

4. FVS Module

Overview

This module, the second in a sequence of four workflow modules managed by the BioSum Manager software, connects the FIA forest inventory data imported into BioSum in Module 1 with the Forest Vegetation Simulator (FVS; Dixon 2002) and its Fire and Fuels Extension (FFE; Rebaun 2010). These models are developed and maintained by the USDA Forest Service, with detailed documentation and model software available at (<https://www.fs.usda.gov/fvs/>).

This overview begins with an extensive (9-page) discussion of how FVS modeling fits into the BioSum framework. A full understanding of this is essential to effective use of BioSum and the interaction between FVS and the BioSum framework is not entirely intuitive or obvious, owing to the complexity and flexibility of both components. If already familiar with building simulations in FVS, most of this chapter will be fairly accessible on a first read-through. Those who need to learn FVS will likely want to return to this discussion after learning to use FVS.

This chapter delves a bit beyond the FVS module because the decisions made in working through this portion of the BioSum workflow (e.g., how prescriptions will be set up, and timing arranged, in the FVS keyword control code that the user must develop to represent management options) have enormous implications for what will be possible to accomplish in later stages of the BioSum workflow. Consistent with the bridging role played by this BioSum module (between FIA and FVS frameworks), terminology shifts here towards what is standard in the FVS model; hence, the forested *conditions* loaded in module 1 (Database) become *stands* from the perspective of FVS simulation. An FVS simulation (referred to as a *run* in FVS) can be applied to some or all of the stands (*forested conditions*) loaded in module 1 to assess the effects of a series of one or more silvicultural prescriptions, typically scheduled over four 5- or 10-year “cycles”.

These sequences of silvicultural activities are referenced in this Users Guide, interchangeably, as silvicultural sequences or “packages” because they are, in essence, “package choices” involving a pre-specified sequence of what may be independently identifiable “treatments” occurring at different times along the silvicultural sequence. This opens the way for two fundamentally different approaches for representing silvicultural choices and their outcomes in a BioSum analysis, best understood via a simple example.

Suppose one wished to initially implement prescription Rx 100 – mechanical thinning across diameter classes to 75 sq ft residual basal area – treatment on a stand that has become overstocked relative to the size of its trees, followed by a separately described Rx 200 – mastication – treatment 20 years later to tamp down the ladder fuels that develop after opening the canopy. Under a “prescription assembly” paradigm, defining the thinning and mastication

activities as separate building blocks, let's call them Rx 100 and Rx 200, these could be assembled into a broad range of packages that could be developed. For example, Packages 1 (P001) could be constructed as Rx 100 (thin) at year 0 and Rx 200 (masticate) at year 20 and P002 as thin at year 10 and masticate at year 30. Mechanical thinning could be implemented in years 20 or 30, but there would not be time in a 4-decade simulation to implement the follow-on mastication 20 years later, so the analyst would need to decide whether to define those packages (which would model only the thinning, not the mastication during the simulation timeline). Moreover, Rx 200 might be incorporated into other packages that execute a q-factor prescription and Rx 100 might be combined in a package that refreshes and sustains treatment effectiveness with a broadcast burn.

An alternate, "chameleon" approach to the same silvicultural option pairing a thin across diameter classes with mastication 20 years later could define one package (P001) that contains one prescription (Rx 100) that is implemented before every growth cycle (at years 0, 10, 20 and 30 in this example). Typically, this prescription would be crafted to include some conditional trigger for when mechanical thinning takes place (e.g., basal area or stand density index exceeding a pre-specified threshold) that might or might not be true at year 0, but if not, might be in a future decade as the stand develops. Additional accounting variables and triggers could be specified in the prescription that implement the mastication treatment 20 years after the thinning (if still within the simulation timeline). It also opens the possibility of mastication being triggered not by a fixed interval following thinning but by a stand condition, e.g., based on a threshold canopy base height or trees per acre less than 5 inches in diameter, that might trigger the mastication activity to occur at 10, 20 or 30 years following thinning (or never).

There are valid reasons to select either approach to simulating management in BioSum, which will be discussed later, but each has advantages and disadvantages with respect to tracking of activities and timing on specific stands and the number of packages that must be defined (and simulated in FVS) to accommodate the options to be evaluated. BioSum's modular architecture around packages and treatments makes both approaches viable for maximal flexibility. The user will want to consider how important it is to know, based on the treatment ultimately selected in BioSum as best for a given stand, exactly what happened in that stand and when it happened. While that's easy under the prescription assembly paradigm, some work is needed in developing the accounting to make that possible under the chameleon paradigm.

Package effects, that is to say, the outcomes of applying the specified sequence of silvicultural activities (which may or may not involve the cutting of trees and in the simplest use case, might be scheduled to occur only in the first BioSum cycle), can be characterized in the Optimizer module. Within the structured workflow around which Optimizer was designed, the stand-level outcomes for each simulated silvicultural sequence can be assessed, and comparisons among sequences evaluated, relative to up to four stand-level descriptors generated by FVS or in the

BioSum framework and, if desired, conditional on economic characteristics of the sequence. Examples of stand descriptors include metrics of crown fire potential (e.g., canopy base height) or metrics relating to stand health and habitat value (e.g., measures of stand density or canopy cover, average tree size, share of trees of preferred species). Examples of economic characteristics include treatment costs and flows and value of wood derived from treatment. Moreover, when more than four attributes need to be considered, the authors have found it useful to construct composite stand indices that consider changes in multiple stand characteristics relative to objectives such as resistance to stand-replacing fire, which depends on both conveyance of fire into crowns, spread of fire among crowns and size and species mediated resistance characteristics of individual trees with respect to, for example, bark thickness.

The four stand attributes selected to represent the effectiveness of management towards one or more goals can be combined via Boolean logic to rate effectiveness as a multi-dimensional accomplishment. An additional attribute can be defined to select the best among multiple effective silvicultural sequences according to rules developed by the user.

In the simplest case (PrePost analysis), effects of treatments implemented at BioSum cycle 1 can be assessed by comparing immediate treatment outcomes to targets (e.g., was canopy base height below 30 feet before treatment and above 30 feet after treatment?) or by considering the magnitude of change (absolute or proportional, or change in classes) between pre- and post-treatment values (e.g., was predicted, severe weather, mortality volume as a percent of live tree volume improved by 25 percent or better?).

A somewhat more nuanced use case (weighted-mean) considers effects of a time-weighted combination of values of such attributes under treatment vs baseline (typically, but not always, grow-only) trajectories from up to 8 time points over the simulation, nominally labeled as Cycle 1 Pre and Post, Cycle 2 Pre and Post, through Cycle 4 Pre and Post. By thoughtful selection of the weights, this weighted mean can be designed to represent a mean value over time, a total, or to select a particular time point for comparison (by setting the weights for the other time points to 0). Stand attributes are estimated or predicted using both standard outputs from and custom-computations generated in FVS and FFE, or, for the adventurous, exogenously outside of the BioSum/FVS workflow via Python or R scripts or database queries. Analysts are strongly cautioned to invest focused effort in QA, testing and documentation of any attributes relied upon for effectiveness determination, regardless of the workflow used to develop them.

Why is it necessary to understand all these details when starting the FVS workflow? It all comes down to defining clear objectives for what answers are sought from BioSum so that FVS keyword control code can be crafted to generate FVS output that will make those analyses possible. Using the wrong timing of FVS cycles or failing to generate needed stand attributes in the FVS output can be a time-costly and frustrating exercise with few options for achieving a remedy short of going back to FVS to revise prescriptions or, in some cases, going all the way

back to one of the 1st steps in this workflow—generating an FVS input file from the BioSum project to serve as the basis for your FVS simulations. For example, if it is critical that seedlings, down wood, and/or growth, removal, and mortality information be included in the FVS input file and thus downstream FVS simulation settings and outputs, that is a choice that must first be made in the Database module when loading plots into the project via the FIADB, then also early in this module’s workflow.

BioSum supplies most of the required input data for FVS, including the StandInit (condition-level data) and TreeInit (tree-level data) tables, via an input database named “FVSin”, which BioSum loads into the *projectname*\fvs\data*variant* directories. BioSum generates FVSin databases as an SQLite database (FVSin.db). For legacy use, it also generates an MS Access formatted database (FVSin.accdb) via a different process that was only fully valid in the states of CA, OR and WA—the Access FVSin database is not recommended, not fully supported and is retained, for now, mainly to provide a band-aid process that permits loading of FIA compatible data in these states that are not part of the public FIADB, such as off-grid plots that are not part of the standard FIA stratification. FVSOnline accepts either database format as valid input. The FVS Suppose interface, now supplanted by FVSOnline, accepts only the Access format (FVSin.accdb).

BioSum also creates rudimentary “stub” keyword script (KCP) files (that should be thought of as templates to start from) containing instructions defining the time horizon of the FVS simulation, which FVS output tables should be produced, and BioSum-specific parameters for those output databases. A BioSum analyst well-versed in FVS—including its operation, internal functioning, and limitations— faces the not undaunting task of fleshing out these KCP files with scripting code (mainly keywords and associated parameters), for executing stand management activities at the appropriate times. It can’t be overemphasized that only a very rudimentary introduction to operating FVS is provided in this documentation. Analysts not deeply acquainted with FVS should enroll in FVS training (which is available via the Forest Management Service Center at Ft. Collins) or, at minimum, reading all available FVS documentation and completing FVS tutorials, which are available on the FVS web site (<https://www.fs.usda.gov/fvs/training/index.shtml>).

New users are strongly advised to carefully study the next section of this chapter (“It’s about time”) for some pointers concerning how to specify the time intervals that define FVS cycles before they edit the template KCP code into the prescriptions that they will model in FVS. Writing (and testing) FVS prescriptions is a demanding exercise for the uninitiated; we believe it’s essential to allow sufficient time to run quality assurance (QA) on this part of the BioSum workflow, and to run the simulations themselves—a process that can take days for large project areas and large numbers of silvicultural sequences. The FVS simulation work is typically the most time-consuming part of a BioSum analysis.

After silvicultural sequences have been modeled in FVS, you'll return to this module to extract data from the FVS output databases and load it into BioSum databases that convey it 1) to Processor, the module which estimates the costs of each silvicultural sequence applied to each stand and the volume and value of products that result from implementing these treatments, and 2) to Treatment Optimizer, which relies on both FVS outputs and Processor outputs to select the best management for each stand and estimate the wood produced, and associated costs and revenues.

It's about time!

