOCIATION/ECOREGION SET TO DEFAULT CODE.

            PLANT ASSOCIATION CODE USED IN THIS PROJECTION IS CHS133

STDINFO     FOREST-LOCATION CODE=     612; HABITAT TYPE= 40; AGE=    0; ASPECT AZIMUTH IN
DEGREES=    0.; SLOPE=    5.%
            ELEVATION(100'S FEET)=  7.0;  REFERENCE CODE=

SPECIES       SF       WF       GF       AF       RF       SS       NF       YC       IC       ES
SDI MAX     450.     450.     450.     450.     450.     450.     450.     450.     450.     450.

SPECIES       LP       JP       SP       WP       PP       DF       RW       RC       WH       MH
SDI MAX     450.     450.     450.     450.     450.     450.     450.     450.     450.     450.

SPECIES       BM       RA       WA       PB       GC       AS       CW       WO       WJ       LL
SDI MAX     450.     450.     450.     450.     450.     450.     450.     450.     450.     450.

SPECIES       WB       KP       PY       DG       HT       CH       WI       __       OT
SDI MAX     450.     450.     450.     450.     450.     450.     450.     450.     450.
```

Figure 14: Part 2 of checking an FVS run is reading from FVS_StandInit as expected: SDI_{max}.

```
                                      NATIONAL VOLUME ESTIMATOR LIBRARY EQUATION NUMBERS
SPECIES CUBIC FOOT BOARD FOOT SPECIES CUBIC FOOT BOARD FOOT SPECIES CUBIC FOOT BOARD FOOT SPECIES CUBIC FOOT BOARD FOOT
------- ---------- ---------- ------- ---------- ---------- ------- ---------- ---------- ------- ---------- ----------
  SF    616BEHW011 616BEHW011   WF    616BEHW015 616BEHW015   GF    616BEHW017 616BEHW017   AF    616BEHW019 616BEHW019
  RF    616BEHW020 616BEHW020   SS    616BEHW098 616BEHW098   NF    616BEHW022 616BEHW022   YC    616BEHW042 616BEHW042
  IC    616BEHW081 616BEHW081   ES    616BEHW093 616BEHW093   LP    616BEHW108 616BEHW108   JP    616BEHW116 616BEHW116
  SP    616BEHW117 616BEHW117   WP    616BEHW119 616BEHW119   PP    616BEHW122 616BEHW122   DF    F00FW2W202 F00FW2W202
  RW    616BEHW211 616BEHW211   RC    616BEHW242 616BEHW242   WH    F03FW2W263 F03FW2W263   MH    616BEHW264 616BEHW264
  BM    616BEHW312 616BEHW312   RA    616BEHW351 616BEHW351   WA    616BEHW352 616BEHW352   PB    616BEHW375 616BEHW375
  GC    616BEHW431 616BEHW431   AS    616BEHW746 616BEHW746   CW    616BEHW747 616BEHW747   WO    616BEHW815 616BEHW815
  WJ    616BEHW064 616BEHW064   LL    616BEHW072 616BEHW072   WB    616BEHW101 616BEHW101   KP    616BEHW103 616BEHW103
  PY    616BEHW231 616BEHW231   DG    616BEHW492 616BEHW492   HT    616BEHW500 616BEHW500   CH    616BEHW768 616BEHW768
  WI    616BEHW920 616BEHW920   __    616BEHW000 616BEHW000   OT    616BEHW999 616BEHW999

SITECODE   SITE INDEX INFORMATION:
            SF=  203.;  WF=  140.;  GF=  140.;  AF=  203.;  RF=  140.;  SS=  140.;  NF=  203.;  YC=  203.; IC=  203.
            ES=  203.;  LP=  132.;  JP=  203.;  SP=  203.;  WP=  203.;  PP=  203.;  DF=  140.;  RW=  203.; RC=  140.
            WH=  140.;  MH=   28.;  BM=  152.;  RA=   65.;  WA=  132.;  PB=  304.;  GC=  142.;  AS=  152.; CW=  172.
            WO=  108.;  WJ=   47.;  LL=  140.;  WB=  142.;  KP=  203.;  PY=   51.;  DG=  122.;  HT=   51.; CH=  101.
            WI=  101.;  __=  203.;  OT=  203.
           SITE SPECIES=DF CODE=   16
```

Figure 15: Part 3 of checking an FVS run is reading from FVS_StandInit as expected: log rules and site index. In this case all species have defaulted to 16 foot logs (NVEL equation numbers starting with 616 indicate Region 6 followed by 16 for the log length—see documentation for the NVEL volume equation table). Note also that Sitka spruce, western hemlock, read alder, and western redcedar have all defaulted to the Douglas-fir site index.

4.4 Integrating FVS and Woodstock

Use of FVS output in Woodstock requires results from FVS be converted into Woodstock's yield file format. Among many ways of making this translation, one of the simplest is to download each FVS run's output table, merge the runs into a large Excel file (a few thousand rows is typical), save the merged result out of Excel as a text file (.txt), and then paste the result into Woodstock's yield section. This requires keeping track of each run's properties—species planting densities, site index used, and so on—to assign a Woodstock `*Y` keyword which integrates with the themes designated for the landscape.

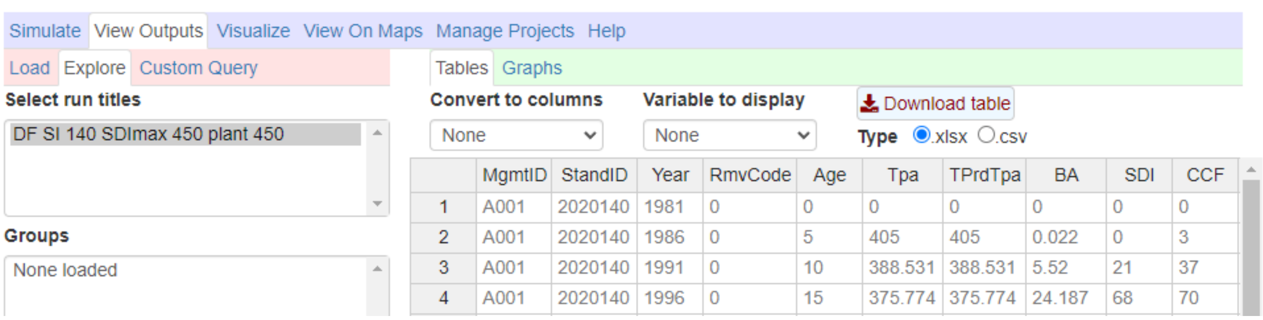


Figure 16: Location of the Download table button under View Outputs → Explore after loading the run's FVS_Summary2 table under View Outputs → Load.

In order to download a run's summary table, select the run under View Outputs → Load and then select FVS_Summary2 under Database tables to consider. Changing to the Explore tab should then show a preview of the run output along with the option to download the chosen table (Figure 16). In some cases, it can be helpful to include and configure multiple stands in a run, for example to simulate a range of productivities all at once, in which case the downloaded file will have a section for each stand.

	A	B	C	D	E	F	G	H	I	J	K
1	; stand age in years		5								
2	*Y ? ? ? ? ? ? DF siteClassIIIlow ?										
3	_AGE	yAgeInYears	yTPA	yBA	ySDI	yCCF	yTopHt	yQMD	yTCuFt	yMCuFt	yBdFt
4	0	0	0	0	0	0	0	0	0	0	0
5	1	5	329.000366	0.01847854	0	2	2	0.10147799	0	0	0
6	2	10	315.755585	4.92099762	18	31	13	1.69039154	35.3057404	0	0
7	3	15	305.185181	20.1822929	56	57	24	3.48208904	219.056992	0	0
8	4	20	298.933807	77.2620926	164	132	36	6.88386202	1064.5824	0	0
9	5	25	295.319824	149.359863	278	215	46	9.62957287	2478.87329	1869.16309	8476.31152

Figure 17: An example of a downloaded FVS table ready to save as a text file from Excel after 1) inserting Woodstock's *Y keyword to indicate the start of a yield table and which themes the yield table applies to, 2) unneeded FVS columns have been removed, and 3) column names have been changed to match Woodstock's _AGE keyword and yield naming convention. Note the five year _AGE divisor in cell C1.

When rearranging FVS output for use with Woodstock it's usually easiest to make the edits in Excel or another spreadsheet program. Some things to keep in mind are:

- Constraints on prescriptions require Woodstock themes to be implementable. The default set of themes described in the accompanying Python document supports some possibilities. However, decisions such as not employing a prescription on south facing slopes or restricting a prescription to higher elevations require themes communicating aspect and elevation to Woodstock.
- Since Excel saves one sheet in the workbook to a text file it's simplest to put all of the FVS growth and yield tables into the same sheet. That way there's only one text file to save and paste into Woodstock's yield section.[6]
- `_AGE` is a Woodstock keyword and must be used as such. Not `_age`, `Age`, `AGE`, or something else. Prefixing other yield names with `y` is not required by Woodstock but is recommended for clarity and consistency as it's a Woodstock coding convention.
- All ages in Woodstock, including `_AGE`, have units of planning periods. Since FVS outputs age in years and Woodstock planning periods are often longer than one year, it is likely the `_AGE` column in Excel will need to be divided by the planning period length. While the FE/FOR 459 lab materials use a five year planning period as a default, different groups may need different lengths depending on their objectives. And, as Woodstock model development proceeds, it may be desirable to change the planning period length. A simple way of accommodating variable period lengths is to create two columns and then calculate `_AGE` in periods from the age in years with a formula (Figure 17).
- Since FVS defaults to 10 year timesteps for a 100 year run, it's easy to accidentally generate FVS output which doesn't match the desired timestep or run length. Merging FVS outputs just after they're finalized provides an opportunity to check this and, if needed, change FVS's time settings and rerun.
- The tables need to be arranged as longform data, meaning there are many rows, few columns, and tables follow each other vertically in the spreadsheet (Figure 18). Placing tables side by side

[6] Multiple sheets and multiple text files work too but, the more complicated the data movement process, the greater the chance of errors.

(wideform data; horizontally with many columns and few rows) will not work as this arrangement is not supported by Woodstock. Woodstock also imposes a maximum line length of 250 characters so, if too many columns are kept from FVS, it's also possible even a single table can fail to be read. It's simplest to merge output from FVS runs first so that unneeded columns need only be deleted once as a final step and tables are more likely to stay consistent.[7]

- The `*Y` keyword lines need to match the landscape themes chosen for the Woodstock model. The masks therefore must have the same number and order of themes as the landscape section and shapefile. Similarly, the naming of theme attributes for species and productivity classes must be consistent across the landscape section, yield section, and shapefile. Care is important in this process since, for example, if a `*Y` is accidentally marked with `SiteClassIlow` instead of `SiteClassIIlow` it's quite difficult to detect the accidental decrease in productivity in Woodstock.
- While attaching comments or other notes to yield tables is often helpful, it's most efficient to set them up as Woodstock comments (indicated with a semicolon) during the mering process. If this is not done the comments will create errors in Woodstock and have to be changed to Woodstock comments anyways.

1320	40	200	66.4542694	581.287109	616	462	223	40.0470161	41049.9258	39139.9141	286962.844
1321											
1322	*Y ? ? ? ? ? ? DFWHRA siteClassImedium ?										
1323	_AGE	yAgeInYears	yTPA	yBA	ySDI	yCCF	yTopHt	yQMD	yTCuFt	yMCuFt	yBdFt
1324	0	0	0	0	0	0	0	0	0	0	0
1325	1	5	415.000916	0.02324586	0	4	3	0.10134084	0	0	0
1363	39	195	68.3937225	590.716675	628	469	224	39.7940102	41785.9648	39836.875	291956.375
1364	40	200	66.4130173	590.810303	624	468	225	40.386261	42040.168	40093.4219	294403.719
1365											
1366	*Y ? ? ? ? ? ? DFWHRA siteClassIhigh ?										
1367	_AGE	yAgeInYears	yTPA	yBA	ySDI	yCCF	yTopHt	yQMD	yTCuFt	yMCuFt	yBdFt
1368	0	0	0	0	0	0	0	0	0	0	0
1369	1	5	415.000916	0.02324586	0	4	3	0.10134084	0	0	0
1408	40	200	66.4410324	594.513428	627	470	227	40.5041008	42625.5664	40623.3242	298610.313
1409											
1410	*Y ? ? ? ? ? ? RA siteClassIIIlow ?										
1411	_AGE	yAgeInYears	yTPA	yBA	ySDI	yCCF	yTopHt	yQMD	yTCuFt	yMCuFt	yBdFt
1412	0	0	0	0	0	0	0	0	0	0	0
1413	1	5	590.00061	0.03312631	0	10	3	0.1014604	0	0	0

Figure 18: An example arrangement of yield tables as longform data. Each table must have its own `*Y` and set of yield headers (`_AGE`, `yTpa`, and so on). Empty lines between tables are not required but are recommended for readability. Note that rows for ages of 10–195 years are hidden here for brevity.

4.5 A note on other FVS options and keywords

FVS's keyword system is much larger than the two examples noted in the previous sections. Many additional keywords can be added to FVS runs using the options under Components to specify other management actions besides planting, tweak growth rates and density constraints, consider disease and climate change effects, and so on. Depending on management objectives, it may be helpful to make use of some of these tools to develop silvicultural prescriptions which better support the objectives. FVS

[7] It's a good idea to make a backup copy of the .xlsx before deleting columns. This minimizes the amount of work if another column turns out to be needed later.

generally inserts keywords following whatever points are selected in the Run Contents and, in some case, out of order keywords can result in run errors. If this occurs, the Cut/Delete and Paste item buttons can be used reposition keywords.

Some areas which are likely to be of interest are

- Components → Management → Categories = Thinning & Pruning Operations. It is not necessary to download runs where thinning is performed for use in Woodstock—the `SHIFT` keyword can be used in Woodstock instead—but simulating thinning in FVS is helpful in determining how many years to `SHIFT` a stand by.
- Components → Keywords → Extensions = Base FVS system. The keywords available here allow detailed tuning of FVS and are mostly beyond the scope of this course. However, in older western hemlock and western redcedar stands FVS fails to enforce SDI_{max}. FixMort can correct much of misprediction by forcing the stand's density trajectory into SDI_{max} compliance. Also, BFVolume, VolEqNum, and Volume are helpful in adjusting merchantable volume and SiteCode allows site index to vary by species.

5 Retaining work in FVS

For each project that's created, FVS creates a directory with that project's name under C:\FVS. FVS project files therefore will not roam between lab machines and will be destroyed when a lab machine is reimaged. Reimaging most commonly occurs between quarters so, in most cases, FVS projects should persist on the lab machine where they were created through the end of a quarter. However, this is not guaranteed as machines may need to be reimaged during a quarter and server mirror space limitations can prompt a reset of a machine's local files. FVS unfortunately lacks the ability to place projects outside of C:\FVS and also lacks the ability to export a project from one machine for import it on another. However, moving projects from one machine to another is possible with a multi-step process.

1. In the project to be moved or saved, go to Manage Projects → Downloads and export the Input database and FVSProjectData.zip. Copy these files from your downloads to location you want to keep them in (OneDrive, T:, Z:, USB drive, *etc*.).
2. Start FVS on the machine where you want to recreate the project. Go to Manage Projects → Manage Project, make a new project, and open that new project.
3. In the new project, go to Manage Projects → Import input data → Upload FVS-Ready database and browse to the input database you exported. FVS will load this database ask what to do with it. Since a new project is being populated, choose Install uploaded database.
4. Change to Manage Projects → Import runs and other items. Browse to the FVSProjectData.zip you downloaded earlier. For each run that's of interest, select it as a run to import and click the Import button.
5. If you switch to the Simulate tab at this point the imported runs should be listed with all their keywords and be runnable.

6 Troubleshooting

Common causes of problems with FVS include

1. Configuration errors leading to the FVS_StandInit table not being read. These tend to result in the STANDSQL information not appearing in the main output file and use of default site index and SDI_{max} values. Typically this signals a configuration error with the FVS_GroupAddFilesAndKeywords or FVS_StandInit tables, often due to the stand not matching a group listed in FVS_GroupAddFilesAndKeywords. If the Kwd: From: FVS_GroupAddFilesAndKeywords keyword isn't included in the run contents the stand will need to be removed from the run and reinserted after correcting the error.
2. There being too many or two few trees, resulting in unrealistic basal areas and volumes. This can result from simulations running with values of SDI_{max} which don't match the ground being simulated. Choices of BASAL_AREA_FACTOR, INV_PLOT_SIZE, and BRK_DBH in FVS_StandInit are important here, even though there are no plot trees in FVS_TreeInit and all trees are planted. (Mismatches between the number of plots indicated in FVS_StandInit and the number of distinct plots listed in FVS_TreeInit can also cause errors.)
3. Getting identical stand trajectories across runs with different site indices. This is usually an issue with FVS_StandInit not being read or not being set up correctly, resulting in FVS either using an unintended site index or falling back to a default site index. Checking the site index section of main output file can confirm and diagnose the issue. However, FVS can claim it's reading StandInit correctly even when it's clearly not.
4. Problems with unrealistic results for some species, especially ones besides Douglas-fir. Often this is due to not specifying species appropriate site indices and SDI_{max} values for the site species. In multi-species stands it can also be the stand's SDI_{max} is inappropriate for the species mixture or that the SDIMax keyword isn't fully adjusting a particular species' SDI_{max}. In the case of FVS-PN (and some other variants) this can be due to density dependent mortality becoming decoupled from SDI_{max}.

Additional Resources

Crookston NL. 2014. Climate-FVS Version 2: Content, users guide, applications, and behavior. Rocky Mountain Research Station GTR-319, US Forest Service. https://doi.org/10.2737/RMRS-GTR-319.

Diaz DD, Loreno S, Ettl GJ, Davies B. 2018. Tradeoffs in Timber, Carbon, and Cash Flow under Alternative Management Systems for Douglas-Fir in the Pacific Northwest. Forests 9(8):447. https://doi.org/10.3390/f9080447 (FVS configuration is described in the supplementary material.)

Heiderman RR, Kimsey Jr MJ. 2021. A species-specific, site-sensitive maximum stand density index model for Pacific Northwest conifer forests. Canadian Journal of Forest Research 51(8):1166–1177. https://doi.org/10.1139/cjfr-2020-0426

Kayahara GJ, Carter RE, Klinka K. 1994. Site index of western hemlock (*Tsuga heterophylla*) in relation to soil nutrient and foliar chemical measures. Forest Ecology and Management 74:161-169. https://doi.org/10.1016/0378-1127(94)03493-G

Klinka K, Kayahara GJ, Chourmouzis C. 1997. Relationship of site index to estimates of soil moisture and nutrients for western redcedar in south coastal British Columbia. Scientia Silvica Extension Series 5, Forest Sciences Department, University of British Columbia. https://doi.org/10.14288/1.0107289

Puettmann KJ, DeBell DS, Hibbs DE. 1993. Density management guide for red alder. Research Contribution 2, Forest Research Laboratory, Oregon State University. https://ir.library.oregonstate.edu/concern/technical_reports/5m60qt10v

McArdle RE, Meyer WH. 1931. The yield of Douglas-fir in the Pacific Northwest. Technical Bulletin 201, United States Department of Agriculture. https://handle.nal.usda.gov/10113/CAT86200195

Weiskittel AR, Hann DW, Hibbs DE, Lam TY, Bluhm AA. 2009. Modeling top height growth of red alder plantations. Forest Ecology and Management 258(30):323–331. https://doi.org/10.1016/j.foreco.2009.04.029

Creating Silvicultural Themes for Woodstock in GIS

1 Introduction

Growth and yield predictions using FVS require both stand level and inventory inputs. Stand level inputs are site properties, such as area, site index, and SDI_{max}. Inventory describes the existing forest through measurements like TPA and basal area. Each individual stand has a unique combination of these attributes. However, because ODF's West Oregon District has over 900 stands, it would take more effort than fits in this course to develop and customize a range of silvicultural prescriptions for each individual stand. An alternative speed-accuracy tradeoff is to group together stands with similar productivity and stocking, representing each group with a common yield curve. This approach saves effort by approximating many stands' growth and yield with a single FVS prediction.

This document introduces a few among many methods of grouping stands, using Python in either ArcGIS Pro or ArcMap, to create fields (*i.e*. columns) in a shapefile which classify stands into groups for FVS simulation on the basis of species composition, site productivity, and stocking level. As noted in the GIS assignment, fields which correspond to landscape themes in Woodstock must be text fields named Theme1, Theme2, Theme3, and so on so that Woodstock is able to connect to them. Additionally, Woodstock requires Theme fields appear in sequential order in the attribute table.

2 Setup

Copy your West Oregon District shapefile with stand, county, murrelet, and riparian data to a new working directory for this lab. It is suggested not to use spaces in the file name as they'll have to be removed when placing the shapefile into a Woodstock project.[1]

This shapefile from the GIS lab already has county, ownership, murrelet, riparian, stand, and desired future condition information in its Theme1–6 fields. Start ArcGIS Pro or ArcMap, whichever you're working in, and add your copy of the shapefile as a layer.

3 Species classification using Python (ArcMap or ArcGIS Pro)

Many methods can be applied to classify tree species composition in forest stands. The approach illustrated here is a simple one using basal area. The SLI_ROOTS_OneLiner table in ODF's inventory data contains basal areas for Douglas-fir (df_ba), red alder (ra_ba), western hemlock (wh_ba), and all other species combined (ba_sp4). If a single species is more than 80% of a stand's basal area the stand's often classified as a single species stand. Since the basal area fraction of tree species *i* is

$$species\ fraction_i = \frac{BA_i}{df\_ba + ra\_ba\ + wh\_ba\ + sp4\_ba}$$

a stand with any species fraction greater than 0.8 is therefore considered single species. Otherwise, it's a mixed species stand. For example, the fraction of Douglas-fir is $\mathrm{df\_ba}/(\mathrm{df\_ba} + \mathrm{ra\_ba} + \mathrm{wh\_ba} + \mathrm{sp4\_ba})$. It is also necessary to check for unstocked stands with zero total basal area and choose some default

[1] This is a regression in Woodstock 2023's .maps file parsing. Woodstock 2022 can handle spaces in GIS file names.

species assignment—otherwise species fractions become `NaN` due to dividing by zero and numerical comparisons are generally false.

3.1 Getting started

Add a text field to the shapefile named Theme7. Right click on Theme7's field header and select Field Calculator. As a reminder from the GIS lab, to run Python code in

- ArcMap: In the Field Calculator window check the Show Codeblock option, which will change the display to Pre-Logic Script Code. Copy and paste the example code below into the Pre-Logic Script Code box. (ArcMap's default length of 50 should be more than sufficient.)
- ArcGIS Pro: Copy and paste the example code below into the Code Block.