The astute reader will have noticed that the 4 cycles are referenced above as generating management activity *nominally* (at least as we refer to timing in this Users Guide) at years 0, 10, 20 and 30 (when 10-year cycles are selected). In BioSum, the four time points are referenced this way (or as years 0, 5, 10 and 15 when 5-year cycles are selected). However, the “actual” years at which management occurs during the simulation, and that are reported out of FVS and loaded into BioSum, depend entirely on 1) the Timeint (and InvYear) parameters supplied when running FVS, and 2) how FVS output table records are assigned to the 8 BioSum Pre- and Post- cycle “time-points” (one can think of them as registers or a vector of values, that may or may not technically be “pre” or “post” treatment years, even though they are labeled that way) during the FVS Output load step at the end of this chapter’s workflow. This architecture is maximally flexible for the analyst but is admittedly confusing for those first learning BioSum.

In the interest of flexibility, BioSum is designed with the capacity to retain the inventory year associated with the FIA data loaded to the project (though note that “measurement year” MEASYR sometimes differs slightly from INVYR) throughout the analysis workflow for those who wish to do so. To support this capacity, BioSum assigns a sequence number to each year of simulation output when FVS outputs are loaded into BioSum (as described at the end of this chapter). It is usually less confusing to the user to instruct FVS to override the INVYR extracted from the FIA data by setting the InvYear FVS variable to 1 in the FVS control code so that all plots are on the same schedule of activities associated with a silvicultural sequence, regardless of the year in which they were intended to be measured in the field (i.e., the INVYR). Setting the FVS InvYear in this way results in a plot with an INVYR of 2020 and another with an INVYR of 2022 both entering the FVS simulation with an FVS Year value of 1.

As you will learn later in this chapter, the analyst can and should specify the length of each FVS simulation cycle for a BioSum analysis. It will likely not end well if FVS is allowed to choose a default timing for its projections, or if you specify a fixed, regular cycle length of 5 or 10 years. FVS output tables typically contain either pre-treatment attribute values or post-treatment values—only a few contain both. Evaluating what treatment has accomplished at a particular time requires simulation output immediately before and after treatment, and for most FVS table types, that can only be accomplished by inserting a 1-year projection between the “real”

projections. Even if conducting a weighted mean analysis (i.e., the mean value of outcomes across the 20 or 40 year simulation period), a 1-year projection is often required so that you will have representation of what a treatment has accomplished in the years immediately following its implementation.

An additional complication is that virtually all the tables produced via the Fire and Fuels Extension (FFE) of FVS (including POTFIRE, FUELS, CARBON and many others) contain no pre-treatment stand attributes if treatment is implemented in the first FVS cycle. Earlier in BioSum’s development, capacity was added to insert results from a “base year” simulation for the POTFIRE table output only, so as to include pre-treatment values for that table in a BioSum analysis. However, that solution makes the workflow more confusing—for example, the base year information gets inserted as year 1 and what was year 1 becomes year 2, year 2 becomes year 3 and so on, but only for that table, such that the timing does not appear to match other tables. Because of its complexity, this solution was not implemented for other Fire and Fuels Extension tables and this capability is now documented only in an appendix. The replacement solution to the problem, and the approach advised for all new BioSum projects, is to always simulate a grow only projection for 1 year before implementing any treatments. That way, your pre- and post-treatment years will likely be years 2 and 3 (for non-FFE tables) and years 1 and 2 (for FFE tables).

The tables below illustrate the timing choices using a set of FVS Timeint values that are mirrored in the last (Period Length) column of the FVS SUMMARY table output. The first FVS cycle is a grow-only cycle as indicated by 0 removed trees per acre (RTPA=0) and evidence of slow increase in basal area, quadratic mean diameter, and timber volume (TCUFT) over that 1 year as the stand age turned 48. The prescription implemented here thins the stand in year 2 and projects forward 1 year to obtain the post treatment summary attributes in year 3 that clearly show decreases in TPA, basal area, QMD and volume. Moreover, the after treatment basal area (ATBA) in year 2 shows the reduction caused by the thinning (a reduction offset in the next year of growth as basal area grew from 124 to 128). Note that year 2 values for all columns except the After Treatment (AT) ones, show only modest growth from the previous year, because the Summary table shows pre-treatment values (except in the AT columns). The third FVS growth cycle takes this stand forward 8 years to the end of the first decade at year 11. This stand is clearcut in year 12, which is the second BioSum cycle (the ATBA is zero).

FVS_Summary										
YEAR	AGE	TPA	BA	QMD	TCUFT	RTPA	RTCUFT	ATBA	ATSDI	PRD-LEN
1	47	457	245	9.9	6893	0	0	245	450	1
2	48	457	252	10.0	7153	139	3648	124	243	1
3	49	349	128	8.2	3615	0	0	128	254	8
11	57	296	159	9.9	5031	0	0	159	293	1

12	58	290	163	10.2	5151	290	5151	0	0	1
13	1	345	0	0.1	0	0	0	0	0	8
21	9	324	15	2.9	140	0	0	15	44	1
22	10	321	22	3.6	228	96	84	14	40	1
23	11	224	19	4.0	213	0	0	19	51	8
31	19	216	65	7.4	1250	0	0	65	134	1
32	20	216	73	7.9	1481	0	0	73	147	1
33	21	215	81	8.3	1647	0	0	81	159	8
41	29	208	145	11.3	3648	0	0	145	254	1
42	30	207	153	11.6	3899	0	0	153	264	0

The POTFIRE table, one of the several that are created by FFE and share a different timing format, shows outputs for the same stand. Timing is different on account of this table containing only post-treatment values. Year 1 shows the stand before treatment has occurred (because treatment was modeled at year 2). The effects of the treatment are obvious—the fire type under severe weather change from conditional crown to surface, total flame length drops, canopy density is reduced and predicted mortality under severe fire weather is greatly reduced both on the basis of percent basal area and cubic foot volume. Note that FFE tables like this POTFIRE table have one less year of output than other tables (the last 1-yr projection does not show up here). Note also that year 12 shows -1 for canopy base height—FFE’s code for a missing value, because having just been clearcut, there are no trees so canopy base height is not defined. When conducting QA, these are very important things to check for as you would probably not want a minus 1 to become part of a weighted average, for example. Interrogation of FVS output tables, such as FVS_Summary and FVS_Potfire, is critical toward ensuring prescription timing and parameters are occurring as intended by the analyst and produce logical values.

YEAR	SURF_FLAM E_SEV	TOT_FLAM E_SEV	FIRE_TYPE_SEV	PTORCH_SEV	CANOP Y_HT	CANOPY_DENSIT Y	MORTALI TY_BA_S EV	MORTALITY_VOL_SEV
1	4.5	64.0	COND_CRN	0.00	36	0.19	100	6893
2	5.7	5.7	SURFACE	0.01	38	0.09	20	667
3	5.8	5.8	SURFACE	0.04	38	0.09	20	682
11	5.9	5.9	SURFACE	0.00	44	0.10	16	756
12	7.9	7.9	SURFACE	0.00	-1	0.00	0	0
13	9.8	9.8	SURFACE	0.00	-1	0.00	100	0
21	7.0	10.3	PASSIVE	1.00	4	0.04	99	139
22	7.0	9.1	PASSIVE	1.00	4	0.03	99	143
23	6.8	10.0	PASSIVE	1.00	4	0.04	99	212
31	6.0	10.9	PASSIVE	0.99	8	0.06	99	1241
32	5.9	11.2	PASSIVE	0.99	8	0.06	99	1469

33	5.7	9.0	PASSIVE	0.98	9	0.07	98	1629
41	5.2	5.2	SURFACE	0.29	17	0.08	43	1546

As indicated earlier, some questions focus on the immediate accomplishments of management, and so call for structuring the timing of simulations and the loading of FVS Output into BioSum so that there are pre- and post-treatment values available in BioSum' Treatment Optimizer for evaluating treatment effectiveness, for a particular cycle or for all cycles. For example, FVS could be directed, as it was in the examples just shown, to treat in year 2, 12, 22 and 32, and report pre-treatment stand metrics (to be used in evaluating effectiveness) for these years and post-treatment metrics (from the Summary and other non-FFE tables) for years 3, 13, 23, and 33. A Treatment Optimizer scenario could explore treatment accomplishment in BioSum cycle 1 (year 3 vs year 2).

Other questions may be best addressed by comparing a weighted combination (e.g., average or total) of stand attributes over the course of several decades for one silvicultural sequence against another, or against a grow-only sequence. In that event, having FVS output representing pre- vs post-treatment time points may be less important. For example, FVS simulations can be crafted to implement treatment at year 1, and report stand metrics at years 2, 7, 12, 17, 22, 27, 32, 37 and 42, with each successive time of reporting what would be expected to be a decline in treatment effectiveness. Treatment optimization might then compare an equal weighted average across all 8 of these time points for each silvicultural sequence against a similarly weighted average for a grow-only sequence. Alternatively, one could weight the average so that the gain over grow-only at year 42 is the sole basis of effectiveness characterization (by setting the weight for year 42 to one and the weights for all other years to zero) if the only concern is the state of the forest at the end of 4 decades of management.

The take-home points here are:

- 1) Although the FVS module and this documentation that describes its use references years 0, 10, 20 and 30 to represent the 4 cycles accounted for in the BioSum analysis framework, in the FVS output databases, these may correspond to very different years, depending on the purpose and structure of the analysis.
- 2) While it may be tempting to launch into tasks with the BioSum manager, which are described in this chapter, it is very important to think through the analysis that you want to carry out so that you create an FVS input file and FVS keyword code that will support those objectives. Taking the time to understand how BioSum operates (e.g., by carefully reading this Users Guide in full) and, if possible, experimenting with the starter project before progressing through an intended BioSum project, is essential to ensuring a valid and useful analysis.

Assigning "Additional CPA (cost per acre)" that apply to specific treatments

There's one more topic to introduce before launching into the FVS Module workflow, because it surfaces in several places in the BioSum workflow and, like timing, needs to be planned for when developing KCP codes. BioSum's Processor module (next in the workflow sequence) provides for calculating, for each stand, a per-acre cost of the harvest activity associated with a silvicultural treatment at a given time using the OpCost model – a script processed in R that transforms summaries of a cut tree list into machine time requirements, and ultimately, harvest costs per acre for that stand and treatment at that time. It is often necessary or desirable to also account for what BioSum labels as "Additional CPA" because they are over and above the costs modeled for the tree harvest, yarding, loading, chipping and mobilization/demobilization expenses accounted for by OpCost. Examples of costs that that may need to be tracked include those associated with surface fuel treatments (such as mastication, prescribed fire or pile burning); mechanical site prep; application of herbicide, fertilizer, lime or biochar; water-barring skid trails; and planting or seeding. Many of these are not harvest costs at all, but rather, costs of activities that are part of a prescription that may or may not include removal of trees and which may be incurred at the time of tree removal, or at another time. For example, a management trajectory that thins a stand in year 1 may return to implement a prescribed fire in year 21 of the simulation via the SIMFIRE keyword. In that case, there is no harvest cost in year 21 (because no trees are harvested), but there is an Additional CPA – the cost of the prescribed fire.

By thoughtful use of KCP scripting, these additional costs can be modeled as contingent on information generated during the FVS simulation of the prescription for a given stand—for example, pile and burning might be considered necessary when activity fuels (down wood resulting from thinning) exceed some threshold, say 15 tons per acre, but not otherwise. If FVS is coded to implement a pile and burn only when that threshold is exceeded, then the cost assigned to pile and burn should only be incurred if FVS calculates activity fuels as > 15 tons/ac and conditionally implements the pile and burn treatment.

Accounting for Additional CPA requires intervention at three points in the BioSum workflow, chronologically: 1) when the prescription is defined or edited in the FVS module of BioSum workflow (accessed via the "Rx" button), 2) in the development of the KCP code that supports the simulation of that prescription in FVS, and 3) in the Additional CPA tab of Processor. Adhering to the guidance for these three workflow steps results in Treatment Optimizer generating the 'Additional_kcp_cpa' table containing a breakout of additional harvest costs and including these additional costs in the overall onsite harvest cost and net revenue calculations in the Processor and Optimizer modules.

Module Components

Finally, we are ready to learn the workflow of BioSum’s FVS module! This module includes six component tasks, all of which should be completed before proceeding to Processor: (1) Plot FVS Variants—review FVS variant and location code assignments for each plot, filling in values for plots not yet assigned variants, via an audit that draws from a reference database table, (2) Rx—define or edit each silvicultural prescription (treatment) that may be applied as part of one or more silvicultural sequences, (3) Rx Package—define or edit each package of treatments (a.k.a. silvicultural sequence) to be applied over up to four BioSum cycles, (4) Tree Species—inspect, audit, edit and assign FVS trees species codes to FIA trees for all species present in the project, (5) FVS Input Data (FVSiN)—create both the input database (containing stand and tree lists) that will be used during FVS processing and the BioSum_Keywords.kcp (keyword control program) that will, when fleshed out by the analyst, control FVS operations, (6) Create FVSOUT MDBs – partitions FVS output data into individual Access databases by variant and RxPackage combination, and finally (7) FVS Output Data (FVSOut)— retrieve FVS tabular output and cut list tables into BioSum project databases, to feed to the Processor and Treatment Optimizer modules.

Plot FVS Variant

FVS simulations rely on equations and computation code that differ among geographic areas, known as “variants” in FVS parlance (Figure 4.1).

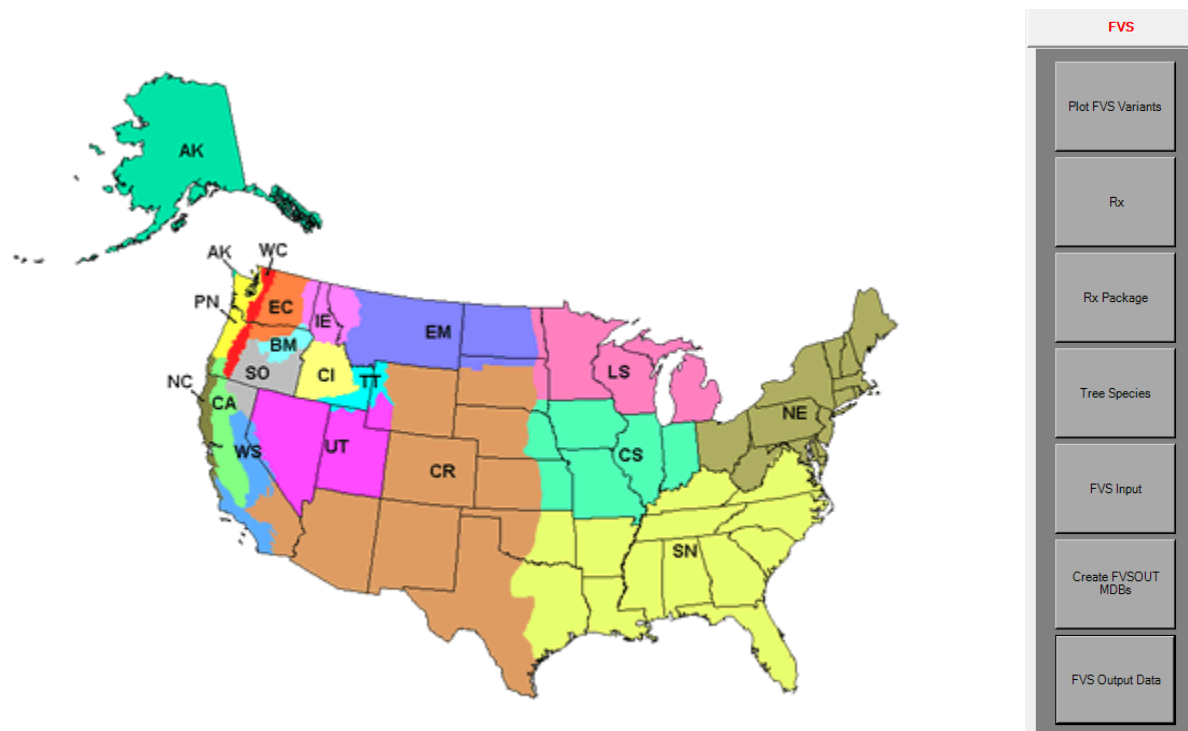


Figure 4.1 – FVS variant map. Source: <https://www.fs.usda.gov/fvs/documents/overviews/FVSVariantMap20210525.zip>

In general, the equations, such as those that predict diameter and height growth, were developed using growth data from trees within a variant and “location code”, a finer-scale geographic unit. Surface fuel model assignment, site class calculation, mortality estimation, crown ratio and crown width parameters, and fuel moisture content assumptions are a few of the many other items that may differ among variants and location codes. It is very important that each plot, and the stands (conditions) that it contains be assigned to the variant and location code in which it geographically resides. The *fiadb_fvs_variant* table in the *ref_master* database contains the correct variant assignments and location codes (typically a concatenation of NFS Region and Forest numbers) for all plots in the FIA database in the states of CA, OR, WA, ID, MT and UT, which makes it easy to auto-assign the right values.

1. To assign the appropriate variant and location code to plot records or to review the current assignments, select **<Plot FVS Variants>**.
2. The **Plot FVS Variants** window will appear, with a list of plots and their existing *fvs_variant* and *fvsloccode* assignment in sienna-brown, editable columns. This table can be browsed, sorted (indexed) or filtered, via functions available from the right click menu when right clicking while the cursor is placed in the desired column. For example, one could specify a particular value to filter on in the county column (Filter by Entered Value) to list all plots in that county, or “filter by entered value” the *fvs_variant* column (leaving the entered value blank) to list all plots that lack a variant assignment. However, it is easier to find cases of plots that lack a variant assignment by running an “audit” on the plot. The prudent analyst will run audits whenever they are offered in the BioSum workflow to identify potential issues that, if ignored, could corrupt or invalidate the BioSum analysis.

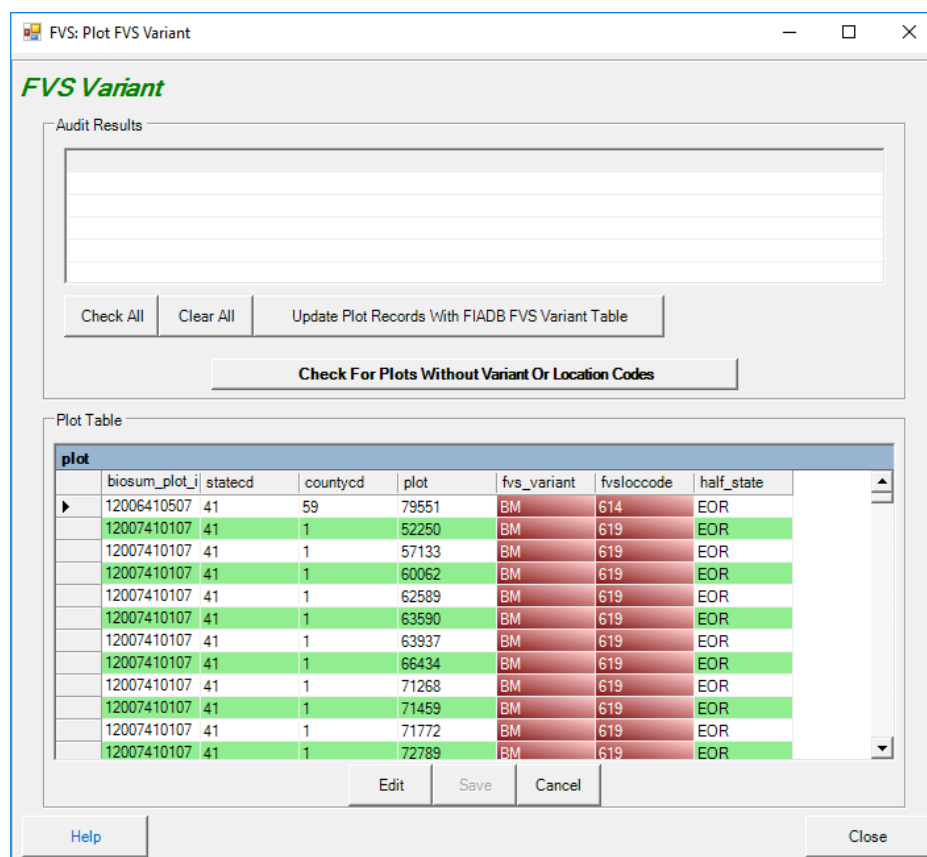


Figure 4.2 – Plot FVS Variants window containing a list of plots and existing FVS variant and location code assignments.