```
def getSpeciesClass(df, ra, wh, sp4):
  totalBA = df + ra + wh + sp4
  if totalBA == 0:
    return "DF"
  elif df / totalBA > 0.8:
    return "DF"
  elif ra / totalBA > 0.8:
    return "RA"
  elif df / totalBA > 0.5 and ra / totalBA > 0.3: # mixed DF-RA stands
    return "DFRA"
  elif df / totalBA > 0.5 and wh / totalBA > 0.3: # mixed DF-WH stands
    return "RAWH"
  else:
    return "DFWHRA"
```

Next, in the Theme7 = box (it's the same in both ArcGIS Pro and ArcMap), enter

```
getSpeciesClass(!df_ba!, !ra_ba!, !wh_ba!, !ba_sp4!)
```

Click OK to run the `getSpeciesClass()` function. The Theme7 field should populate with DF, DFWHRA, and so on. The 80%, 50%, and 30% thresholds (`0.8`, `0.5`, and `0.3` in the code above), along with the Douglas-fir default are somewhat arbitrary and should be adjusted if doing so better supports management objectives.

3.2 Brief Notes on Python

Python is a commonly used programming language (by some assessments the commonly used language worldwide) and is frequently used for automating or customizing GIS tasks. A wide range of supporting documentation and tutorials is available from python.org and, more specifically, for ArcGIS Pro and ArcMap. It's beyond the scope of this document to describe all of this material but, if you haven't previously coded Python, the list below covers a few key things to be aware of.

- In both ArcGIS Pro and ArcMap, an attribute table field is accessed from Python by placing it between exclamation marks in the function call.
- If errors are encountered during processing, checking the resulting messages from ArcGIS Pro (in the error message popup) or ArcMap (Geoprocessing > Results) frequently offers some insight into what the trouble is.
- Python code is case sensitive. For example, `totalBA` and `totalBa` are different variables.
- The indentation at the start of lines is part of Python's syntax (spaces are preferred). Changing indentation therefore changes the meaning of the code, even if the change is an invisible one between tabs and spaces.
- The operators for comparing two variables (*i.e*. a species' basal area and the stand's total basal area) are `>` (greater than), `<` (less than), `>=` (greater than or equal), `<=` (less than or equal), and `==` (equals). Note the equals operator is two equals signs in a row (`==`) rather than one (`=`).
- Assignment of a value is done using `=`.
- A comment in Python starts with `#`.

If you're unfamiliar with coding it may be best most efficient to make small, incremental changes and test them before moving on to the next change. (Consider also maintaining a copy of your last known good script in Notepad++ or another program separate from ArcMap or ArcGIS Pro.)

3.3 Understanding what the code does

The first line defines `getSpeciesClass()` as a function which takes four arguments, one for each tree specie's or species group's basal area.

```
def getSpeciesClass(df, ra, wh, sp4):
```

This function is called (invoked) once for each row of the attribute table with its arguments set to the field values on that row, hence the need to include `getSpeciesClass(!df_ba!, !ra_ba!, !wh_ba!, !ba_sp4!)` to tell ArcMap or ArcGIS Pro how to pass the fields to `getSpeciesClass()`. To make the code easily readable, it makes sense to name the variables as `df`, `ra`, `wh`, and `sp4`, where `df` corresponds to DF_BA in the row of the attribute table, ra to RA_BA in the same row, and so on.

The body if the function is an if-elif-else clause (`elif` is short for else if). The initial if statement handles the zero basal area case with a return statement which defaults zero basal area stands to Douglas-fir. Since the line of code which sets the `totalBA` variable isn't an if-elif-else case it doesn't end with a colon.

```
totalBA = df + ra + wh + sp4
if totalBA == 0:
   return "DF"
```

This case is important for two reasons. First, it assigns a default species to stands which do not have inventory information, either because they've just been clearcut or because they haven't been inventoried (inv = 'NONE' in SLI_ROOTS_OneLiner, with age = 0 for recent clearcuts or with a larger value of age for stands which have not been inventoried). Second, this check avoids divide by zero errors in the following cases.

The two following `elif` statements first check to see if Douglas-fir's basal area fraction is large enough to classify the stand as Douglas-fir only. If this is not the case then red alder's fraction is checked.

```
elif df / totalBA > 0.8:
      return "DF"
elif ra / totalBA > 0.8:
      return "RA"
```

If a stand does not qualify as a single species, then the minimum fraction for including a species must be determined. Suppose you've decided it's justifiable to include only species with fractions above 30%. Furthermore, to show that one species is predominant, it's preferable to apply a higher threshold and place it first in the class listing. Let's say you've also determined a species is predominant within a mixed species stand if its weight is more than 50%. Therefore, a two species stand will have to have a species with a basal area fraction greater than 50% and a second species with more than 30%. This leads to the next two cases

```
elif df / totalBA > 0.5 and ra / BA > 0.3: # mixed stands of DF and RA
   return "DFRA"
elif df / totalBA > 0.5 and wh / BA > 0.3: # mixed stands of DF and RA
   return "DFWH"
```

The `else` statement catches all species compositions which don't match the `if` or any of the `elif` cases:

```
else:
   return "DFWHRA"
```

4 Site productivity classification using Python

Site productivity can be classified using an approach similar to species classification. For example, the function below could be called as `getSiteClass(!site_ind!)` to populate a Theme8 field in the shapefile.

```
# site classification from 50-year Douglas-fir site index
# Follows DeYoung (2016).
def getSiteClass(siteIndex):
  if siteIndex >= 135:
    return "siteClassI"
  elif siteIndex >= 115:
    return "siteClassII"
  elif siteIndex >= 95:
    return "siteClassIII"
  elif siteIndex >= 75:
    return "siteClassIV"
  else:
```

```
        return "siteClassV"
```

However, naïvely applying this script would result in poor classification performance as stands in the West Oregon District are mostly site class I and II. A more accurate approach is to choose a set of productivity classes which well approximate the District's site indices, starting from area weighted quantiles (Figure 1) or some comparable method such as natural breaks classification in GIS (Symbology > Graduated colors in ArcGIS Pro, Symbology > Quantities > Graduated colors in ArcMap).

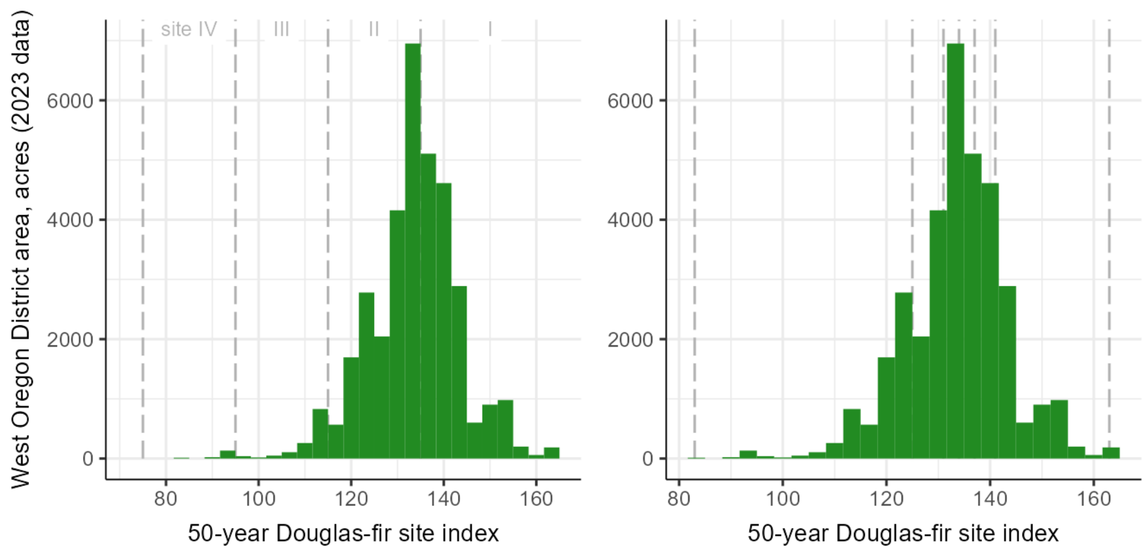


Figure 1: Distribution of site productivity on the West Oregon District broken by site class (left) or seven equal area quantiles (right). The mid-quantile site indices are 120, 127, 131, 134, 137, 140, and 147 feet.

With these considerations in mind, `getSiteClass()` can be revised to perform a more granular classification. Depending on the number of productivity classes chosen and how wide the classes are, the resulting code might look something like

```
def getSiteClass(siteIndex):
  if siteIndex >= 151:
    return "siteClassIhigh"
  elif siteIndex >= 143:
    return "siteClassImedium"
  elif siteIndex >= 139:
    return "siteClassImediumLow"
  elif siteIndex >= 135:
    return "siteClassIlow"
  elif siteIndex >= 126:
    return "siteClassIIhigh"
  elif siteIndex >= 115:
    return "siteClassIIlow"
  elif siteIndex >= 101:
    return "siteClassIIIhigh"
  else:
```

```
        return "siteClassIIIlow"
```

FVS simulations for stands with Douglas-fir as the site species could then use site indices in the middle of these ranges. For stands with other species, the site index would be translated from the Douglas-fir index.

5 Stand thinning status and density classification

Stocking levels vary widely across the West Oregon District (Figure 2) due to choices of planting density, mortality, and thinning. Representing all stands within a given species and productivity classification with a single stand trajectory starting from a given level of planting survival can therefore introduce substantial classification error. ODF handles this by including H in the vegetation labels of highly stocked stands and L for stands with low stocking. Stands may have low stocking for a range of reasons, the most common of which is thinning. As will be seen in the next assignment, Woodstock requires a theme to keep .

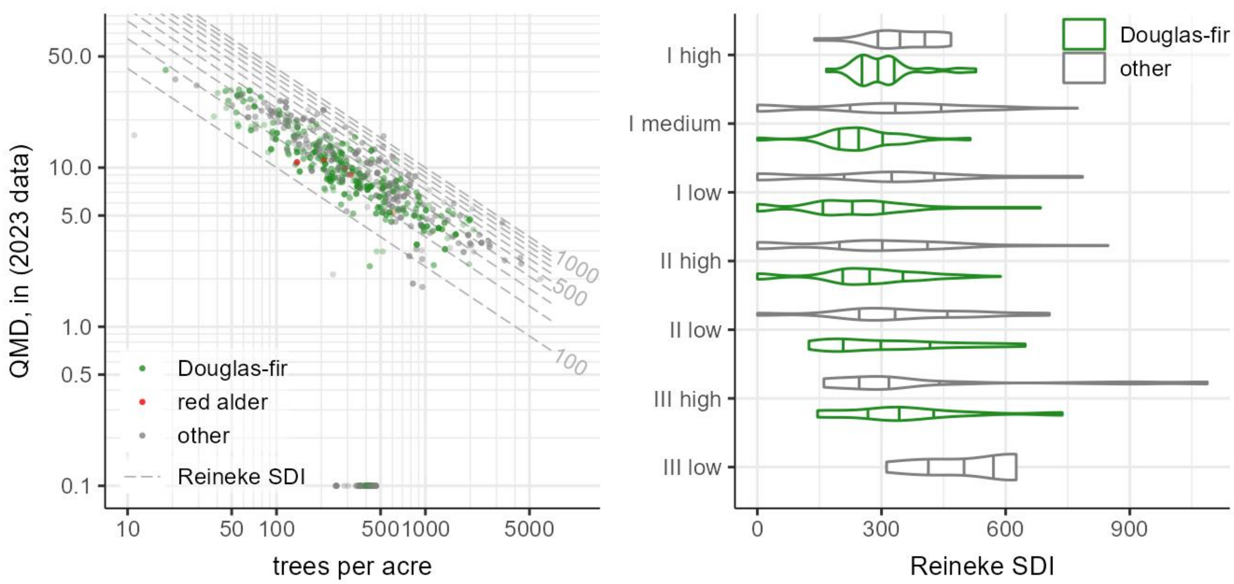


Figure 2: Stand density distribution on the West Oregon District as measured by ODF inventory data (left) and violin plots from an example classification by stand species and site productivity (right).

Create a Theme9 field in the shapefile and, for now, use the field calculator to default all of its values to unthinned. It will likely be desirable to analyze ODF’s trees per acre or basal area inventory data, possibly by calculating Reineke SDI or other metrics, to further classify this stocking or stand density theme. However, the choice of classes depends on both the silvicultural prescriptions chosen in this assignment and the implementation of Woodstock thinning actions which are discussed in the next lab.

6 Tasks

Once you understand how classify stand species and site productivity, modify this document's example code to have appropriately configured `elif` statements for the species, productivity classes, and stocking densities aligned to modeling your group's objectives.

1. Once species, site, and (potentially) stocking or thinning classes are defined you can proceed to constructing corresponding stand growth and yield predictions in FVS.
2. Remember that classifying existing stock on the District is distinct from future stocking. If your silvicultural plans call for the creation of stand types not currently present on the District, FVS predictions will be needed for those stands as well as for existing stands.

Both the shapefile with species and site themes added and the growth and yield tables generated by FVS will be loaded into a Woodstock model of the West Oregon District in the next assignment.

References

DeYoung, J. 2016. Forest Measurements: An Applied Approach. Open Oregon Educational Resources. https://openoregon.pressbooks.pub/forestmeasurements/

S1 Supplementary Material: Classification in QGIS

S1.1 Getting started

Open the layer's attribute table, enter edit mode, and launch the Field Calculator. Create a new text field named Theme5 with a length of at least 10. Copy and paste the code below into the expression box

```
with_variable('totalBA', "df_ba" + "ra_ba" + "wh_ba" + "ba_sp4",
  case
  when @totalBA = 0 then 'DF'
  when "df_ba" / @totalBA > 0.8 then 'DF'
  when "ra_ba" / @totalBA > 0.8 then 'RA'
  when "df_ba" / @totalBA > 0.5 and "ra_ba" / @totalBA > 0.3 then 'DFRA'
  when "df_ba" / @totalBA > 0.5 and "wh_ba" / @totalBA > 0.3 then 'DFWH'
  else 'DFRAWH'
  end
)
```

Press OK to run the Field Calculator and the Theme7 column should populate with DF, RA, DFRA, and so on. If you're satisfied with the changes, save them and end the edit session. If you need to recalculate the classification, bring up the Field Calculator and, instead of creating a new field, choose Update existing field.

Similarly, for site productivity,

```
case
when "site_ind" >= 151 then 'SiteClassIhigh'
when "site_ind" >= 143 then 'SiteClassImedium'
```

```
when "site_ind" >= 139 then 'SiteClassImediumLow'
when "site_ind" >= 135 then 'SiteClassIlow'
when "site_ind" >= 126 then 'SiteClassIIhigh'
when "site_ind" >= 115 then 'SiteClassIIlow'
when "site_ind" >= 101 then 'SiteClassIIIhigh'
else 'SiteClassIIIlow'
end
```

S2.2 Understanding the code

The logic and classification thresholds here are the same as with Python in ArcGIS Pro and ArcMap but the syntax changes to that used by QGIS's field calculator. `with_variable()` defines `@totalBA` as a variable and sets its value to the stand's basal area. The case-when-else statements function identically to Python if-elif-else statements, just with different naming. The other change is that, while single and double quoted strings function identically in Python (*e.g.* `'DF'` and `"DF"`, with the type of quotes often becoming a stylistic convention signaling how the string is used), in QGIS's field calculator all strings are single quoted and double quotes indicate an attribute table field.

Model Development Guide for Woodstock Optimization Suite

1 Introduction

This document provides a workflow for setting up an initial model for creation of strategic forest management plans on ODF's West Oregon District using Woodstock Optimization Suite 2023.02. A base Woodstock model consists of 14 text files, 11 of which need to be set up to run Woodstock, two of which are generated by Woodstock, and one file which does not need to be used in this course. Woodstock refers to each of these 13 files as a section and gives each file an extension similar to its name (Figure 1).

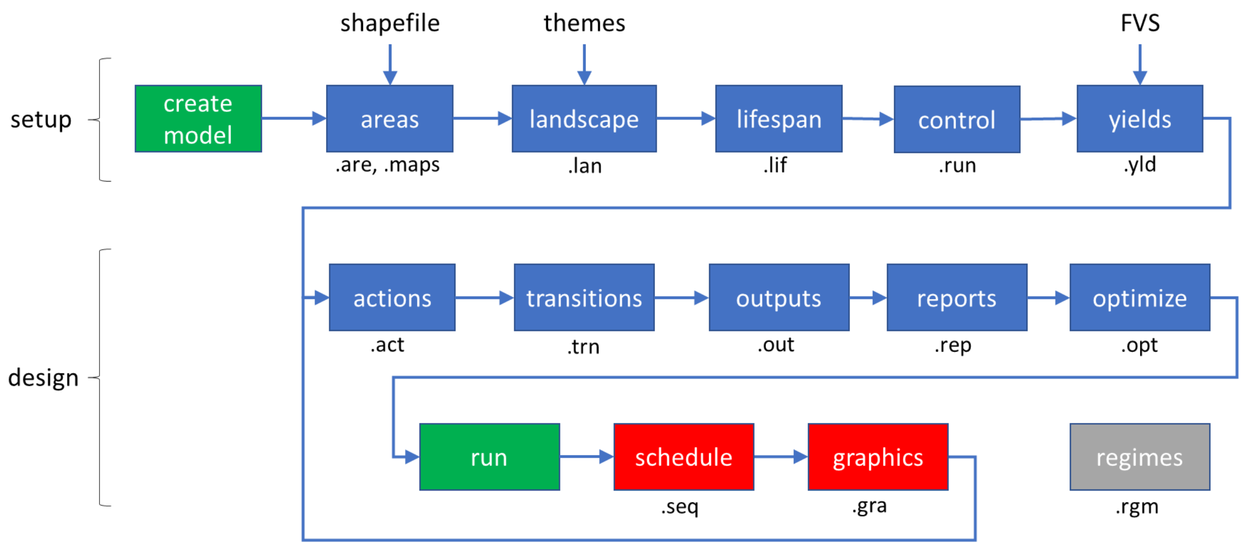


Figure 1: Suggested workflow for creating and configuring a Woodstock model. Section 2's subsections follow this order.

As a reminder from FE/FOR 457, searchable documentation and guidance on Woodstock sections and the keywords they contain is online. Since the documentation requires a Woodstock license, it is accessed within Woodstock by choosing Help → Documentation → Remsoft Documentation System from Woodstock' menu. It is nearly certain substantial time will need to be spent with the Woodstock documentation in order to understand how to write the yield sections, actions, transitions, outputs, and optimization objectives needed to communicate your group's management objectives to Woodstock.

2 Base Woodstock Model Setup: Initial Population of Component Files

To begin, start Woodstock, from either the desktop icon or the start menu (on a lab machine), and choose File → New → New model.

1. Give the model a name and choose a location to save it. Initially, the model's files are likely to be around 30 MB with eventual growth towards 250 MB or more. Since Woodstock writes large working files to disk while running linear solvers use of OneDrive is therefore suggested for

performance, though T:\Teach\Temp\FE459_FOR459\<your group> typically also produces acceptable runtimes.[1]

2. Leave all of the sections checked and press OK to create the model.
3. The Model View pane at left will populate with the sections just created. The Allocation, Constants, LpSchedule, Queue, and Maps sections will be crossed off as those files haven't been created.

2.1 Spatial setup from maps: areas and landscape

Maps read by Woodstock have, historically, been shapefiles.[2] To add a map to Woodstock and use it to populate the areas and landscape sections,

1. Copy the shapefile you want to use (either from the FVS assignment or some later update) into the folder where the Woodstock model is located.
2. Double click on Spatial Setup in the Model View pane. Choose the location of your shapefile and then select the shapefile's name in the Primary dataset(s) dropdown.
3. Scroll down to AGE1 as the Age field name and set ACRES as the Area field name. Set the age divisor to the planning period length in years and the area divisor to NONE. The ignore polygons options and reference layers can be left unset. Click Save to load the shapefile.
4. The Maps section should uncross and, if you double click it, you should be able to view the shapefile's contents by selecting different fields in the Quick Legend pane.
5. Fields which Woodstock links to are shown in green in the Quick Legend. These should be Age1 and Acres along with Theme1, Theme2, Theme3, and so on. Clicking through these should cause the map Woodstock displays to change through counties, ownership, murrelet areas, and the other themes. However, Woodstock may not display anything.
6. Even if nothing's shown, right click somewhere in the map view and choose Build Section → Areas Section. Change Destination to Overwrite existing Areas Section and click Build. Woodstock should respond by switching to the now populated areas section. The areas section should not have more entries than there are polygons in the shapefile. (There may be fewer areas if Woodstock merged some of them but the total area reported in the areas section's header should match the total of the ACRES column in GIS.)
7. Go back to the maps view and right click again for Build Section → Landscape Section. Change Destination to Overwrite existing Landscape Section and click Build. Woodstock should switch to the landscape section, which should have a *THEME keyword listing the unique values for each theme's attribute present in the shapefile.

After the landscape section is generated, descriptions can be added for the themes and their attribute values. This can be helpful for keeping track of what various attribute codes mean (Figure 2). Also useful is `*AGGREGATE`, which creates a name for a group of attribute codes which can then be used

[1] USB sticks can also work but, depending on the performance of the stick and the USB port it's plugged into, can slow Woodstock noticeably.
[2] Woodstock 2023 adds support for ESRI geodatabases (.gdb).

in the attribute masks of actions and transitions. Such use of aggregates simplifies constraining where management actions can occur and can make writing transitions and yields easier.

8. Add the `canClearcut` aggregate to the desired future condition theme (Figure 3). This aggregate will enforce ODF's policy of avoiding clearcutting stands which have multilayered or old forest structure or are intended to develop structural complexity.

```
*THEME {1} county
 Benton
 Lincoln
 Polk

*THEME {2} owner
 AdminSite special stewardship areas
 BOF Board of Forestry
 CSL Common School Fund lands

*THEME {3} wildlife
 murrelets known marbled murrelet management areas
 noListedSpecies not specially managed for threatened or endangered species

*THEME {4} slope position
 riparian stream buffers
 upland harvestable stands
```

Figure 2: Example of the first few themes in a landscape section with descriptions added.