- To run an audit, select the button labeled **<Check For Plots Without Variant Codes>**, which initiates a script that identifies plots without a variant assignment. A message box will indicate whether the audit passed or failed, and if it failed, those plots without variant assignments will appear in the table in the upper half of this task window, provided they exist in the `fiab_fvs_variant` table in the database
`C:\Users\username\AppData\Roaming\FIABiosum\biosum_ref.acddb.`
- Select the records to be updated using the **<Check All>** button, or manually toggle the checkbox for individual plot records. After records are selected, update the plot table using the **<Update Plot Records With FIADB FVS Variant Table>** button to automatically make variant and location code assignments based on the values in the `fiab_fvs_variant` table in the `ref_master` database. Inspect and sort the `fvs_variant` column of the Plot Table to make sure that `fvs` variants have been assigned for all plots. To confirm that all plots have been assigned a variant, the audit can be re-run by selecting **<Check For Plots Without Variant Codes>**.

5. If plots exist outside the states for which the `fiadb_fvs_variant` table contains variant assignments, it will be necessary to manually assign values to replace NULL values. It may be helpful to consult the FVS website (<http://www.fs.fed.us/fmnc/fvs/>) for information on assigning an appropriate variant code. When location code is omitted, FVS will revert to the default location code for the variant, which may or may not result in valid projections for those stands.
6. Click **<Save>**, then **<Close>** to save the FVS variant assignments and exit the task window.

Rx (Prescriptions)

The next step is to define and label the silvicultural prescriptions (treatments) to be simulated in FVS. We refer to treatments as “Rx” throughout this guide. Treatments ultimately consist of a file of keywords and parameters provided to FVS to guide what happens at a particular time. For example, one might define treatment 200 as “thin across diameter classes (a constant harvest proportion applied to all diameter classes) to a residual density of 85 sq. ft basal area per acre, while cutting no trees larger than 36” in diameter, using mechanized whole tree harvest on gentle slopes and cable manual logging on steep slopes”. The harvest systems are not specified in the KCP file, but the implications of their use are, for example, via “yardloss” keyword statements indicating the quantity of wood residues left behind. To model a different residual density or upper diameter limit, you would create additional prescriptions (e.g., 201 with residual basal area=100, upper diameter limit 36”, etc.). Sometimes two treatments will be identical except that one uses whole tree harvest systems and captures all or most forest residues for utilization (and/or piles them at the operation landing) while the other relies on a log length system that brings only merchantable logs to the landing, leaving residues in the forest, possibly to be treated later at additional expense. Treatments may also include activities that take place in the stand that are not harvest *per se*, for example, piling and burning or masticating harvest residues; however, these are usually specified as supplemental (to tree harvest) activities that incur additional harvest costs specified by the user. Once treatments are defined, they can be assigned to one or more prescription packages, which are simply sequences of either the same or different silvicultural prescriptions, implemented over 4 cycles. Note that some cycles in a package may have Rxs that are essentially “grow-only” and harvest no trees. By convention, it can be useful to define Rx 999 as grow-only, for repeated use in a grow-only silvicultural sequence.

Prescription definition workflow is as follows:

1. Define silvicultural treatments to be applied in FVS by clicking <Rx> to load the Rx task window.
2. For each treatment to be added to the project:
 - a. Select <New> to open the **Treatments** window, which will display four tabs: Treatment, Harvest Method, Harvest Costs, and Associated FVS Command(s), which we explain below.
 - b. **Treatment:** After selecting the treatment tab, choose a category, sub-category, and ID number. Each treatment must be assigned a value between 001 and 999. Treatment ids 001 to 699 represent pre-defined treatment categories, and treatment ids 700 to 999 can be used to label any custom-defined treatment. However, compliance with these conventions is optional—any prescription number can be assigned to any prescription. After choosing an Rx number for the new Rx, enter a brief description of that treatment. These descriptions usually prove to be far more important and useful than compliance with pre-defined treatment categories. Later in the BioSum workflow, these descriptions may be essential to help recall what the Rx was intended to accomplish. Apply your edits by clicking <Select>.



The screenshot shows the 'FVS: Treatment (New)' window with the 'Treatment' tab selected. The window includes a toolbar with buttons for OK, Cancel, Properties, New, Edit, Delete, Clear All, and Open. Below the toolbar are three tabs: Treatment, Harvest Method, and Additional CPA. The Treatment tab is active and contains the following elements:

- Category:** A list box containing 'Traditional Thinning 001-299', 'Fuel Treatments 300-499', 'Other Thinning Operations 500-699', and 'General Custom Defined 700-999'. 'Fuel Treatments 300-499' is selected.
- Sub-Category:** A list box containing 'Thin From Below 300-324', 'Thin A Species Across A DBH Range 325-349', and 'Thin With Fuel Piled And Burned 350-374'. 'Thin From Below 300-324' is selected.
- Available Id's:** A list box containing numbers 300 through 309. '300' is selected.
- Treatment ID:** A text box containing '300'.
- Category:** A text box containing 'Fuel Treatments'.
- Sub-Category:** A text box containing 'Thin From Below'.
- Description:** A text box containing 'Thin from below: basal area target 90.'
- Packages that use this treatment:** An empty text box.

Figure 4.3 – Treatment tab of the FVS:Treatment window. Treatment category, sub-category, ID #, and description are assigned at this stage.

- c. **Harvest Method:** Choose the Harvest Method tab. Select a harvest method for low slopes and a harvest method for steep slopes. The percent slope threshold above which slope is categorized as steep will be specified later in the Processor module of BioSum (chapter 5). Once a harvest method is selected, a brief description of the method will appear in the Description text box. Refer to **Appendix A** for a list and description of harvesting methods available for selection.

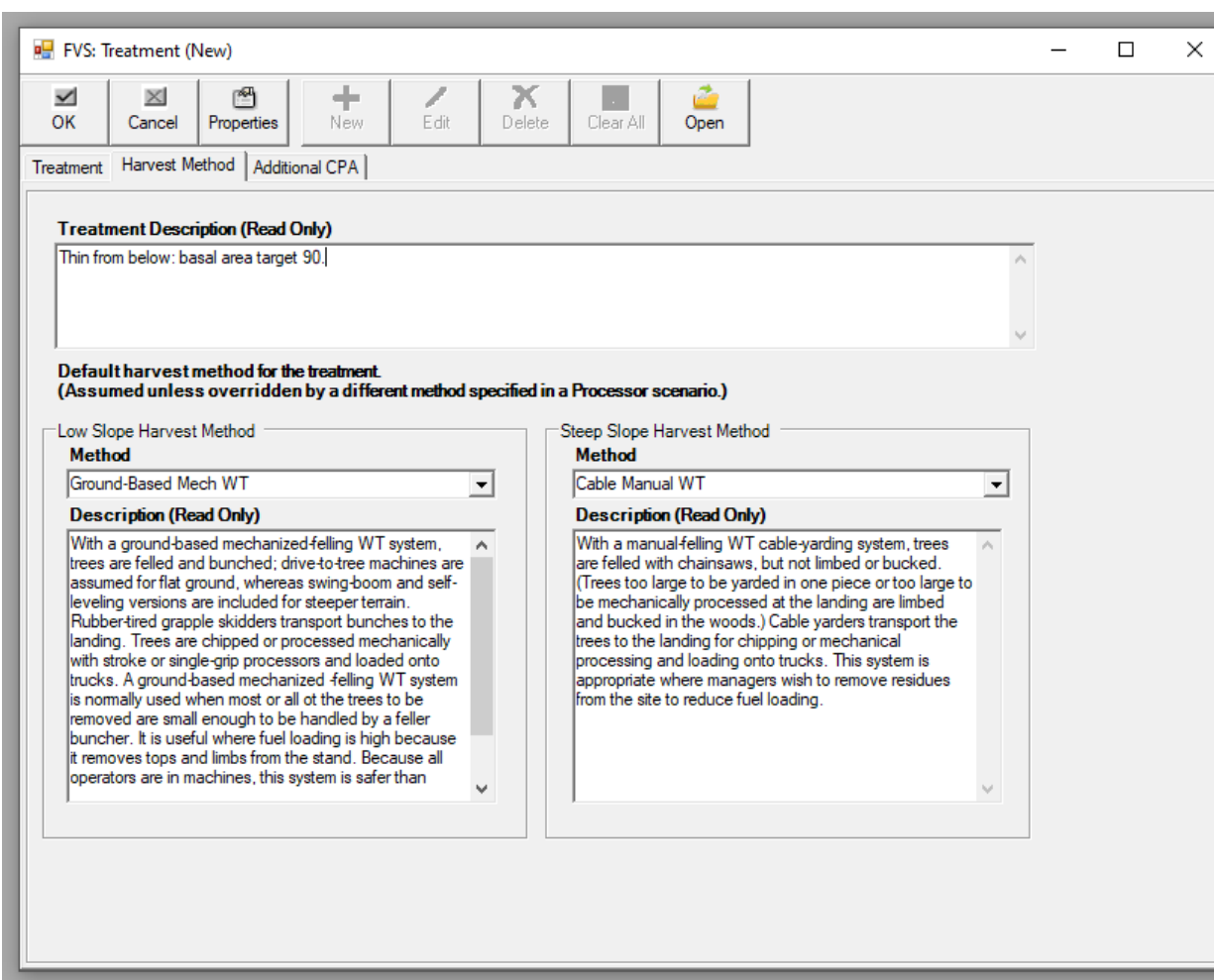


Figure 4.4 – Harvest Method tab of the FVS:Treatment window. Select harvest methods for low and steep slopes.

- d. **Additional CPA:** Add any additional cost components – i.e., those that are not accounted for by the OpCost model, such as the cost for surface fuel treatments (e.g., mastication, prescribed fire or pile burning); mechanical site prep; application of herbicide, fertilizer, lime or biochar; water-barring skid trails; and planting or seeding, as discussed earlier in the module Overview. Define a cost component by selecting <New> and entering the name for the cost component and a brief description in the window that pops-up. The analyst will subsequently need to set an activity flag (that will be passed along in FVS’s COMPUTE table) with the same name as this cost component in the FVS KCP script that controls the simulation. To ensure compatibility with FVS scripting, the name/flag needs to be no more than 8 characters, start with a letter, and avoid spaces and special characters, so choose ‘RXFIRE’ rather than ‘Rx Fire’ or ‘RX_FIRE’. The actual

costs (in dollars per acre) of these components will need to be assigned in the Processor module (Chapter 5) and will be incurred in the accounting for management costs when the flag is set in the FVS simulation.