```
*THEME {6} desired future condition (FMP_NW_RevisedApril2010_Combined.pdf)
 --- DFC not designated
 GEN intensively managed plantations
 LYR layered (understory reinitiation)
 OFS old forest structure
 *AGGREGATE canClearcut
  --- GEN
```

Figure 3: Use of `*AGGREGATE` to define areas which can and cannot be clearcut. (Depending on choices made in the FVS assignment, desired future condition might not theme six.)

2.2 Procedural sections: lifespan and control

The lifespan section establishes the maximum age a management polygon may reach before it will be considered dead, after which Woodstock's builtin `_DEATH` action restarts the area from age zero. With long-lived tree species (such as Douglas-fir, western hemlock, and western redcedar), natural regeneration, and protection of old growth stands there is no age where this Woodstock behavior is realistic. Therefore, a reasonable default is to set the maximum lifespan to a value beyond the oldest stand age at the end of the planning horizon so that the `_DEATH` action never triggers (Figure 4). Also, since model size and solve times increase with longer lifespans, Remsoft's guidance is not to set a lifespan much greater than needed.

The Control section tells Woodstock model how to process input files and, most importantly, how many planning periods there are within the planning horizon. Woodstock does not know how long a planning period is, so the length of a planning period and number of periods is something the user has to keep track of. Ten five-year periods are a reasonable default for this course (but see also Section 5.2). It is also good to turn on warnings (Figure 5) as Woodstock then displays a warnings file when it's able to detect issues with a model.

```
;Lifespan
? ? ? ? ? ? ? ? ? 60 ; maximum stand age before _DEATH = 60 5 year planning
periods = 300 years (Woodstock max is 1000 periods)
; 201 year old trees in inventory + 50 years = max age of 251 years = 51 5
year periods
```

Figure 4: A default lifespan section for a 50 year planning horizon on the West Oregon District. If there are less than nine or more than nine landscape themes the number of question marks in the mask will need to be adjusted to match (this holds for all masks in Woodstock).

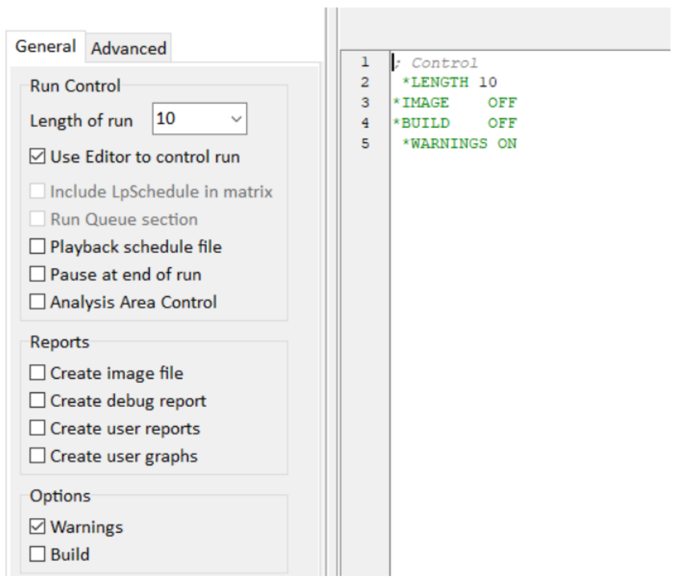


Figure 5: Control section adjusted from Woodstock's default settings.

2.3 Yields: specifying forest growth

To transfer the yield table spreadsheet created in the FVS assignment to Woodstock,

a. Open the spreadsheet in Excel and save it as a tab delimited file (File → Save As and select Text (Tab Delimited) (*.txt) in the dropdown).
b. Open the resulting tab delimited file in Notepad++ or a similar tool and copy and paste its contents into Woodstock's yield section.[3]

[3] If this results in the yield tables pasting into Woodstock as one long line it's likely the text file was saved from Excel using carriage returns (CR) for newlines rather than carriage returns plus linefeeds (CR LF). An easy way to check is open the text file in Notepad++ and look at the lower right part of the status bar, which will say either Mac (CR) or Windows (CR LF). If Mac (CR) is shown saving the text file from Excel on a Windows machine should fix the issue.

In addition to time-invariant yield tables with the `*Y` keyword, Woodstock offers `*YC` and `*YT` for complex and time-dependent yields. Two complex yields are useful as starting points.

1. Add MBF and MMBF yields to the end of the yields section (Figure 6). Depending on management approaches and silvicultural systems it may be appropriate to choose a different deduction for defect and breakage in `yMBFrecovered`.

```
; 1000 standing BF/MBF / (100% - 5% defect and breakage) => 1053 standing BF harvested/pond delivered MBF
*YC ? ? ? ? ? ? ? ? ? ?
yMBFrecovered _DIVIDE(yBdFt, 1053)

*YC ? ? ? ? ? ? ? ? ? ?
yMMBFstanding _DIVIDE(yBdFt, 1000000)
```

Figure 6: Complex yields (`*YC`s) for calculating net harvested MBF and standing MMBF.

2.4 Actions and transitions: enabling Woodstock to schedule harvests and reforestation

In order for Woodstock to plan land management actions it needs to know what actions can be taken and how those actions modify landscape themes. For every action that's defined, Woodstock requires a corresponding transition indicating whether the action modifies the themes of any of the areas it is performed on. Woodstock models therefore have a series of `*ACTION` statements in the action section, each of which has a matching sequence of `*CASE` statements in the transitions section. The two are linked together by the name of the action which, by convention, starts with `a`.

1. Create a clearcut action in in the actions section (Figure 7). If the themes are in a different order the attribute mask will need corresponding adjustment to be valid (when a mask is valid Woodstock underlines it).
2. Create a transition for the clearcut action in the transitions section (Figure 8). Since this transition doesn't modify any themes it just passes land through the clearcut and reforestation process.

```
;Actions
*ACTION aClearcut Y clearcut and reforest
*OPERABLE aClearcut
? ? noListedSpecies upland ? canClearcut ? ? ? _AGE >= 10 and _AGE <= 30 and yQMD >= 15
```

Figure 7: A basic clearcut action named `aClearcut`. The Y following `aClearcut` indicates the action resets the management area's stand age and the attribute mask in `*OPERABLE` indicates what areas Woodstock can use the action on.

There are a few things which may not be obvious in how `aClearcut` is specified.

- Since clearcuts can be performed in any county, land ownership, or stand classification those themes have `?` in the mask so that no restrictions are applied.

- Constraining aClearcut to noListedSpecies and upland areas means murrelet management areas and riparian buffers will not be clearcut.
- The 50 year lower bound on _AGE and the quadratic mean diameter constraint follow from ODF policy (see ODF_Capstone_Constraints.pdf).
- Placing an upper bound on _AGE is not required but, where such restrictions are practical, they are recommended by Remsoft as they simplify the linear program and make it faster to solve. The 150 year stand age limit in this example is somewhat arbitrary but it's reasonable to limit clearcutting to a sociopolitically acceptable age range.

```
;Transitions
*CASE aClearcut
*SOURCE ? ? ? ? ? ? ? ? ?
*TARGET ? ? ? ? ? ? ? ? ? 100
```

Figure 8: The passthrough transition for aClearcut. The unrestricted *SOURCE and *TARGET masks indicate the transition applies to all land areas aClearcut is scheduled on. The 100 at the end of the *TARGET statement means the *CASE applies to 100% of the clearcut area.

2.5 Outputs, reports, and graphics: tracking the management Woodstock chooses

Since Woodstock does not automatically report the management actions it chooses, it's necessary to ask Woodstock to output measurements of interest in the outputs section. These outputs are then plotted as specified in Woodstock's graphics section, which is usually configured graphically rather than edited as a text file, and are written to log files as specified in the reports section.

1. Add a few initial outputs to the outputs section for monitoring inventory levels and harvest rates (Figure 9).
2. Change the default report format from a text file to a .xlsx file for use with Excel by changing the *TARGET file name's extension (Figure 10).
3. Graphics are most easily configured once the model is running, so they're handled in the following section.

In the inventory group of Figure 9, _TH1 and _TH2 indicate those outputs should be broken out by themes 1 and 2, which are county and owner, respectively.

Since Woodstock outputs only two decimal digits in reports it can be useful to produce the same output in different units or to use scale option in Woodstock Graphics. For example, a small MMBF output might plot acceptably within Woodstock Graphics but be too rounded in a .xlsx file to plot well in Excel when making figures for reports or presentations. If the output's instead expressed in MBF three more digits of precision are available, likely avoiding .csv limitations. In Woodstock Graphics setting a scale of 1000 will then change the output's y-axis to MMBF.

2.6 Optimize: asking Woodstock to plan management

In addition to tracking forest inventory and management actions, outputs are important to Woodstock modeling because the management objective and its constraints are specified in terms of outputs in the optimize section. Since only a few outputs have been created at this point what can be done in the optimize section is limited at this point.

1. Ask Woodstock to find the maximum harvest volume, subject to an even flow constraint so that it doesn't simply wait until the end of the planning period when standing volumes are highest and then harvest the entire West Oregon District in a year (Figure 11).

```
;Outputs
*GROUP inventory
*OUTPUT oInventoryCounty(_TH1) standing inventory by county, MMBF
*SOURCE ? ? ? ? ? ? ? ? ? _INVENT yMMBFstanding

*OUTPUT oInventoryOwner(_TH2) standing inventory by landowner, MMBF
*SOURCE ? ? ? ? ? ? ? ? ? _INVENT yMMBFstanding

*OUTPUT oClearcutEligibleAreaByAge acres aged 50-150 that can potentially be clearcut
*SOURCE ? ? noListedSpecies upland ? canClearcut ? ? ? @AGE(10..30) _INVENT _AREA

*OUTPUT oClearcutEligibleAreaByQmd acres with QMD >= 15 inches that can potentially be clearcut
*SOURCE ? ? noListedSpecies upland ? canClearcut ? ? ? @YLD(yQMD, 15..999) _INVENT _AREA

*GROUP harvest
*OUTPUT oClearcutArea clearcut area, acres
*SOURCE ? ? ? ? ? ? ? ? ? aClearcut _AREA

*OUTPUT oClearcutScribner clearcut yield, MBF
*SOURCE ? ? ? ? ? ? ? ? ? aClearcut yMBFrecovered
```

Figure 9: Basic outputs for tracking inventory and harvest. `yMMBF` is the MMBF yield added in Section 2.3, `_INVENT` tells Woodstock to add up all standing inventory for each planning period, and `_AREA` tells Woodstock to add the total area the `aClearcut` action was performed on in a planning period. `@AGE()` and `@YLD()` provide additional filtering beyond what's possible with the mask.

```
;Reports
*TARGET FOR_459_Assignment_5_allrep.xlsx
_ALL 1.._LENGTH
```

Figure 10: An example of a reports section modified to write all outputs to a .xlsx file. Changing the extension to .csv writes the same report as a comma separated value file instead.

```
;Optimize
*OBJECTIVE
_MAX oClearcutScribner 1.._LENGTH

*CONSTRAINTS
_EVEN(oClearcutScribner, 10%) 1.._LENGTH

*FORMAT MOSEK
```

Figure 11: An introductory optimization section which seeks to maximize an even flow clearcut harvest volume.

Woodstock defaults to using Mosek (`*FORMAT MOSEK`) as its linear program solver with Cplex also being available (`*FORMAT CPLEX`). Both solvers should be installed on lab machines with Woodstock already configured to be aware of them (Section 7). Both solvers should produce identical results, within numerical precision, but Mosek is likely to be faster on FE/FOR 459 problems.

2.7 Verifying the model works

Woodstock now has a minimal set of information and instructions to develop an entry level harvest approach for the West Oregon District. Woodstock offers Run Woodstock (the ▶ button), Generate Solve (▶▶), and Generate Solve & Run (▶▶▶) as options for running models. Run Woodstock replays the most recent solution, recalculates outputs, and redisplays the graphs. While this is useful as a time saving option for reviewing results or adjusting outputs, if model changes are made it's necessary to choose one of the generate options to rebuild the linear program Woodstock is solving and then update the solution to the newly changed program. If the generate and solve steps are skipped after changes in the areas, yields, actions, transitions, optimize, control, or lifespan sections Woodstock will continue to show old results which no longer reflect the current model.

1. Run the model (Run → Generate Solve & Run, F12, or the ▶▶▶ button). If there are no immediate errors, Woodstock will show a couple progress windows, pop up a command line, and eventually show a blank Woodstock Graphics window. If there are errors, fix them and rerun. There will be a warning about unactionable areas, which can be ignored (see below).
2. After the Woodstock Graphics window opens, drag OINVENTORYCOUNTY from the Outputs pane to the main area of the window. In the resulting popup window, double click Polk, Lincoln, and Benton to add those inventories.
3. Repeat the plot drag and double clicking for OINVENTORYOWNER. Drag OCLEARCUTAREA and OCLEARCUTSCRIBNER over as well. Right clicking any of the graphs offers a properties menu where colors, markers, and other aspects of the display can be adjusted.
4. Once configured, the graphs should show fairly steady inventory levels and a stable harvest volume. Values will vary depending on the stand classifications and FVS models used but roughly 45 MMBF annual harvest with around 2700 MMBF standing on the district is reasonable.
5. Drag OCLEARCUTELIGIBLEAREABYAGE to a new plot. Drag OCLEARCUTELIGIBLEAREABYQMD onto this plot as well to show the two series together. These two diagnostic outputs should show several thousand acres staying available for clearcut. It's likely age availability will have a more or less flat overall trend but that QMD availability will show a gradual decline.
6. Close the graphics window and save the changes.

There will be a warning in the log file saying there are development types which no actions could be performed on. This should be Woodstock indicating there are no harvest actions capable of doing logging in riparian and marbled murrelet areas, which is desirable as clearcutting should not occur in these areas. Opening the NoLpChoices.csv file generated in Woodstock's working folder can provide confirmation.

The warning about development types for which no actions can be scheduled can be interpreted as an indication these areas don't need to be included in the model. Sometimes, it may end up being desirable to drop them. However, their removal is best approached carefully. Neglecting reserved riparian and murrelet areas is a substantial omission to tasks such as inventorying forest types present on the District, assessing carbon stocks and wildfire fuel loads, and measuring biodiversity or habitat connectivity. Conversely, including these reserved areas may mean it's important to differentiate between noncommercial inventory in reserves and merchantable inventory in harvestable upland areas.

2.8 Addressing uncertainty and variability in harvest revenue and discount rates

Timber prices fluctuate over time (WA DNR 2023), as do the timber assortments a harvest will yield, success in offering timber sales, the cost of performing a harvest (West et al. 2022), preferred silvicultural prescriptions (West et al. 2021), financial market rates of return, and levels of disturbance. As a result, a range of revenue and discounting scenarios are usually considered when developing forest management plans. Simplifications are common, such as assuming real stumpage values, reforestation costs, and discount rates remain constant within the planning horizon. The values in ODF_Capstone_Constraints.pdf are reasonable defaults for costs and revenue (Figure 12) but should be adjusted to account for the different species classifications established while writing the management plan introduction and during the FVS assignment.

```
;Yields
*YC ? ? ? ? ? ? DF ? ?
yStumpageClearcut _MULTIPLY(yMBFrecovered, 500)

*YC ? ? ? ? ? ? RA ? ?
yStumpageClearcut _MULTIPLY(yMBFrecovered, 325)

;Outputs
*OUTPUT oStumpageClearcut stumpage revenue, US $
*SOURCE ? ? ? ? ? ? ? ? ? aClearcut yStumpageClearcut

*GROUP economic
; ODF reforestation cost = 125 + 225 + 75 + 30 + 350 + 85 + 125 = US$
1015/acre
*OUTPUT oReforestationCost reforestation cost, US $
*SOURCE oClearcutArea * 1015

*OUTPUT oNetRevenueClearcut net revenue from clearcutting, US $
*SOURCE oStumpageClearcut - oReforestationCost
```

Figure 12: Examples of basic stumpage revenue calculations using species masks and ODF_Capstone_Constraints.pdf clearcut revenues. Stands with DFRA, DFWH, and other species mixtures would have different stumpage prices, as would thinnings or clearcuts at atypical ages.

Depending on group objectives, it may also be appropriate to consider

- Use of time-varying yield tables (*YT instead of *YC) to capture shifts in stumpage value as stands age due to assortments shifting towards 2-saw and higher grades, increases in defect and breakage, and—eventually—increases in harvest costs as trees become too large for mechanized harvest equipment to easily handle.

- High and low revenue cases, quite possibly over a range of discount rates. These might be individual Woodstock runs to compare the resulting management or Woodstock might be configured to optimize over a weighted sum of possibilities as a way of encouraging development of a plan which would be more robust across multiple possible outcomes.
- Altering the District's merchantable timber inventory over time by using transitions to replant stands with different species. Since the passthrough transition in Section 2.4 does not change the stand species theme, it leaves `aClearcut` reforesting with the same species as were harvested. It is often the case this default is poorly aligned with management objectives and ODF policy.

When discounted revenues need to be considered, `_DISCOUNTFACTOR()` can be used to create a yield for each discount factor of interest (Figure 13). Net present value (NPV) can then be calculated in the output section by multiplying revenue outputs with the discount factor yields. If NPV is a management objective it may be helpful to `_SUM()` these outputs in the optimization section.

```
; if net present values are needed, create discount factor yields
*YT ? ? ? ? ? ? ? ? ? ?
yDiscountFactor3 _DISCOUNTFACTOR(3%, 5, Half)

*YT ? ? ? ? ? ? ? ? ? ?
yDiscountFactor4 _DISCOUNTFACTOR(4%, 5, Half)

*YT ? ? ? ? ? ? ? ? ? ?
yDiscountFactor5 _DISCOUNTFACTOR(5%, 5, Half)
```

Figure 13: An example of discount factor yields. The 5 in `_DISCOUNTFACTOR()` indicates annual discounting and the `Half` tells Woodstock to assume revenue is received in the middle of the discounting period.

3 Strategic Planning Tasks

Extend the model created in the previous section to manage for your group's objectives, subject to the complete set of ODF's constraints and implementation of ODF's policies. In cases where policies have flexibility, default choices should be altered if doing so better fits management objectives—for example, a biodiversity objective favors planting different selections of seedlings than financially oriented objectives.

The following four sections in this document offer support for this task by describing ways of extending the base model, limitations of Woodstock models, managing model data, and troubleshooting issues with Woodstock. In general, the more complex a model becomes the longer its generate, solve, and run cycles require and the more memory the model requires. It can therefore be helpful to save time by shortening the planning horizon in the control selection when doing model development and then extend it back out to evaluate changes.

4 Thinning: tracking stand state and _SHIFTing yields

Implementing thinning in Woodstock easily becomes complicated. Three approaches are available.

1. Simulate all thin timings and intensities of management interest in FVS, transfer all of the results to Woodstock yield tables, and write a network of thinning actions and transitions which constrain Woodstock to use only the harvest timings and intensities simulated in FVS. This approach is the most biologically accurate option but is time consuming to implement—enabling just a single set of stand entry timings for ODF's three default thinning prescriptions quadruples the number of `*Y` yield tables.
2. Approximate a thin by using `_SHIFT` to tell Woodstock to treat the stand as younger than it actually is. With this approach, a thin's yield becomes the stand's growth over the number of years it's `_SHIFT`ed by. For example, if a 40% thin is approximated by `_SHIFT`ing a 30-year old stand with 18 MBF ac^{-1} back to age 25 in its yield table when it had 11 MBF ac^{-1} then the thin produces 7 MBF ac^{-1}. Such translations in time are not entirely correct—thins' effects on taper are neglected, for example—and supporting work in FVS is needed to estimate the number of years to `_SHIFT` by. However, a limited number of `*YC` yield tables is required instead of many `*Y`s and the required actions and transitions are simpler.
3. Use `*PARTIAL` to have Woodstock harvest part of the stand's area. This approach is the simplest as it requires only creation of thinning actions marked with `*PARTIAL` with no additional work in FVS or yield tabling. However, it's the least accurate as, for example, a 30% thin is approximated by clearcutting 30% of a stand and leaving the other 70% to grow as if it was undisturbed. Partial cutting of an area means all thins are proportional and retained trees do not expand to reoccupy the growing space left by the thin. `

As a compromise between accuracy and complexity, this section describes use of the `_SHIFT` approach. The number of years to `_SHIFT` a stand by that best matches ODF's thinning prescriptions varies with species composition and increases as site productivity declines. Since Woodstock performs linear interpolation between the five year FVS timesteps in the `*Y` yield tables, one year planning periods enable approximating thins by shifting stands any integer number of years. However, analysis of FVS predictions shows that, for most stands on the West Oregon District, `_SHIFT`ing by 5 or 10 years per thin is acceptable. Conifers start at five years in site class I and increase to 10 years in low site class III, red alder plantations tend to be more in the 7–10 year range. It's therefore convenient to introduce stand type and productivity aggregates in the landscape section. Also needed is a theme which tracks the extent to which a stand has been `_SHIFT`ed by thinning (Figure 14).

Depending on choices made in the FVS assignment, it may be attractive to integrate thinning information with the density theme. In other cases the theme will need to be added to the landscape shapefile, all polygons defaulted to `unthinned` (to increase accuracy lower density stands could be set to one of the `thinShift` levels—read the rest of this section before starting to edit the shapefile, though), and the areas section rebuilt from the revised shapefile (see Section 6).

```
;Landscape
*THEME {7} stand species classifications
 DF
 DFRA
 DFRAWH
 DFWH
 RA
 *AGGREGATE notRA
  DF DFRA DFWH DFWHRA

*THEME {8} site productivity
 siteClassIhigh
 siteClassImedium
 siteClassImediumLow
 siteClassIlow
 siteClassIIhigh
 siteClassIIlow
 siteClassIIIhigh
 siteClassIIIlow
 *AGGREGATE notSiteClassIII
  siteClassIhigh siteClassImedium siteClassImediumLow siteClassIlow
siteClassIIhigh siteClassIIlow
 *AGGREGATE siteClassIII
  siteClassIIIhigh siteClassIIIlow

*THEME {9} amount of thinning performed on a stand
 unthinned
 thinShift05
 thinShift10
```

Figure 14: Landscape aggregates and theme changes to support thinning. The example species and productivity classifications shown here should be replaced with those chosen in the FVS assignment. Also, the theme used for tracking thinning status might be theme seven, eight, or nine.

An Action to perform thinning, subject to the `*OPERABLE` restrictions set by ODF's policies, are needed. The associated transition moves stands from `unthinned` to what will be either a five year _SHIFT (`thinShift05`) or a 10 year _SHIFT (`thinShift10`) (Figure 15). Note that

- Aggregates in a transition's `*SOURCE` are matched by `?` in the `*TARGET`—if an aggregate were to be specified in `*TARGET`, Woodstock would reject it because it's neither a passthrough (`?`) or a specific theme value.
- `aClearcut`'s transition is updated to move stands back to the `unthinned` state. Without this change a stand would remain marked as thinned even after it's been clearcut, which is incorrect.

For simplicity in defining outputs, it's useful to have a consistent name for a particular yield, regardless of whether the yield is coming from an unthinned stand, a stand that's been thinned once, or thinned more than once. Since `_SHIFT` must create yields with a name distinct from the yield it's shifting, one way of having simple yield naming to write outputs against is to attach a `MaybeShift` suffix to all of the yields. For unthinned stands the yields are passed through unchanged by `_MULTIPLY`ing by one and for thinned stands they're `_SHIFT`ed (Figure 16). `yMBFrecovered` and `yMMBFrecovered` are updated to the possibly shifted board foot yields and, finally, `yStumpageThin` is introduced as a parallel to `yStumpageClearcut` in Section 2.8. As with `yStumpageClearcut`, `yStumpageThin` requires expansion to cover other species classes and, possibly, conversion to `*YT`.

```
;Actions
*ACTION aFirstThin N ODF first thinning from 195 ft²/ac basal area
*OPERABLE aFirstThin
? ? noListedSpecies upland ? ? unthinned ? yBA >= 195 & _AGE <= 9

;Transitions
*CASE aFirstThin
*SOURCE  ? ? ? ? ? ? notRA notSiteClassIII unthinned
*TARGET ? ? ? ? ? ? ? ? thinShift05 100 _LOCK 2

*SOURCE ? ? ? ? ? ? notRA siteClassIII unthinned
*TARGET ? ? ? ? ? ? ? ? thinShift10 100 _LOCK 2

*SOURCE ? ? ? ? ? ? RA ? unthinned
*TARGET ? ? ? ? ? ? RA ? thinShift10 100 _LOCK 2

*CASE aClearcut
*SOURCE ? ? ? ? ? ? ? ? ?
*TARGET ? ? ? ? ? ? ? ? unthinned 100
```

Figure 15: An action to approximate ODF's silvicultural prescription for a first thin and its associated transition. The restriction of thinning to stands age 45 and younger is somewhat arbitrary and can be adjusted if needed. `_LOCK` locks the area for the specified number of periods so that Woodstock won't, for example, immediately reenter the stand for a second thinning or to clearcut it.

```
;Yields
; passthrough FVS yields in unthinned stands
*YC ? ? ? ? ? ? ? ? ? unthinned
yTpaMaybeShift _MULTIPLY(yTpa, 1)
yBAmaybeShift _MULTIPLY(yBA, 1)
ySdiMaybeShift _MULTIPLY(ySDI, 1)
yCcfMaybeShift _MULTIPLY(yCCF, 1)
yTopHtMaybeShift _MULTIPLY(yTopHt, 1)
yQmdMaybeShift _MULTIPLY(yQMD, 1)
yTCuFtMaybeShift _MULTIPLY(yTCuFt, 1)
yMCuFtMaybeShift _MULTIPLY(yMCuFt, 1)
yBdFtMaybeShift _MULTIPLY(yBdFt, 1)

; _SHIFT yields in thinned stands
*YC ? ? ? ? ? ? ? ? ? thinShift05
yTpaMaybeShift _SHIFT(yTpa, 1)
yBAmaybeShift _SHIFT(yBA, 1)
ySdiMaybeShift _SHIFT(ySDI, 1)
yCcfMaybeShift _SHIFT(yCCF, 1)
yTopHtMaybeShift _SHIFT(yTopHt, 1)
yQmdMaybeShift _SHIFT(yQMD, 1)
yTCuFtMaybeShift _SHIFT(yTCuFt, 1)
yMCuFtMaybeShift _SHIFT(yMCuFt, 1)
yBdFtMaybeShift _SHIFT(yBdFt, 1)

*YC ? ? ? ? ? ? ? ? thinShift10
<same as thinShift05 but with _SHIFT(..., 2)>
```

```
; 1000 standing BF/MBF / (100% - 5% defect and breakage) => 1053 standing BF
predicted by FVS/pond delivered MBF
*YC ? ? ? ? ? ? ? ? ?
yMBFrecovered _DIVIDE(yBdFtMaybeShift, 1053)

*YC ? ? ? ? ? ? ? ? ?
yMMBFstanding _DIVIDE(yBdFtMaybeShift, 1000000)

*YC ? ? ? ? ? ? DF ? ?
yStumpageThin _MULTIPLY(yMBFrecovered, 250)
```

Figure 16: Example of yields to updated with a `MaybeShift` suffix to support thinning. Actual yield names depend on choices made in the FVS assignment. There may also be more yields to `_SHIFT` than shown here or there may be fewer yields.

With yields in place, thinning outputs (Figure 17) can be added and the optimization section updated to include thinning and the economic outputs from Section 2.8 (Figure 18).

```
;Outputs
*GROUP harvest
*OUTPUT oThinArea area thinned, acres
*SOURCE ? ? ? ? ? ? ? ? ? aFirstThin _AREA

*OUTPUT oThinScribner thinning yield, MBF
*SOURCE ? ? ? ? ? ? ? ? ? aFirstThin yMBFrecovered

*OUTPUT oStumpageThin stumpage revenue from thinning, US $
*SOURCE ? ? ? ? ? ? ? ? ? aFirstThin yStumpageThin


*GROUP economic
*OUTPUT oNetReveneue total net revenue
*SOURCE oStumpageThin + oNetRevenueClearcut
```

Figure 17: Outputs to add to support thinning.