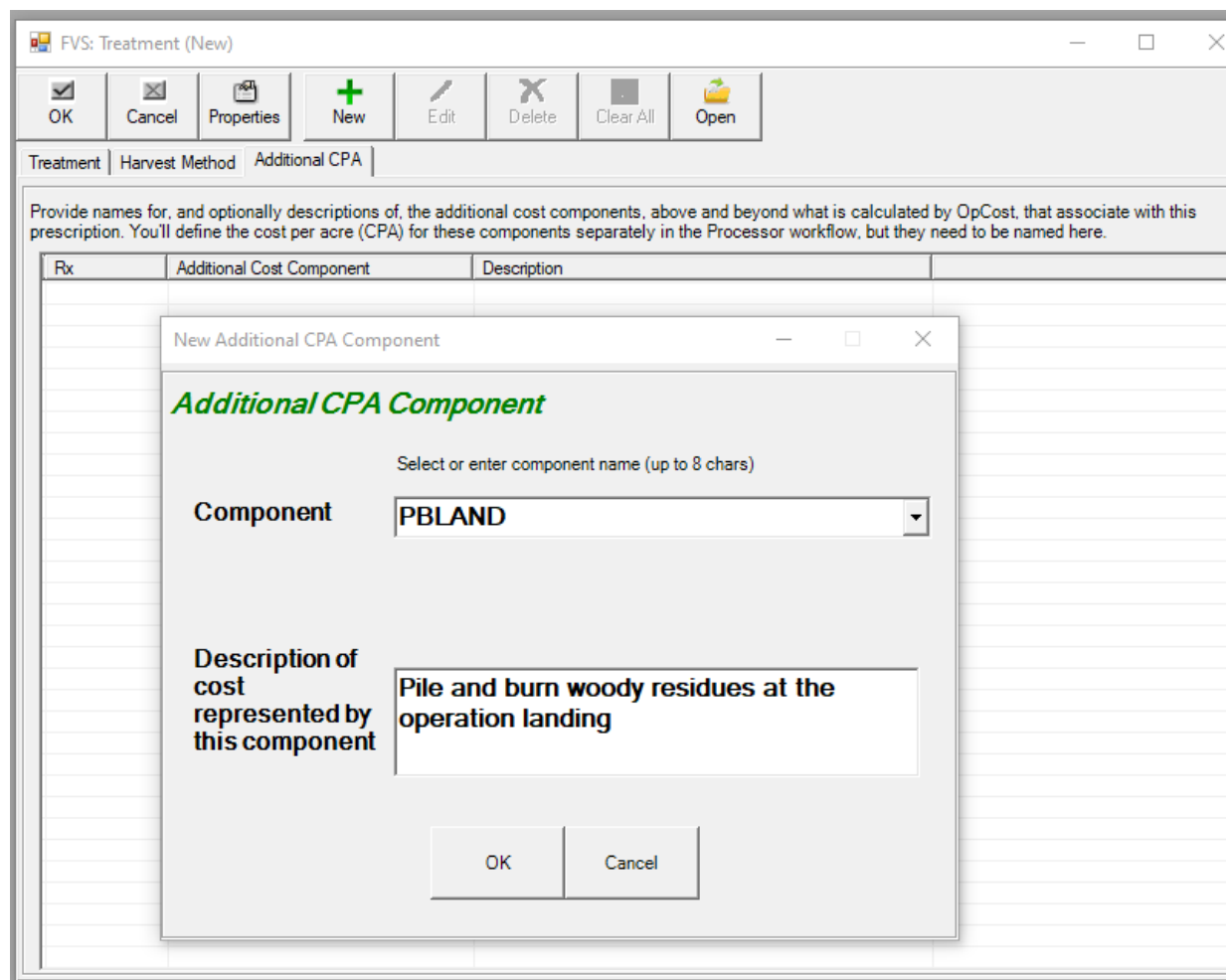


Figure 4.5 – Additional cost per acre (CPA) tab of the FVS: Treatment window. Add or edit any additional cost components associated with this treatment.

- e. When all desired information has been entered (treatment and harvest method are required; Harvest Costs is optional), select <OK> in the upper left corner to return to the Treatment List. Your treatment will not be saved in the project until clicking <Save> at the bottom of the treatment list.
3. To make changes to a treatment already displayed in the treatment list, select it (it will show as highlighted) and then click <Edit>.
4. When finished creating or editing treatments, click <Save>, then <Close>, to exit the **Treatments** window.

Note that one can add new prescriptions (and silvicultural sequences) to a project at any time; it is not uncommon to think of new prescriptions to try after seeing how those that were first modeled play out. You would simply come back to this module, define the new prescriptions and sequences, run them through FVS and load the FVS output from those sequences back into BioSum and proceed to Processor to process them.

What is a Silvicultural Sequence (a.k.a. RxPackage), conceptually?

Given that FVS is a very customizable model, there are more ways to design and conceptualize a simulation than can possibly be described in this Users Guide. Some thought must be given to what the meaning an analyst intends as by a silvicultural sequence, management alternative, prescription package, or whatever they choose to call a sequence of potential activities over time. BioSum expects to see FVS output that represents stand attributes of interest to the analyst at 8 time points—for example, Pre and Post management activity at each of 4 cycles of management opportunity (whether or not management activity actually occurred at those times). Thus sequence 001 could be defined as implementing Rx 250 at each of 4 cycles (Repeat Every Cycle); however, Rx 250 might be defined in the KCP file to conditionally cut trees only when a minimum basal area is exceeded, for example, in which case trees might be cut at only some, one or none of those cycles (Repeat Every Cycle Contingent). Or, sequence 001 could be defined as implementing Rx 250 at cycle 1 and not in the other cycles, sequence 002 could be defined as implementing Rx 250 only at cycle 2, etc. (Specific Cycle). Alternatively, sequence 001 could be defined as implementing Rx 250 at cycle 1 or any cycle at which some stand attribute threshold is reached, then implementing a different, “tune-up” treatment Rx 300 (perhaps one that cut no trees but implemented a prescribed fire to clear out ladder fuels arising from regeneration) 1 or 2 cycles after implementing Rx 250 (Mixed Rx Contingent). What a silvicultural sequence means then, and how results can be interpreted, are very much interdependent. For example, when comparing the effectiveness among treatments at cycle 1, a Specific Cycle “sequence” not timed to occur at Cycle 1 may never be effective (if effectiveness is judged at cycle 1), while a Repeat Every Cycle Contingent sequence may be effective in stands where actual treatment was triggered at cycle 1 and not for other stands where no actual treatment occurred at cycle 1.

There is no one right answer to the question posed in the title of this section—different choices make sense for different purposes. However, analysts are advised to carefully consider and document the intention and definition of every sequence modeled and to keep these in mind when interpreting BioSum outputs later on.

Defining a Silvicultural Sequence (Rx Package) in BioSum

A silvicultural sequence (or Rx [prescription] package) is a series of treatments or management activities applied to a stand either at one time or over multiple times during the four, 5- or 10-year cycles of a BioSum analysis. In the Treatment Optimizer module, these silvicultural

sequences can be evaluated and compared to one another and/or to a grow-only (no treatment) silvicultural sequence, depending on analysis objectives. An Rx package consists of a sequence of one or more of the Rxs already defined, with the possibility that some cycles will have no Rx assigned or will assign a prescription that is defined as grow only (these have the same net effect but result in different labeling of the project databases).

1. Assemble a sequence of silvicultural prescriptions to be applied in FVS by clicking **<Rx Package>** to load the Treatment Packages task window.
2. To create a new package:
 - a. Select **<New>** to open the Treatment Package Item window, which has two tabs: Package and Harvest Costs.
 - b. **Package:** To create a new package:
 - i. Select an available package ID number from the **<Package ID>** drop-down list. It is usually easiest to start at 001, and increment from there as new Rx packages are defined.
 - ii. Write a brief description of the package in the text box next to **<Description>**.
 - iii. Next, specify whether your Rx package will utilize 5- or 10-year cycles by selecting the radio button next to the cycle length of choice in the **<FVS Cycle Length>** box. (Note: all packages within a BioSum project must have the same cycle length to generate valid results)
 - iv. To add a treatment to a cycle year, select the year: 00, 10, 20, or 30, and click **<Edit>**.
 - v. Uncheck the box next to **<Skip Treatment>** (by default, each cycle will be populated with a “skip treatment” via the check box, which will assign a null Rx coded as 000).
 - vi. Assign a prescription from the dropdown list next to **<Rx ID>**. You will see a list of prescription ids that have been defined for this project in a dropdown list; select one to assign. If a grow-only prescription was formally designated earlier as 999, that choice will be available here also.
 - vii. Select **<OK>** to add the treatment to the package. The selected treatment ID will appear next to the desired year along with its description.
 - viii. To add another treatment to the package, return to step iv. If no treatment is to be administered in a cycle year, no action is necessary; BioSum will assume no treatment as indicated by the “skip treatment” box being checked. However, it is equally valid, and perhaps preferred, to assign a grow-only (e.g., Rx 999) if treatment is not planned, and doing so may help avoid confusion over whether a treatment was inadvertently omitted when the sequence was defined (causing a 000 skipped treatment code [that appears as null in the BioSum interface] that will persist in all

databases and kcp files associated with the package). **One very important point to emphasize** is that BioSum pays very close attention to these codes and assumes that a skipped treatment means just that. It is not adequate to enter a Rx for cycle 1 and leave skipped treatment checked for the other cycles. Even if the FVS kcp file is structured as Repeat Every Cycle Contingent (as described above), and FVS generates a cut list in cycle 3 (because the contingency is met), BioSum will not read that cut list data from the FVS output if the Rx coded for cycle 3 is 000 (skipped treatment). If using contingent treatment specifications that may or may not actually harvest trees in a given cycle, it is essential that they be specified for every cycle in which FVS **may** generate a cut list if the contingency is met.

Package ID: 010

Description: Thin from below to 50BA < 21DBH in cycle 00

FVS Cycle Length

5 Year Cycles 10 Year Cycles

Year	Treatment	Description
00	001	RX1: Thin from below to 50BA < 21DBH...
10	999	Grow Only
20	999	Grow Only
30	999	Grow Only

Simulation Year 00

Skip Treatment

Rx ID: 001

Description: RX1: Thin from below to 50BA < 21DBH in Cycle 1; prefer to keep fire resistant species

OK Cancel

Figure 4.7 – Package tab of the FVS: Treatment Package window. Define a package as a sequence of previously defined Rxs.

- c. **Harvest Costs:** This window contains a list of harvest cost components (not the actual per acre costs, just the names of the components and the cycles in which they apply), assigned during **Rx** creation, for each prescription in the package. These items are read-only since they are associated with the treatment, not the

package. Remember, the cost per acre values will be specified at a later stage in the analysis.