```
; Optimize
*OBJECTIVE
_MAX oNetReveneue 1.._LENGTH

*CONSTRAINTS
_EVEN(oNetReveneue, 10%) 1.._LENGTH
_EVEN(oClearcutArea, 30%) 1.._LENGTH
_EVEN(oThinArea, 30%) 1.._LENGTH
```

Figure 18: Optimization section revised to make use of economic and thinning outputs.

Once all of its sections have been updated to support thinning, the model can be debugged and put through another Generate Solve & Run cycle to include a first thinning as a silvicultural option. After running, the same drag, drop, and modify process can be used to incorporate the newly added thinning outputs into the graphs. The output should show some use of thinning in conjunction with clearcutting. If no thinning occurs that's most likely a bug and should be investigated to find out why the thinning action isn't triggering—it may be helpful to introduce diagnostic outputs similar to `oClearcutEligibleAreaByAge` or to temporarily comment out `aClearcut` while investigating.

This process of adding support for a first thinning can then be extended to support second and third thins.

1. Perform actual thinning sequences in FVS or go through the FVS outputs to assess how many years each individual thin should `_SHIFT` by as a function of species and site classification. Add the second thin's shifts to the first to find the total number of years to shift by when both thins are performed.
2. Create corresponding `thinShift15`, `thinShift20`, … `*YC`s in the yields section.
3. Create a second thin action with `*OPERABLE`s that match the `thinShift05`s and `thinShift10`s from the first thin and set the `*TARGET`s of its transition to increase the amount of shift.
4. Add outputs for the second thin and update total outputs to include the second thin.
5. Debug the changes, Generate Solve & Run, and add graphs for the second thin.
6. Once everything looks good, repeat 1–5 for a third thin.

Keeping to simple choices of how many years to `_SHIFT` by is suggested. Models can usually be made arbitrarily detailed and it's easy to end up writing a `*YC` for each thin in each species and productivity combination. However, it's relatively unlikely tactical and operational planning, along with the process of executing timber sales, will end up close enough to the harvest dates Woodstock schedules to justify the effort of doing high precision modeling. Also, given the growth and yield approximations being made due to the use of `_SHIFT` and the classification process used for FVS inputs, the value of precisely determining `_SHIFT` amounts is questionable.[4]

5 Woodstock Model Limitations in FE/FOR 459

This section details a few limitations of Woodstock models which groups are likely to find relevant. These issues should be considered but, depending on management objectives and model development priorities, may not need to be addressed by adjusting model setup or changing how the model is coded. It is, however, valuable to remain aware of errors, assumptions, and approximations when modeling. Not checking for such limitations increases the risk of creating forest management plans which are precise but not meaningful because they're based on inaccurate modeling.

5.1 Climate change and uncertainty in growth and mortality

By default, FVS predicts future growth and yield based on measurements of growth under past climates. This approach almost certainly results in increasing prediction error over time, even in areas such as Oregon's Coast Range which are buffered from warming temperatures by cold upwelling water in the eastern Pacific Ocean (Halofsky et al. 2022). A Climate-FVS extension is included with FVS but it's unclear how its site productivity and carrying capacity multipliers should be configured as a function of the climate pathway being considered (Fekety et al. 2020). Growth models which avoid this difficulty are in

[4] A quick approximation for the second thin is probably 10 years in site class I and II, 15 years in site class III or with significant red alder, and possibly a few 20 year special cases. After the third thin, shifts of 15–25 years are likely, possibly up to 30 or 35 in some cases.

use for forest planning in the Coast Range (OSU 2023, §4.4) but, as they're currently research rather than operational tools, their application is beyond the scope of this course.

While adjusting FVS predictions for climate is not required, unless called for by group objectives, some simpler alternatives to consider are

- Use of time-varying yield tables to derate harvest volumes based on assumptions of reduced growth and elevated mortality due to increased moisture stress and, secondarily, Swiss Needle Cast growth (Jarecke et al. 2023, Lee et al. 2022).
- Assuming replanting with stock from different seed zones will minimize error in growth predictions. This appears to most likely be optimistic but it is not uncommon to include what are effectively past climate predictions as controls when comparing predictions across different model configurations.
- Using transitions to replant stands with different species or different species mixtures which may be more resilient to future climates. In particular, substantial uncertainty exists over behavior and persistence of the coastal fog zone (Halofsky 2022).

5.2 Different planning period lengths: granularity versus faster model solutions

Woodstock does not know how long a planning period is. As a result, all ages in the areas, yields, lifespan, outputs, and optimize sections are expressed in planning periods. While five year planning periods are a convenient default, the computational cost of solving a linear program over an N period planning horizon increases as $O(N^2)$ or faster. Generating, solving, and running a plan for 20 periods therefore takes at least four times as long and four times as much memory as a plan for 10 periods. This can result in long solution times which make it hard to get FE/FOR 459 work done, can prevent Woodstock from being able to solve models when it needs more memory than a lab machine can provide, or cause Woodstock to run extremely slowly in cases where Mosek requires somewhat more memory than a lab machine has physically available.

One mitigation for these problems is to reduce the number of periods within a planning horizon by increasing period length. For example, a 50 year plan with five year periods and a 100 year plan with ten year periods both have a 10 periods planning horizon and therefore, other aspects of the models being equal, require the same amount of computation to generate, solve, and run. However, the models' code differs in any aspect dealing with time, including and Spatial Optimizer settings.

Alternatively, if code-based greenup constraints (Section 5.5) are to be modeled then shorter period lengths are required to capture open area constraints (for example, Figure 19). One, two, or three year periods are suggested as four year periods align poorly with Oregon Forest Practices Act greenup requirements.

The primary steps in a Woodstock models' period length are

1. Create a copy of the FVS yield tables and use formulas in Excel (or another spreadsheet program) and divide the _AGE column of the yield table by the period length. Update Woodstock's yield section with the new tables.

2. Go through the model's code and multiply or divide age related numbers (Table 1) by the new period length, as appropriate. Given a working model with one year periods, it's suggested to copy the entire model, change the copy's period length, and then compare the two models' outputs to check for places where age conversions may have been missed.

Table 1: Differences between Woodstock models using one year planning periods and multi-year planning periods.

section	one year planning periods	multi-year planning periods
spatial setup	age divisor is none or one	age divisor is length of period
lifespan	max period matches years	max period is age divided by length of period
yields	_AGE column is FVS age	_AGE is FVS age divided by length of period
actions	_AGE matches years	_AGE is years converted to periods
outputs	@AGE matches years	@AGE is years converted periods
optimize	constraints have annual values	constraints have values for full period length

```
;Actions
*ACTION aClearcut Y clearcut and reforest
*OPERABLE aClearcut
? ? noListedSpecies upland ? ? ? canClearcut _AGE >= 50 & _AGE <= 150 & yQMD
>= 15 ; 10 five year periods = age 50 years, 25 periods = age 125 years
```

Figure 19: The clearcut action from Figure 7 with its constraints converted from five year planning periods to one year planning periods.

5.3 Thinning in murrelet and riparian buffer areas

Because all murrelet features are dissolved, the shapefile constructed in the GIS assignment does not distinguish between murrelet management areas and their surrounding buffers. Similarly, the GIS assignment's stream buffering does not distinguish between the no harvest zones adjacent to streambanks, inner riparian management areas, and outer riparian management areas. As a result, the West Oregon District's presentation to Woodstock by default lacks the polygons needed to schedule streamside and murrelet buffer thins. Since there are 4600 acres of riparian buffer and 2200 acres of murrelet buffer, these choices exclude thinning from around 20% of the District's 36,000 acres.

If thinning and, possibly, underplanting operations in these areas would be valuable to your group's objectives then it may be worth altering the shapefile to distinguish between no touch murrelet and riparian cores and thinnable buffer areas. Steps in this task are

1. Return to the murrelet part of the GIS assignment. Instead of dissolving all murrelet areas, use the STATUS field to remove prior boundaries and prior buffers and then flow STATUS into the union and onwards to providing additional values in Theme3. The distinction between extant and draft MMMAs and buffers can be simplified away or retained. (But remember Woodstock does not support spaces in theme attributes.)
2. Redo the riparian buffer part of the GIS assignment with a series of variable buffer widths for the no harvest, inner RMA, and outer RMA zones. Union these areas together to create a layer

whose slope position has a more detailed annotation than simply riparian or not and then union this more detailed riparian layer into the main layer.

3. Rerun sliver elimination, rebuild the theme fields, and export the updated later as a shapefile for Woodstock. This process can go quickly since the Python for the themes is now already setup and just needs to be pasted into the field calculators.

5.4 Tree growth is limited on heavily used landings and most roads

While the GIS assignment includes steps to generate a road layer for the West Oregon District, it does not subtract the areas occupied by roads and landings from the stands. As a result, Woodstock has no way of knowing few trees are likely to grow in large yarder landings with compacted soils, maintained roads, or well compacted but unsurfaced roads which weren't ripped after being used for a harvest. The growing area of a stand's management polygons is therefore likely to be overestimated, leading the model to somewhat overestimate growth and yield.

If it's desired to do so, this error can be reduced by

1. Returning to the roads part of the GIS assignment and following the riparian buffer establishment and unioning process to create a Woodstock shapefile with road polygons designated by a theme.
2. Update the yield tables' masks to exclude roads and update the rest of the model for the additional theme.

There is a powerline south of Highway 20 which, by default, is also treated as having unrestricted tree growth. This right of way is a good candidate for a theme attribute which would exclude it from consideration as a forest stand similar to an exclusion for roads.

5.5 Woodstock models are nonspatial by default

Internally, Woodstock operates by generating a linear program from the content of a model's various sections, invoking the solver specified by `*FORMAT` to find the sequence of management actions which best accomplishes management objectives, and processing that sequence to create reports and graphics. Unless spatial constraints are included in this process, the resulting pattern of clearcuts may violate the Oregon Forest Practices Act (OFPA)'s maximum opening size of 120 acres. There are two main approaches for avoiding creation of illegal forest plans.

1. Use a planning period long enough there's sufficient slack in the plan any given clearcut can green up before an adjacent one needs to be performed. Often it's assumed planning periods which are at least as long as the greenup time provide adequate slack but a combination of bad luck with stand geometry and clearcut scheduling can still make it difficult or impossible to satisfy regulatory constraints in tactical or operational planning. The risk of this approach offset by having a comparatively small linear program, making it faster to generate and solve.
2. Include greenup constraints in the model which impose a minimum separation in time between clearcuts of adjacent stands sufficient to meet OFPA free to grow requirements. Given reliable

reforestation success, this approach carries less operational risk than long planning periods and helps ensure strategic planning sets expectations which can be met by tactical planning. However, the model setup becomes more complex as Woodstock must know where all of the stands are and generate and solve times easily increase by an order of magnitude.

The first approach can be implemented by using a planning period length of five years or more. The second approach can be implemented by enabling `_GREEENUPCONSTRAINTS()`. Once the necessary setup is in place, enablement is fairly simple and follows from the Woodstock documentation (Figure 20).

Woodstock does not enforce a maximum opening size when solving linear programs. However, because greenup constraints prevent the creation of an opening larger than the blocking theme specified for grouping polygons when building the feature adjacency tables, the largest opening a linear solution can create is the area of the largest clearcuttable polygon present in the shapefile.

Setup tasks for `_GREEENUPCONSTRAINTS()` are

1. Build a feature adjacency table for stands using the process described in the Woodstock documentation for building a feature adjacency table. The feature will be the theme containing the stand IDs and, since the Oregon forest practices require a 300 foot separation between clearcuts (ORS 527.740), a 300 adjacency distance is sufficient.[5] Remsoft recommends enabling clique identification.
2. Include `_GREEENUPCONSTRAINTS()` in the optimize section and create its supporting outputs (Figure 20). `DATABASENAME` will be the path to the stand adjacency database created above.

```
;Outputs
*OUTPUT oClearcutAreaByStand(_TH5) area clearcut in each stand, acres
*SOURCE ? ? ? ? ? ? ? ? ? aClearcut _AREA

*OUTPUT oIsHarvest(_TH9) harvest flag for greenup constraints
*SOURCE _IF(oClearcutAreaByStand > 0, 1)

; Optimize
_GREENUPCONSTRAINTS(DATABASENAME = .\GIS\West Oregon District adjacency 300
foot.dbf, _PERIODS = 1.._LENGTH, _BINARYOUTPUT = oIsHarvest, _USECLIQUES =
ON, _GREENUPDELAY = 5)
```

Figure 20: Example of `_GREEENUPCONSTRAINTS()` with one year planning periods and its associated outputs, assuming cliques were identified when building adjacent features.

Once it's set up, the model can be tested with `_GREEENUPCONSTRAINTS()` and debugged to get all of the changes working. To shorten generate and solve times, `_GREEENUPCONSTRAINTS()` can be commented out when developing other aspects of the model and then turned back on as needed to check for spatial feasibility. It is likely stand polygons larger than 120 acres are present in the GIS data

[5] ORS 527.740 specifies a 300 foot slope distance. Unless the ground is flat, using a 300 foot map distance in Woodstock therefore overestimates adjacency between management polygons.

and would need to be reconfigured into smaller management units to avoid clearcuts larger than 120 acres.

6 Data Management

Woodstock models are self-contained and can be relocated by moving or copying the entire model folder. A shared OneDrive and T: both support collaborative development of a Woodstock model by multiple group members. However, since Woodstock lacks support for merging changes, it's best if only one person works on a particular copy of the model at any given time—changes developed concurrently by different group members must later be merged manually. Periodically backing up the model is recommended as it's often helpful to refer back to known good code if changes aren't going well. Having backups also reduces risks such as having to rebuild shapefiles if, for example, Woodstock's shapefile hasn't been copied and something goes wrong with ArcGIS Pro.

A not uncommon need is to revise the shapefile, either to add themes or to adjust the specification of existing themes. Woodstock has abilities to join tables to the shapefile and to run scripts over the shapefile. For the most part, however, it's easier to edit the shapefile in GIS and then restart Woodstock to pick up the changes. If the shapefile is modified in this way, both the areas and landscape sections need to be rebuilt. Since manual edits to the areas section are uncommon, in most cases Build Section → Areas Section can be used with overwrite existing section. Since rebuilding the landscape section destroys any added aggregates and descriptions it's generally easier to make changes to theme values manually.

If FVS runs are redone then the FVS to Excel, yield table modification, and Excel to text cycle can be repeated, followed by copy and paste into the relevant parts of the yield section. Same for adding or removing yield tables. It's suggested to place `*YC` and `*YT` tables at the end of the yield section to separate them from the `*Y` tables from FVS. However, Woodstock imposes no required structure.

Within the model folder, Woodstock creates __history and temp folders. In some cases these can eventually become large, in which case they can be cleared out.

7 Troubleshooting Woodstock

Woodstock warnings and errors are generally informative about both what a problem is and where it is. Reading the messages is therefore effective in fixing many problems, especially if done in conjunction with checking the Woodstock documentation to understand valid syntax for model code.

More involved issues commonly include

- Inconsistent use of time, usually due to missing an edit when changing period lengths or use of unshifted yields when `_SHIFT`ing for thinning. In the first case the fix is to finish the changes, in the second is to ensure all outputs are written with possibly shifted yields. These errors have to be caught by inspection of model outputs and critical reading of model code.
- Actions not running as expected due to lack of eligible ground to perform the action on. In general, when adding an action it's helpful to also add corresponding diagnostic outputs (such as Figure 9's `oClearcutEligibleAreaByAge` and `oClearcutEligibleAreaByQmd` outputs). Woodstock unfortunately does not support combining `@AGE()` and `@YLD()` for more specific

diagnostics but, in most cases, it's possible to construct to sufficiently informative sets of diagnostic outputs from combinations of masks, `@AGE()`, and `@YLD()`.

- Actions not running because they're not helpful to achieving management objectives. If diagnostic outputs show an action has operable ground available but the linear solution isn't scheduling the action, typically it's an indication the action isn't worth taking. This can be verified by using a constraint to force use of the action and confirming the objective function becomes less desirable. Note that, if suitable ground for an action isn't available in every period, using a `1.._LENGTH` constraint will make the model infeasible. Use diagnostic outputs to determine if a more restricted range of periods is needed.

Additionally,

- Woodstock has what's arguably both a bug and a feature where changes to output descriptions don't automatically flow to the corresponding graphs once the graphics section has been populated. A quick solution is to open the graphics section and repeat edits to the descriptions.
- Run → Check Syntax will sometimes catch issues not reported by Run → Run Woodstock and possibly other pathways. The control section also has an option to turn on debug output that's written to ErrorWk32.log.
- Connections between Woodstock and linear program solvers are somewhat fragile, possibly because the solvers are installed separately from Woodstock proper. For example, if the solver specified by `*FORMAT` in the optimization section is not available the model will run without warnings or errors but no actions will be scheduled. In general, both Cplex and Mosek should have paths listed for them listed in Tools → Paths and Programs. If these paths are empty, scan C:\Remsoft\<version>\Solvers for solvers. If the Solvers directory isn't present that signals no solver installer was run. If the Cplex or Mosek directory is missing under Solvers then it's most likely that the installer for that solver wasn't run.
- Remsoft licenses are potentially fragile as well. Fifty licenses are available for each program (Woodstock, Cplex, Gurobi, and others are used) and a given machine may take multiple licenses. It's therefore conceivable running Woodstock may fail because all 50 licenses are in use even when fewer than 50 machines are in simultaneous use. The Remsoft License Manager has, on occasion, been observed not to release licenses after they no longer need to be held. The license manager sometimes also crashes (segmentation violation in RemsoftLm.exe).

References

Fekety PA, Crookston NL, Hudak AT, Filippelli SK, Vogeler JC, Falkowski MJ. 2020. Hundred year projected carbon loads and species compositions for four National Forests in the northwestern USA. Carbon Balance and Management 15:5. https://doi.org/10.1186/s13021-020-00140-9

Halofsky JE. 2022. Adapting to Changing Climate and Disturbance Regimes in the Pacific Northwest. Forest Health in Oregon 2022: State of the State conference, College of Forestry, Oregon State University. https://www.forestry.oregonstate.edu/cpe/forest-health-2022

Halofsky JE, Peterson DL, Gravenmier RA. 2022. Climate change vulnerability and adaptation in southwest Oregon. Pacific Northwest Research Station GTR-995, US Forest Service. https://www.fs.usda.gov/research/treesearch/63850

Jarecke KM, Hawkins LR, Bladon KD, Wondzell SM. 2023. Carbon uptake by Douglas-fir is more sensitive to increased temperature and vapor pressure deficit than reduced rainfall in the western Cascade Mountains, Oregon, USA. Agricultural and Forest Meteorology 329:109267. https://doi.org/10.1016/j.agrformet.2022.109267

Lee EH, Beedlow PA, Brooks JR, Tingey DT, Wickham C, Rugh W. 2022. Physiological responses of Douglas-fir to climate and forest disturbances as detected by cellulosic carbon and oxygen isotope ratios. Tree Physiology 42(1):5–25. https://doi.org/10.1093/treephys/tpab122

OSU CoF. 2023. Elliott State Research Forest Management Plan: December 2023. College of Forestry, Oregon State University. https://www.forestry.oregonstate.edu/elliott-state-forest/forest-management-plan

WA DNR. 2023. Timber Sale Query/Log Prices. Washington State Department of Natural Resources. https://www.dnr.wa.gov/programs-and-services/product-sales-and-leasing/timber-sales/timber-sale-querylog-prices

West T, Sessions J, Strimbu BM. 2021. Heuristic Optimization of Thinning Individual Douglas-Fir. Forests 12(3):280. https://doi.org/10.3390/f12030280

West T, Sessions J, Strimbu BM. 2022. Steep Slope Harvest System Models for Small to Large Trees 13(2):305. https://doi.org/10.3390/f13020305

Spatial Optimization Setup for Woodstock Optimization Suite

1 Introduction

A limitation of Woodstock is the linear solution process finds the optimum management that's possible within what the model describes. Since Woodstock models are nonspatial and fairly high level, the optimum linear solution easily become impractical due to model limitations and operational realities (Figure 1:Figure 1). Beyond `_GREENUPCONSTRAINTS()`, Woodstock has three other mechanisms which help with understanding and reducing differences between the theoretical maximum found by the linear solution and what is realistically possible in practical approaches to forest management.

1. Use combinations of `_INVENT`, `_IF`, `@YLD`, and possibly other keywords to create fixed penalties for each use of an action, such as setting minimum administrative costs on conducting timber sales or covering equipment move in and move out costs, which can encourage more realistic harvest scheduling. This technique can work best when combined with supporting GIS work to group areas operationally and break larger stands into smaller strata for Remsoft to work with. However, reworking ODF's stand polygons is not expected in this course.
2. Use Woodstock's sequencing and crew movement features to generate cost for small and scattered harvests. As such development of move sequences is intended primarily for operational and tactical planning by landowners who run their own harvest operations it is not considered here.
3. Use Woodstock's spatial optimizer to assess the extent to which the linear solution is spatially feasible. Unlike linear solvers, which deterministically locate the global optimum solution of a linear problem, the spatial optimizer is a heuristic optimizer. It works by iteratively generating guesses at what might be a good spatial plan based on the linear solution. After each initial guess is made a computation phase is applied to search for incremental improvements.[1]

```
Lincoln BOF NoListedSpecies Upland Stand18571 GEN DF SiteClassImediumLow
Unthinned 83 3.69640489 aClearcut 3 _EXISTING ;A987 49.4% of class
Lincoln CSL NoListedSpecies Upland Stand18353 GEN DFRAWH SiteClassIlow
Unthinned 34 3.81675875 aFirstThin 2 _EXISTING ;A743 7.8% of class
Polk BOF NoListedSpecies Upland Stand19120 GEN DF SiteClassIIlow Unthinned
75 0.02191 aClearcut 5 _EXISTING ;A1767 100.0% of class
```

Figure 1: Example of a few harvest actions from a Woodstock linear solution. From top to bottom, Woodstock's chosen to clearcut 49.4% of a 3.696 acre polygon, thin 7.8% of a 3.817 acre polygon, and clearcut 100% of a 0.022 acre polygon. The resulting harvest sizes are 1.8, 0.298, and 0.022 acres, respectively. These choices let the solution push the absolute limit of what's possible under even flow constraints but are unlikely to fit well the process of offering a timber sale, bidding the sale, and executing the harvests. (`_EXISTING` indicates these are stands current inventory as opposed to the `_FUTURE` ones created by transition and reforestation activities following `aClearcut`.)

[1] In optimization terminology and in operations research, generating new solutions to a problem is called diversification and refining solutions is called intensification. Since different diversification-intensification approaches and balances work better or worse on various problems many different algorithms are in regular use. Remsoft does not document which methods they've implemented in the spatial optimizer.

This document is devoted to the third option, describing how to use Woodstock's spatial optimizer to assess the implementability of a linear solution. The most important result from such assessments is the reality check on the extent to which the outputs predicted by Woodstock's linear solution are achievable. For example, if the optimum linear solution generates annual revenues of US$ 4 million for the West Oregon District and the spatial optimizer estimates 80% of that revenue is possible, that suggests against committing to the strategic plan. Doing so would likely result in failing to meet the district's minimum revenue requirement. In more comprehensive forest planning than permitted in this course's limited timeframe, spatial optimization assessments might also be used as feedback to create more completely implementable strategic plans.

2 Estimate How Much of the Optimum Solution is Achievable

Woodstock's spatial optimizer is, not unsurprisingly, accessed through Woodstock. Since the spatial optimization process alters the Woodstock model, its shapefile, and generates various additional files and subfolders within the model folder, it's suggested to get familiar with spatial optimization using copy of your group's current Woodstock model. After making the copy, start Woodstock and open the model.

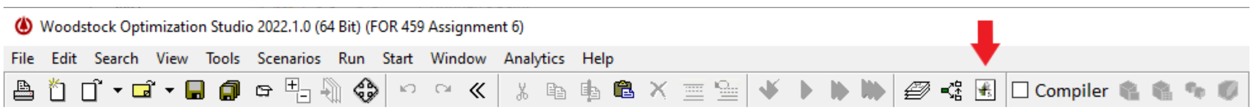


Figure 2: Location of the spatial optimizer button on Woodstock's toolbar. Even when the Remsoft License Manager obtains a Spatial Optimizer license Woodstock may fail to show the button and remove Spatial Optimizer from the maps right click menu. Restarting Woodstock while leaving the license manager running typically resolves this issue.

2.1 Task: Start searching for a good spatial solution

To run the spatial optimizer,

1. Start the spatial optimizer by clicking its toolbar button (Figure 2) or right clicking from the maps view and choosing spatial optimizer.
2. In the general tab enter set the range of periods to block to cover the full planning horizon (commonly periods 1..10 if using five year periods and a 50 year planning horizon), set the maximum number of periods a harvest can deviate from its scheduled time in the linear solution (two is a reasonable default), and set the most important outputs in the objective drop downs based on your management objectives.
    a. Blocking over the full horizon asks spatial optimizer to find the best overall approach rather than sacrificing future operability in favor of nearer-term results.[2]

[2] This consideration also applies to choice of planning horizon. Numerical optimizers, whether linear, heuristic, or based on other methods tend to exploit the model they're given. It's therefore likely that, for example, running a Woodstock model to 20 or 100 years will yield harvest plans clearly distinct from those obtained for a 50 year period. Assessing a model's sensitivity to different planning horizons before committing to specific harvest decisions is therefore likely wise.

   b. Since they repeatedly call into the linear model and tend to self-retract, the objective dropdowns are slow and often unreliable. Clicking on the left side of the dropdown arrow tends to be more successful than clicking on the right side.

3. The blocked activities tab will say "No activities found in attribute table." This is fine. Block size distribution constraints on the block distribution tab can be left unchecked as the distribution of block sizes does not need to be constrained.
4. In the Rule Set 1 tab click [Build…] in the Adjacent distance dropdown to open the Build Adjacency Tables dialog. Since clearcuts are restricted within 300 feet of each other by Oregon forest practices rules, enter 300 in the Distance box (or the equivalent if your Woodstock shapefile has units other than feet). After typing 300, press enter in the distance box to add 300 as a distance. The Calculate button will enable. Press it. It will likely take over minute for the adjacency to build.
   a. The upper part of the Build Adjacency Tables dialog does not render reliably. If there's no distance dropdown, try dragging or resizing the dialog. This issue affects several other Spatial Optimizer dialogs and the same workarounds apply.
5. After the adjacency build completes, 300 will disappear from the Build Adjacency Tables dialog. This is confusing but fine. Close the adjacency dialog to go back to Rule Set 1. Select 300 in the Adjacent distance dropdown.
6. Then
   a. Enter five acres as the default minimum block size.
   b. Enter 30 acres for a default target block size.
   c. Set the proximal distance to 300 feet as well.
   d. Enter five years as a default greenup delay (remember that this dropdown's units are in planning periods).
   e. Per the Forest Practices Act, enter 120 acres as the maximum opening size.
   f. If using planning periods in the range of 1–3 years, check allow multi-period openings. This lets the spatial optimizer do things like harvest a 30 acre stand one period and an adjacent 30 acre stand in the next period.
   g. In Rule Set Applies to Actions, check all clearcut actions.
7. Click the save icon or use File → Save to save this scenario. Give it an informative name, perhaps something like "periods 1-10 300 foot adjacency version 0.spf5".
8. Chose Run → Options.
   a. Change Stop Conditions to five minutes (select the minutes button to enable the dropdown).
   b. Check save multiple solutions. It's somewhat arbitrary, but setting the save threshold to 5% and one decimal in the solution name is a reasonable default.
   c. Click Ok.
9. Choose Run → Run or press the ▶▶ button.
   a. If needed, say Yes to adding Remsoft_id in the attribute table. This will add several other fields the spatial optimizer uses, like Action and Cut_period.

   b. The optimizer will spend some time looking for candidate polygons and prompt to build adjacency for zero distance. Say yes to building the adjacency.
   c. If you get warnings about objectives not applying to the actions chosen, cancel the run, resolve them, and rerun.
10. The spatial optimizer will add tabs for the objectives specified in the rule set and start running improvement phases.
    a. Switch to each output's tab (e.g. Figure 3, Remsoft calls these information output tabs) and set the desired flow tolerance. Since performing adjustments while the user interface constantly redraws as optimizer runs is difficult it's often easiest to pause the run, make a change, and then resume the run.
    b. If Woodstock is drawing the shapefile, switching back to the maps view in the main Woodstock window (the main window unlocks after candidate polygons are found) and selecting one of the newly added attributes, such as Cut_period, will cause the map to be periodically updated with the solutions the optimizer finds.
11. Monitor the spatial optimizer as it runs.
    a. The percent of optimal will vary from one optimization pass to another (note that the allocated bars periodically reset and the improvement phase counter restarts) but a best result of 85–95% is often achievable on relatively simple outputs with even flow.[3]
    b. The percent impossible should be near zero (these areas the optimizer cannot group with other areas as they have no adjacent areas).

If spatial optimization achieves only a low percentage of optimal this indicates a problem, such as lack of sufficient slack in the Woodstock model to accommodate spatial constraints in its implementation, something not being set up correctly within spatial optimizer, or possibly inability of the spatial optimizer to derive a good implementation from the linear solution. In general, spatial optimization's goal is to reach a good percentage of optimal and gain an understanding of how model outputs are influenced by spatial constraints. Slacking off constraints (max deviations, minimum and target block sizes, greenup delay, maximum opening size, flow tolerance, weights…) and observing effects on precent of optimal achieved can identify binding constraints which limit implementation of the linear solution. It may also be useful to use unrealistic values (such as a minimum block size of zero, maximum openings larger than 120 acres, or no greenup delay) in investigating how the optimizer performs.

Spatial Optimizer runs can be stopped at any time by pressing the stop button. If something's obviously wrong or a run's clearly not going to be an improvement, there's no point in waiting. Stop the run, adjust parameters, and start another run. Absent an intervention, the run will stop at five minutes as configured in the steps above. Because the save multiple solutions option was turned on, spatial optimizer will keep track of the best solutions it finds (the best plans or, equivalently, the highest scoring harvest sequences as indicated by the objectives and weights chosen).

[3] This is not a statement that 85–95% should always be achievable. More complex outputs, lower linear model slack, and problems with optimizer configuration (see Section 3) can all result in lower values.

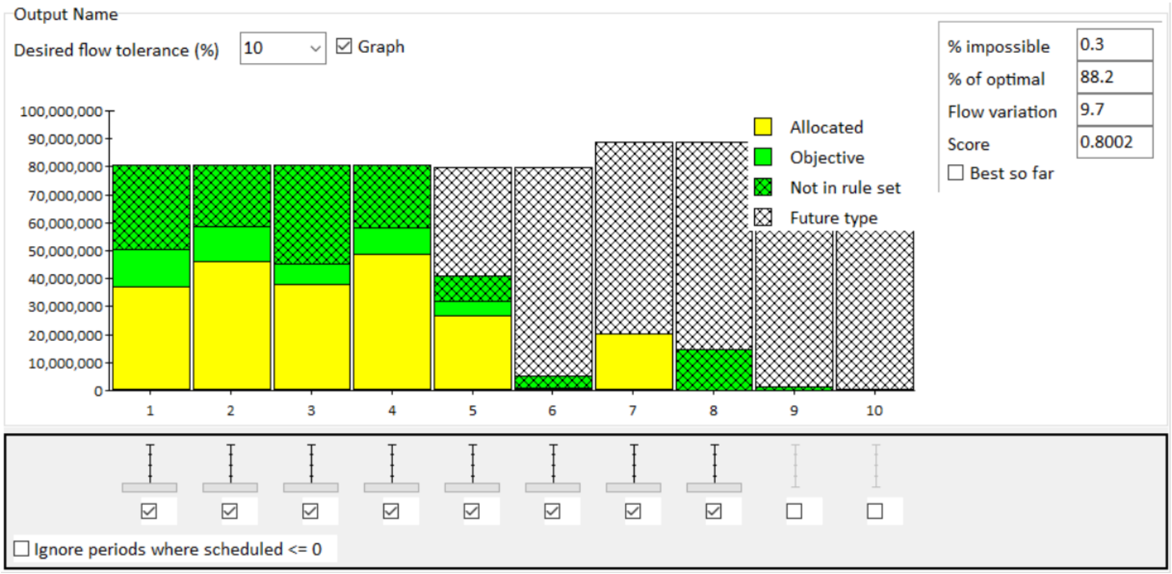


Figure 3: An example from a simple Spatial Optimizer run with a single objective—net revenue—whose configuration permits a relatively high percentage of optimal under a fairly tight flow tolerance. The checkbox and slider below each period's bar control whether Spatial Optimizer considers the period and, if so, the weight given to the period. Checkboxes and sliders are reset each time a Spatial Optimizer run is started. Adjusting sliders often has little effect but unchecking later periods can make Spatial Optimizer iterate through solutions substantially faster.

Each tab's graph (e.g. Figure 3) distinguishes primarily between stands in inventory at the start of the model run (allocated, objective, and not in rule set; green and yellow parts of the bars) and future development types (white), which is Remsoft terminology for areas which have had an action performed on them since the start of simulation. Since spatial optimization applies only to the unhatched portions of the bars, the percent of optimal which Spatial Optimizer reports is the ratio of allocated to objective. Not in rule set refers to the portion of the output produced by actions which aren't checked in the Rule Set Applies to Actions box. What percent of optimal means therefore depends on the model's actions and the rule set's configuration but, for FE/FOR 459, most commonly it refers to the fraction of the linear solution's clearcuts which the Spatial Optimizer is able to schedule. Since greenup constraints do not apply to thinning, typically the not in rule set fraction of the output is generated by thinning actions.

2.2 Task: Refine the spatial optimizer's search

The spatial optimizer's parameters (maximum deviation, selection of objectives, adjacency distances, block sizes, and so on) all work together to define what the best implementation of management objectives is. Perhaps the most important parameters are weights assigned to the objectives on the general tab. These are located to the right of the objective dropdowns and default to one (Figure 4). Giving one objective a larger weight relative to other objectives means the optimizer gives that objective greater management priority than other objectives. Conversely, reducing an objective's weight makes it less important. Setting the weights thus helps steer the spatial optimizer's heuristic searches towards the most valuable solutions.