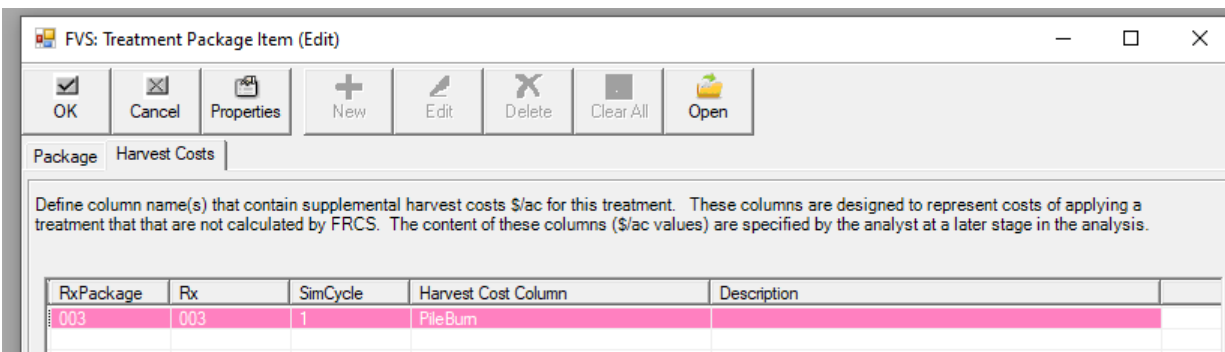


Figure 4.8 – Harvest Costs tab of the FVS: Treatment Package window. View a list of harvest cost components associated with Rx's in this package.

- d. Click **<OK>** to close the Treatment Package Item window. Note: the changes are not yet saved (see next step).
3. Repeat step 2 for each package you wish to create. When finished, click **<Save>** then **<Close>** to exit.
4. To make changes to a treatment package you have already created, simply select the package from the Treatment Package List table and click **<Edit>**. Select **<Delete>** to delete a package from your project. Remember that no changes, including deletions, are saved as final until clicking **<Save>** at the bottom of the Treatment Package List.
5. To display full documentation of prescriptions and packages in the project, (warning: this can take several minutes to complete) click **<Properties>** from the Treatment Packages window to open an html file in your default browser. The page will open in your browser and display details for every treatment assembled for this project database. Clicking the hyperlink **<Printer Friendly>** in the upper left hand corner will display a printable version of the summary table.
6. Click **<Close>** to exit the Treatment Packages window.

Tree Species

FVS predictions are customized to both geography and tree species, via allometric, regional and tree species specific equations for tree growth, mortality, and volume. These equations are

embedded in the FVS framework and result in a geographic variant of FVS. Therefore, each tree in a BioSum analysis must be assigned an FVS tree species code. The FIA database, however, has a greater number of tree species than are specifically simulated in FVS. The *tree_species* table in the project's *ref_master* database contains FVS species code assignments for many FVS variant and FIA species combinations. If you have a tree species in your data that is not recognized by the FVS variant used for prediction, the *tree_species* reference table may be customized to “map” that species to one that FVS does recognize for that variant. This step in the FVS module offers an opportunity to ensure all trees in your study area are assigned an appropriate FVS species code. Trees that have not been assigned a FVS species code valid for the variant in which they are modeled can lead to unpredictable results, including invalid predictions of growth and biomass and volume that “disappear” from the analysis.

1. Open the Tree Species window by selecting <Tree Species> from the side menu.
2. The **Tree Species** window will appear, with the *tree_species* reference table visible at the bottom. This table can be browsed, sorted, or filtered to, for example, find specific tree and variant combinations, or reference oven dry weight and dry to green values for specific tree species. To sort or filter, right click in a cell under the desired column type and select the desired type of filtering.
3. To ensure every FIA tree species and FVS variant combination in your project has an assigned FVS species code, select <Run Audit> from the Tree Species window.

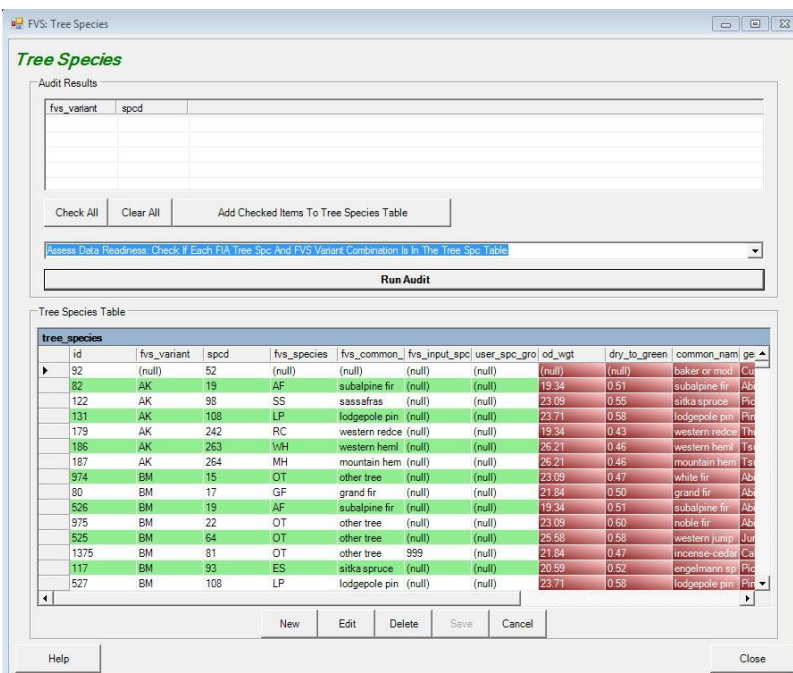


Figure 4.9 – Tree Species window. Run an audit to check for records without an FVS species code assignment.

- A list of species\variant combinations without an assigned FVS species code will appear in the **Audit Results** table in the upper half of the window, and a pop-up window will appear with audit details. If there are no missing values in your project database, a pop-up window will appear stating that the audit has passed. If your project contains tree species that do not have a corresponding FVS species code in the Tree Species table, an error message will be displayed to alert you that the audit failed (*Figure 4.10*).

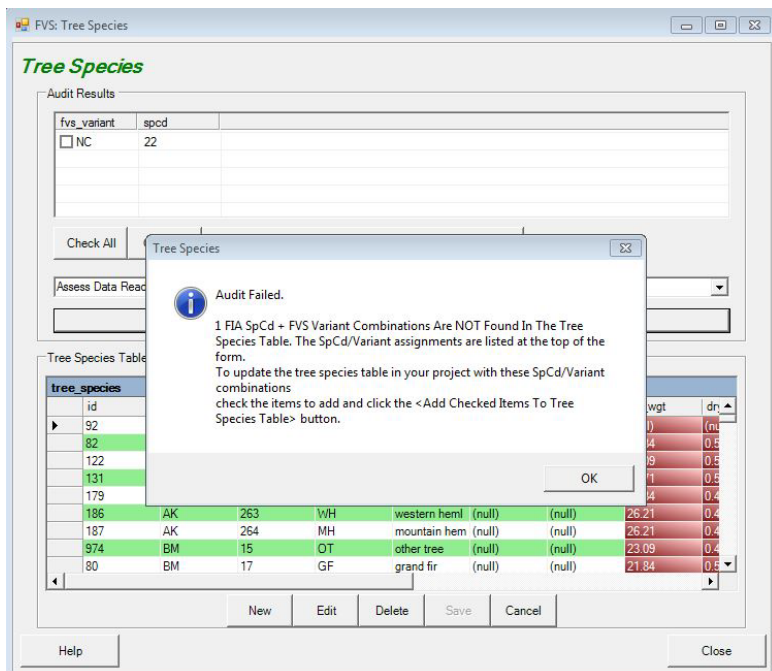


Figure 4.10 – Tree Species audit window. Audit will fail if a variant\species code combination is not found in the tree_species table.

- To add the records with null values (missing or invalid FIA/FVS species code combinations) to the tree species table, select the check boxes next to each record, or use the **<Check All>** button to add all records with null values, and then click **<Add Checked Items To Tree Species Table>**. To clear any selected records in the Audit Results table click **<Clear All>**.
- Once records are in the *tree_species* table, a record can be edited by clicking the **<Edit>** button to open the Processor Tree Species Edit window to edit the FVS variant, FIA tree species code, common name, oven dry weight, and other descriptors associated with the selected entry.
- Save edits to the *tree_species* table by clicking on the **<Save>** button. Changes to the table will not be updated until saved.
- If duplicate tree species\variant combinations are found within the table, a warning will appear when changes are saved (*Figure 4.11*). Close the warning window using the **<OK>** button, select one of the duplicate records in the table, and click **<Delete>**. Be sure to **<Save>** your update.

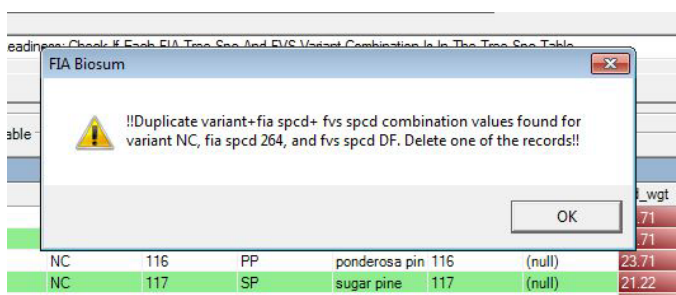


Figure 4.11 – If BioSum identified duplicate FIA species/FVS variant combinations, it will not allow the tree_species table to be saved. Duplicate records must be deleted before continuing.

FVS Input Data

The next task is to create the files necessary for FVS growth projection: tree and stand data that serve as simulation input. While there are several pre-processing tools that can be used to create the input files needed to run a simulation in FVS, it is essential that when using BioSum, these **files be created following BioSum's FVS Input Data workflow** to ensure linkage between FIA data and FVS simulation output.

As noted in Step 24 of the Database chapter of this user guide, all choices about what data will be available to be passed to FVS, including calibration, down wood, and seedling data must be made at the time plot data is imported into a BioSum project. Otherwise, these data cannot be included in the SQLite FVS input files (i.e., FVSin.db) created by BioSum.

To create FVS input files:

1. Open the **Create FVS Input** window by clicking **<FVS Input Data>** button from the side menu.
2. The top section of the interactive window summarizes the BioSum project up to this point, showing a list of variant/Rx package combinations currently loaded into the project. Two interactive hyperlink buttons (Treatments and Packages) can be used to jump back to previously completed steps, if needed. To revisit these steps, simply click the corresponding button.