Woodstock's spatial optimizer, like most heuristic optimizers, lacks well defined stopping criteria. In general, heuristic optimizers tend to find reasonably good solutions within their first few improvement phases. Beyond that, as the heuristic continues to run it's likely to occasionally find incrementally better solutions. The longer it runs, the less likely finding a better solution becomes and the more likely it is that any improving solution will be only slightly better than other solutions which have already been found. A judgement call is therefore required to decide when a heuristic's far enough into diminishing returns it's not worth continuing the search. Where the stopping point is depends on how long it takes to evaluate the model being optimized, the amount of computing power available, and how important finding small improvements is. For this course, giving the spatial optimizer a few minutes to explore each parameterization it's given is a reasonable default. As a final parameterization is reached a somewhat longer run, perhaps 10 minutes, isn't a bad idea.

Figure 4: Location of the objective weights on the right hand side of the spatial optimizer's general tab. In this example, maintaining a steady clearcut area has been weighted to be give times less important than maintaining steady revenue, giving the spatial optimizer more latitude to even out revenue through thinning.

To see the solutions the spatial optimizer's found, use View → Saved Tables and change to the Stanley tab where solutions are listed by scenario name and score.[4] Checking one of the boxes will show that solution in the map view in the main Woodstock window (Figure 5). If you like the solution, clicking Restore on the Tables dialog's menu will prompt to save that solution into the shapefile. Since the

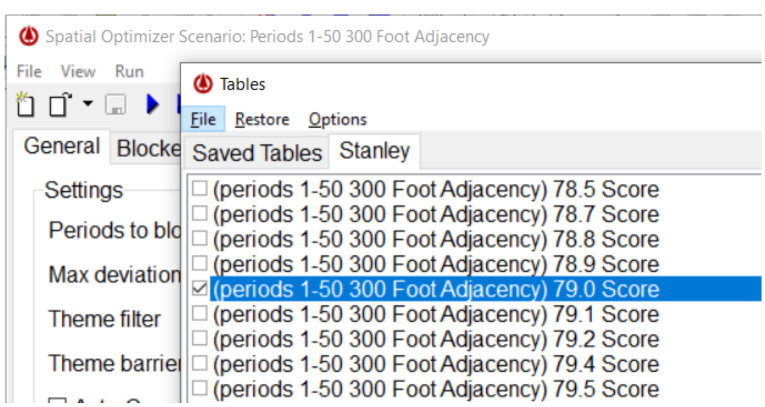


Figure 5: Picking a solution saved by Spatial Optimizer to review.

[4] Historical note: from at least 2004 up through Remsoft Spatial Planning System 2017, the spatial optimizer was called Stanley. Remsoft referred to it as Stanley / Spatial Optimizer in RSPS 2019, and then started calling it just Spatial Optimizer in Woodstock Optimization Suite 2020. However, as of 2023, Stanley still appears in many places in both Woostock's interface and documentation. Stanley is also used in some model subfolders.

spatial optimizer keeps track of these saved solutions by scenario name and score, it's best to use File → Save As… before starting a run with new parameter choices. If you just want to experiment and investigate possibilities, consider turning off saving (Run → Options) or using some working scenario name. When spatial optimizer's parameters are close to final, solutions can be saved from one or two of the most interesting parameterizations for later comparison if needed.

2.3 Task: Pick a most preferred solution from Spatial Optimizer

Once you're satisfied with the results you have, pick the one which appears to give the most attractive balance of tradeoffs and restore it into the shapefile as described above. After exiting Woodstock, the model's shapefile can be opened in ArcGIS Pro or QGIS and the Cut_period column used to identify and map the stands scheduled for harvest in the first five years of planning.[5]

When selecting a result, consider how the amount of clearcutting Spatial Optimizer achieves affects the ability to satisfy constraints and meet planning objectives. In some cases it may be necessary to adjust the linear program to provide more slack to accommodate opening size limits or other constraints, for example by increasing revenue requirements or relaxing clearcut area limits. The specific percent of optimal for each period is available by unchecking Graph in an output tab or choosing View → Reports from Spatial Optimizer's menu. Because spatially implementable actions become more uncertain as the amount of future type area increases over time, it's desirable to plan conservatively for clearcutting slack in later periods.

2.4 GIS data limitations

In reviewing Spatial Optimizer's choices, you're likely to notice awkwardly shaped harvest areas. Because Woodstock works with the shapefile it's given, the areas Spatial Optimizer can select to harvest are controlled by choices made in the GIS lab. The intersect-union-sliver elimination process used is effective at delineating manageable areas but, at the same time, exposes operational limitations (Figure 6). A more complete planning process than time permits within this course would likely take steps to address such limitations. These include

- Considering whether biodiversity, carbon objectives, aquatic anchors, or special management areas mean default riparian buffers should be widened to follow existing boundaries between riparian and upland stands.
- Breaking multipolygons into single polygons (using the Multipart To Singlepart (Data Management) tool in Arc or, equivalently, Vector → Geometry tools → Multipart to singleparts in QGIS) so operable areas separated by streams can be addressed individually and are accessible to sliver elimination processes.
- Creating a theme which groups management polygons together into operational areas and then using Woodstock's blocking features to make it aware that it should, for example, reach downslope to include part of an adjacent stand below a ridgetop stand that it's scheduling for harvest.

[5] For planning periods of lengths besides one and five years a proportional fraction of the stands can be selected.

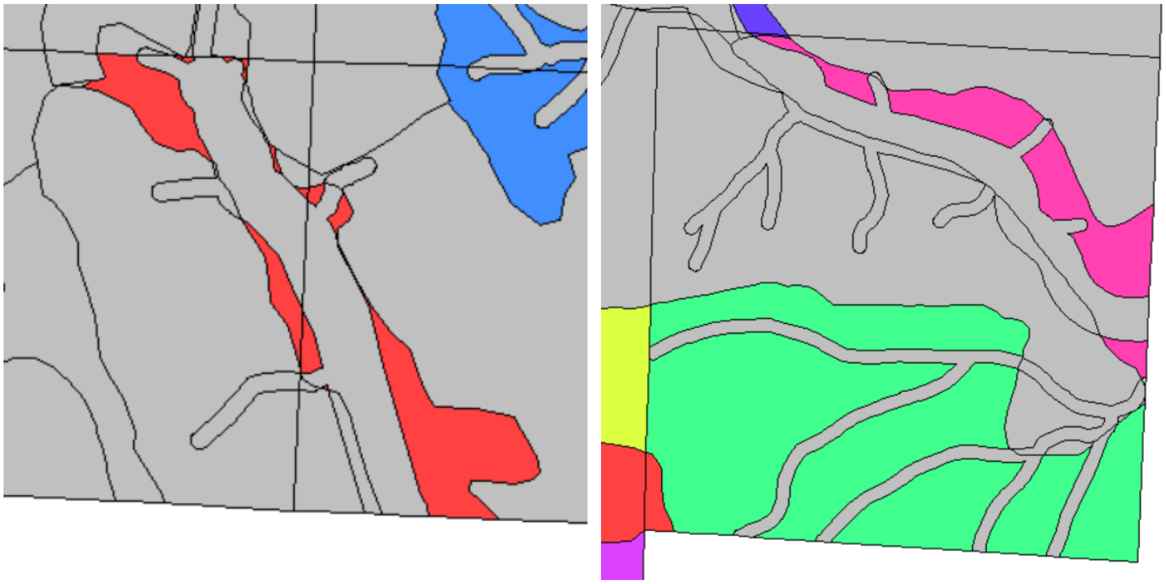


Figure 6: Example of harvests chosen by Woodstock's Spatial Optimizer which are operationally impractical. The red area at left is composed of fragments of a riparian stand which are far enough upland to be harvestable. Most likely these would absorbed into the adjacent larger upland stands during harvest unit planning. The pink area at upper right is another example of riparian fragmentation, running awkwardly across slope, being split by small streams, and having one area on the opposite side of the stream from the other areas. The green area at lower right is also problematic as it leaves downslope areas and fragments which will be difficult to access when the stand below it is harvested.

Since the stands scheduled for harvest in the first five years in this lab become the basis for selecting a stand for FE/FOR 469's project, having at least five operationally attractive areas to choose from is recommended as an additional criteria when selecting the most preferred solution among the ones provided by Spatial Optimizer.

3 Troubleshooting

3.1 Known Spatial Optimizer issues

- Woodstock, and Spatial Optimizer in particular, are prone to access violations. An access violation (or segmentation fault) indicates bug where code is trying to read or write from a location in the computer's memory which is not available. This indicates a program's in a bad state and, nearly always, the only option available to a user is to quit the app and restart. After Spatial Optimizer access violations it may not be possible to close Woodstock or force close it from Windows' task bar. Killing Woodstock from Task Manager may therefore be necessary.[6]

[6] Perhaps the most notable such failure observed in FE/FOR 459 was an instance where typing values into Spatial Optimizer dialogs was modifying the in memory copy of the solution's .pri file, resulting in the solution becoming invalid because Woodstock could no longer locate the actions section. In general, it's likely wisest to decline Woodstock asks about saving unexpected changes to any file after running Spatial Optimizer, especially if an access violation's occurred.

- If Spatial Optimizer can't work with an output or fails to find any feasible solution it will often display yellow dashes across the columns rather than yellow bars. This can be forced by inadequate slack in the linear model but, most commonly, results from overly restrictive even flow constraints on the objective. In particular, for bioconstraints (as Remsoft calls them) where management seeks to do things like maximize old forest area even flow is not desired and the Spatial Optimizer should be slacked off to 100%.[7]
- Spatial Optimizer tries to maximize an output's value and reports its percent success as a fraction of the linear solution. This is intuitive for even flow outputs and outputs where higher values are better. It's incorrect for outputs where lower is better and can be particularly problematic if the model's objective is to `_MIN` the output instead of `_MAX`.
- If an output violates its constraints it may not appear in Spatial Optimizer's dropdown list. Or it may appear but not be operable by the optimizer, resulting in yellow dashes or 0% or negative solution achievement. If this occurs, uncheck the Graph box on the output's tab to see the target values as Spatial Optimizer understands them—these may differ in the table view from what Spatial Optimizer displays in its graph view. It's possible, but unconfirmed, that Spatial Optimizer may have similar issues with outputs which have negative values or which cross between negative and positive.
- Woodstock can't be closed after using Spatial Optimizer. If none of the close button, closing from the taskbar, or File → Exit work, the next option is to kill Woodstock from Task Manager → Processes → End task.
- The starting random number is a seed value which the optimizer uses in guessing solutions. Such seeds are common and it's also common software which works with seeds allows specification of a fixed seed value. In most cases seeds should be randomized but, in special circumstances, fixing the seed value is a useful way to replay a previous run. [8] In Spatial Optimizer seeds are, by default, saved in the optimizer's scenario files and will be restored when the scenario is reopened. Whether or not starting a new optimization run generates a new seed depends on the state of the Auto-Generate starting Random number [sic] checkbox on the General tab as well as which optimizer tab is selected when the run is started. It's possible, though unlikely, poor optimizer results can be caused by bad luck with the seed. Clicking the dropdown arrow next to the seed will generate a new one.

---

[7] As of Woodstock 2022, Spatial Optimizer tends to perform poorly on such bioconstraints.

[8] A random number seed indicates use of a pseudorandom number generator. From a given seed, a pseudorandom generator produces a stream of numbers which appear random but are actually a deterministic sequence that will eventually repeat. There are many kinds of pseudorandom generators—subtractive methods and Mersenne Twisters are perhaps the most commonly used—with good implementations having output that's very close to truly random and repeat intervals well above 10^{10} numbers. Seeds are often taken from the current time, and thus also somewhat deterministic, but may also be obtained from cryptographic random number generators which provide truly random numbers. It's unclear how Spatial Optimizer obtains seeds but the apparent range of 1–9999 is uncommonly small and suggests the optimizer is designed to access only 10,000 sequences of pseudorandom numbers.

3.2 Spatial operations just don't work

Dialog boxes for spatial tasks in Woodstock may fail to appear. This can occur when attempting to run spatial utilities from the map and when trying to start Spatial Optimizer. The causes are unclear as of Woodstock 2023 but, once spatial operations start to fail, simply restarting Woodstock or restarting the machine may not be sufficient.

- One cause seems to be Remsoft License Manager obtaining a Spatial Optimizer license but not actually granting that license to Woodstock, even when many licenses are free to be assigned. In this case there may be a RemsoftLm.log file created, either in the license folder (open Remsoft License Manager by right clicking its taskbar icon and choose File → Explore license folder) or in the appropriate version directory under C:\Remsoft.
- A second cause appears to be corruption or confusion in a model's working files which causes silent failures when attempting to start spatial utilities or Spatial Optimizer. Recovery by deleting all Woodstock files except the model shapefile, .pri file, and the section files listed in the .pri has been successful. However, this does require rebuilding adjacency tables and risks accidentally deleting other files such as .xlsx versions of yield tables from FVS.