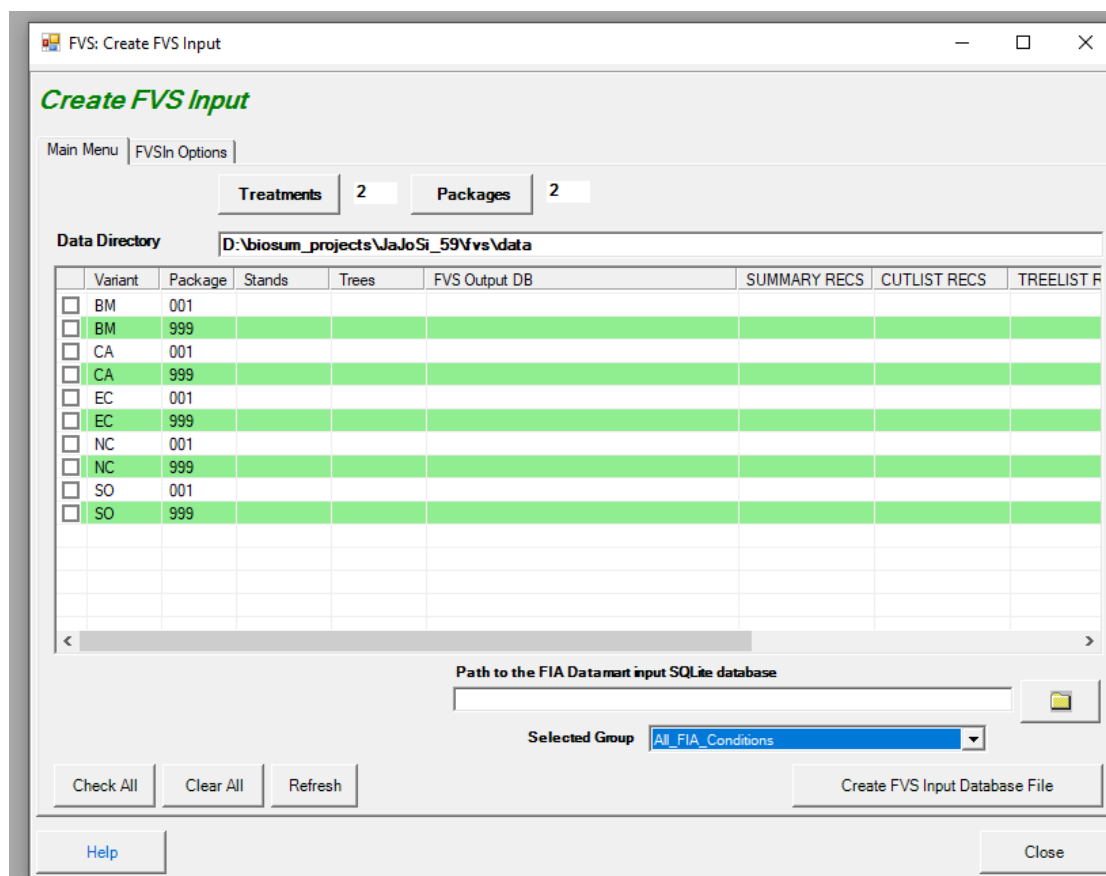


Figure 4.12 – Create FVS Input window.

3. **FVSIn Options tab:** The controls in this dialog affect what data is brought into the FVS input database files, which can significantly affect FVS simulation results. It is up to the analyst to understand how to tell FVS whether and how to use the information that ends up in this database. FVSIn options should be selected as needed at this point before proceeding further with the generation of FVSIn files.

For example, you can choose to rely on the surface fuel model recorded by FIA field crews (for the years it is available—2013 and later) or the calculated down wood biomass—both can be propagated to the FVSIn file, but the analyst will need to tell FVS which to use as the basis for representing surface fuels (relied on heavily to calculate the estimates in the POTFIRE and CARBON tables, for example).

Down wood data can be filtered by transect length—for example, if a forested condition occurs on only one subplot, and on only a part of that subplot, the coarse wood transect length could be as short as 5 feet or less—a VERY small sample on which to base surface fuel biomass estimates, so the analyst might choose to set a

higher threshold and pass coarse wood data to FVS only for conditions (stands) where the transect length on which down wood was sampled, exceeds that threshold. It's advisable to consult the field guides for each year of data you are interested in using, and/or the down wood tables, and/or an FIA analyst, for assistance in making these choices. The number, orientation and length of down wood transects has changed repeatedly over the past two decades.

Early in the annual inventory, confusion about duff and litter protocols (some due to the use of tenths of inches as a unit) leaves many analysts with little confidence that they are consistently and correctly recorded. These filtering checkboxes provide the option to load duff and litter data (for example to represent fuels or forest floor carbon) except for certain years (in which case FVS will assign what it “thinks” is a viable default for stand data collected in those years).

Checking the “Use GRM calibration data if available” propagates the GRM data loaded into the BioSum project into the FVSin database for use in FVS projections. This will populate tree-level height and diameter growth measurements (FVS_TREEINIT_DG; FVS_TREEINIT_HTG) into the FVSin database among trees that have been remeasured. When these values are populated in the FVSin database, FVS will use them to calibrate its species-level growth models towards values empirically observed within each stand. By default, at least 5 trees per size class and species, with remeasurement data available, are required to calibrate the small tree height or large tree diameter growth models. The CALBSTAT keyword can be used to change the minimum number of samples (trees with remeasurement data available) required to calibrate the growth models. Generally, including GRM data should increase the accuracy of FVS's growth projections, but it is up to the analyst to make this decision and alter any related calibration parameters as appropriate (e.g., via the CALBSTAT keyword).

Finally, checking the “Include seedlings” box will populate the FVS input database files with seedling records from the inventory data. Seedlings (and saplings) are sampled on a plot size that is less than a tenth the size used for trees over 5 inches in diameter. While this may be adequate for characterizing densities of the smallest trees over a large landscape, it is arguably less so for representing them at stand level – the most granular scale on which BioSum evaluates management outcomes. It is fine to include them, but important to realize that they can generate apparently anomalous results—for example, 7 seedlings observed on the 6.8 foot radius microplot on one subplot for a condition that occupies only a quarter of the full plot will translate to a total of 2100 trees per acre $(7 \times 75)/(0.25)$. If many of these are modeled by FVS to

survive and grow, this could translate to quite a high density of ladder fuels within a decade or two, so the analysis would depend very much on the accuracy with which FVS can do this. However, choosing not to include the seedling data may underestimate ladder fuel density (and the fire effects that would result) and also understate competition effects on growth, especially over longer simulations (e.g., >20 years).

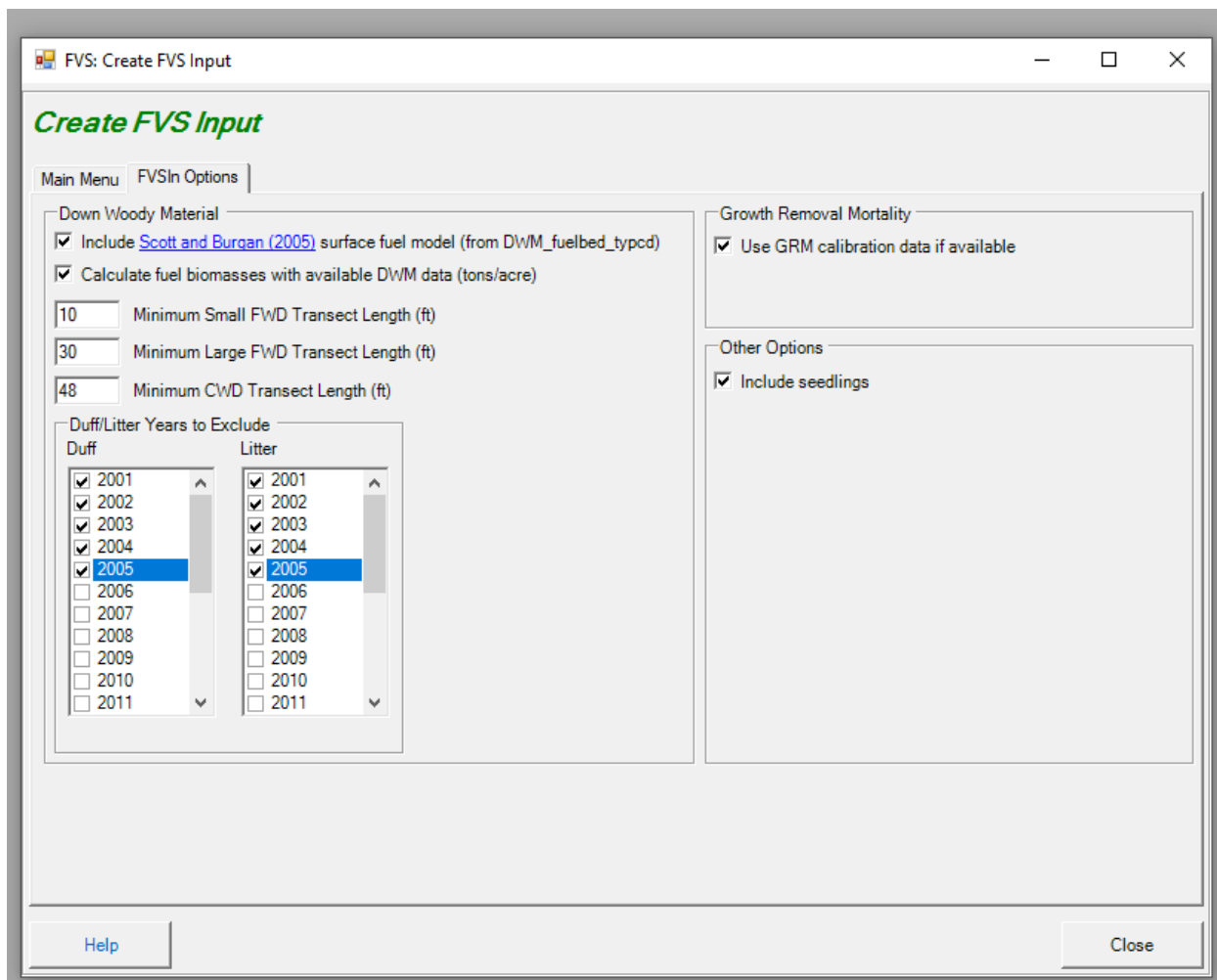
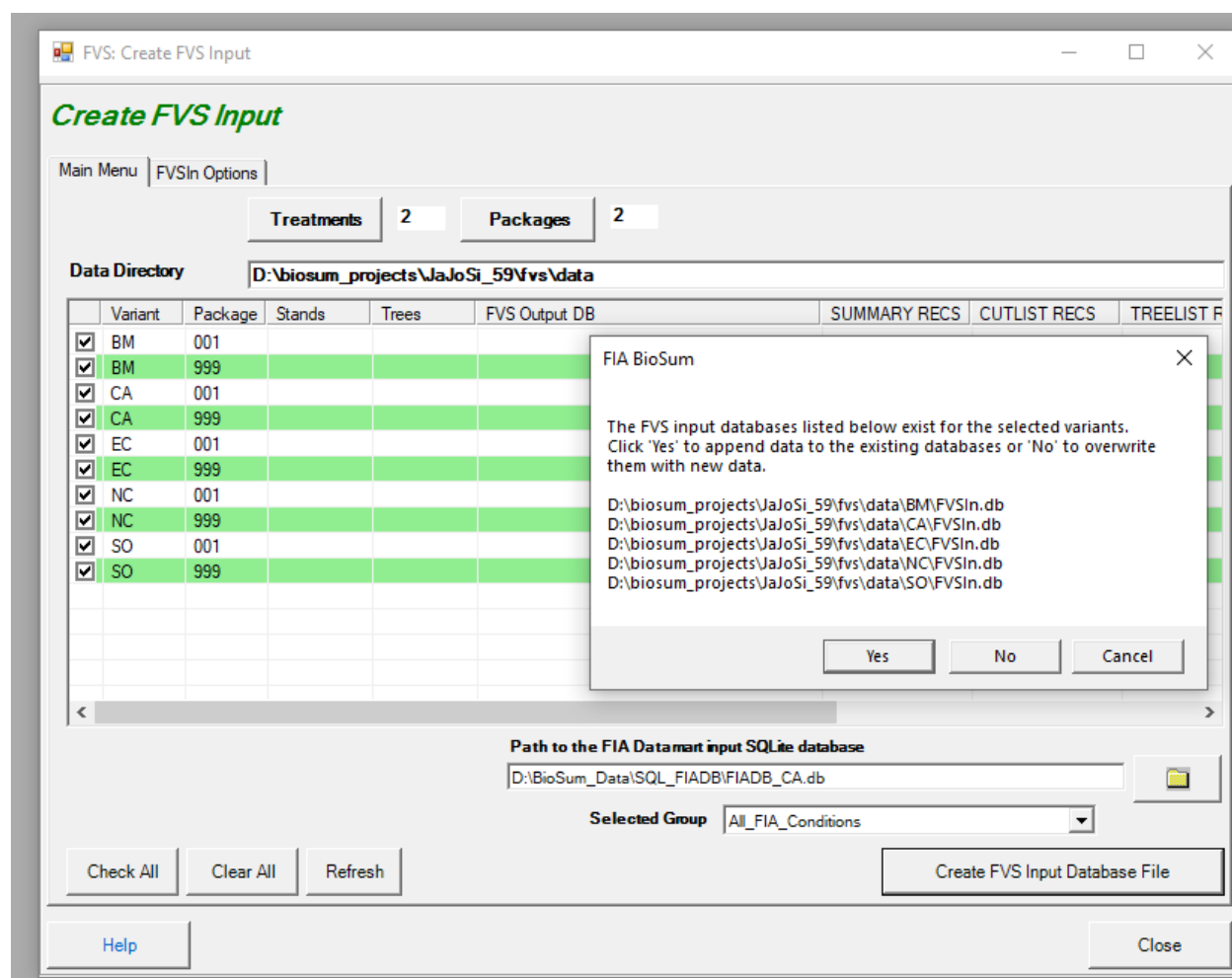


Figure 4.12.1 – The FVSIn options dialog for controlling how and whether field observations of down wood, surface fuel model, seedlings and measurements from previous visits (GRM data) are loaded to the FVSIn database for use by FVS.

4. Click on the Main Menu tab to return to the Rx package selections. Click **<Check All>** (in the lower left corner) to select all variant/Rx package combinations or select the desired packages by clicking on the check box next to each variant/package combination. Stands (FIA conditions) that do not belong to one of the selected combinations will not be included in the FVS input files that BioSum generates.

5. Click the “Selected Group” dropdown menu and select the ALL_FIA_CONDITIONS option.
6. Click on the file button under “Path to the FIA Datamart input SQLite database” and navigate to the location of the state specific FIADB SQLite (.DB) database that was used to load FIA plots to the project under the already-completed Database module workflow. Click the <Create FVS Input Database File> button to initiate the input database file creation process. When the process is completed, the columns next to each variant/Rx package will populate.
 - a. If FVS input databases already exist in a project, they can be either overwritten (replaced) or appended to. We recommend overwriting the files, unless the area of interest includes multiple states, in which case follow the directions below to append.
 - b. For a multi-state project, step 6 must be repeated for each state. Starting with the 2nd state, use the append option (see example of dialog box below for choosing between the append [yes] and overwrite [no] options). Appending a state to an FVSIn will increase the record counts for Stands and Trees; if these don’t change, that may be a sign of a problem.



FVS

The 2nd module in the BioSum workflow facilitates simulation of silvicultural sequences with the Forest Vegetation Simulator (FVS), accounting for growth and silvicultural management. These instructions assume that the user is running the FVS growth model through the FVSONline interface. All documentation and the most recent version of FVSONline can be found on the FVS website (<http://www.fs.fed.us/fmsc/fvs/software/complete.php>).

Some caveats demand consideration before beginning the FVS workflow. FVS is a complex and highly customizable application that will generate much more realistic predictions when operated by an experienced user. Model performance can be greatly improved by customizing settings and/or calibrating models to more closely reflect inventoried stand conditions, rather than relying on defaults. For more information on modifying model performance by establishing baseline trends, refer to Vandendriesche (2010).

In addition to calibration and setting customization, users may want to account for regeneration in FVS growth projections if it is important to obtain accurate estimates of, for example, ladder fuels and the sustainability of stand structure as stands respond to treatments that open the forest canopy. Currently, the only areas for which FVS provides a “full” establishment model are eastern Montana, central and northern Idaho, and coastal Alaska. The remaining states are equipped with only a “partial” establishment model, meaning that only regeneration from stump resprouting or analyst-defined tree establishment (natural or planted) is included in the growth projections. Vandendriesche (2009) developed a method for estimating and including natural tree establishment within FVS simulations, but that method relied on an exogenous program application that is now defunct. An alternative natural tree regeneration imputation procedure that does not rely on exogenous program(s) has recently been developed. See Appendix C for an overview of how to include regeneration in FVS growth projections using this method.

Strategies for Specifying FVS Cycles and Intervals

The analyst running BioSum is responsible for understanding how both FVS and BioSum “understand” simulation cycles. The BioSum side of this is fairly straightforward, in that the program supports 5- and 10-year cycles (which can also be thought of as projection periods, or intervals), though only one BioSum cycle length should be selected for a project. As described in the introduction to this module, for simplicity, BioSum’s developers have always instructed FVS, via the KCP file, to start all stands at a nominal year 1, regardless of the year in which the data was collected; any other choice is likely to result in profound complexity and analyst confusion. This is accomplished with the InvYear 1 code as follows (with the “1” in column 20):

```
!!*****
!!**Number Of Cycles and Years Between Cycles
!!*****
```

```
InvYear      1
```

We also recommend a “throwaway” year (a 1-yr projection without management) to allow for obtaining a pre-treatment value of stand state variables from FFE patterned tables such as POTFIRE, CARBON and FUELS. Following that 1-yr projection, from the BioSum perspective, choosing 5-year cycles provides an opportunity for treatment to occur (depending on how the KCP file is coded) at any or all of year 2, 7, 12 and 17; choosing 10-year cycles provides for treatment implementation at years 2, 12, 22 and 32. As currently programmed, BioSum generates a kcp template with these parameters:

```
InvYear      1
Timeint      1          1 # PRE for FFE; “throwaway” grow-only
```

```

Timeint          2          1 # treat, grow 1yr, PRE for non-FFE
Timeint          3          8 # POST for all tables, grow to next PRE
Timeint          4          1 # PRE for FFE tables
Timeint          5          1 # treat, grow 1yr, PRE for non-FFE
Timeint          6          8 # POST for all tables, grow to next PRE
Timeint          7          1
Timeint          8          1
Timeint          9          8
Timeint         10          1
Timeint         11          1
Timeint         12          8
Timeint         13          1 # Needed for FFE output at year 41
NUMCYCLE        13

```

A 5-year cycle length would require a different set of keyword code, with the same general principles and a 1 1 3 repeating pattern instead of a 1 1 8. This pattern has the greatest flexibility because it arranges for FVS output to include PRE and POST treatment stand state values (such as metrics of stand volume and density, canopy cover, crown fire potential, carbon pools, etc.) for each of 4 BioSum simulation cycles (not to be confused with FVS projection cycles—the above example includes 13 of those), each with the possibility of implementing management activity. If no management activity needs to be modeled beyond BioSum cycle 1, a simpler code is possible, for example:

```

InvYear          1
Timeint          1          1 # PRE for FFE; "throwaway" grow-only
Timeint          2          1 # treat, then grow 1yr to post
Timeint          3          8 # →10 Years post-treatment
Timeint          4         10 # →20 Years post-treatment
Timeint          5         10 # →30 Years post-treatment
Timeint          6         10 # →40 Years post-treatment
NUMCYCLE        6

```

Any simulation for which it is important to know the state of the stand both before and immediately after treatment and that includes plans for use of output from FFE tables in that representation requires a pattern that includes back-to-back 1-year projections. If instead of the 6 FVS cycle pattern above a pattern of 10 10 10 10 were used with management occurring in the first FVS projection cycle (i.e., at year 1), then the FFE tables would contain no output describing pre-treatment stand conditions and one would need to extract different sets of columns (from the Summary table) or rows (from the STRCLASS table) to extract both pre and post treatment stand state from the non-FFE tables, which is not how the BioSum workflow is structured. Using the standard BioSum workflow, only the pre-treatment state in non-FFE tables would be loaded for each cycle, so if management were implemented in cycle 1, you would not

see its effect on the state of the stand in those tables until 10 years of projected regrowth had occurred.

Ultimately, the choice of projection timing, made via edits to the KCP, is the responsibility of the analyst. The admittedly complex timing specifications are a result of a wrinkle in FFE's architecture that causes the reported values for any given year in those tables, including stalwarts typically used in BioSum analyses that track crown fire potential, fuels and carbon, are calculated **after** any harvest has occurred and **before** any surface fuel treatment (e.g., via pile and burn or prescribed fire) has taken place.

In earlier versions of BioSum, the challenges posed by FFE's odd architecture were addressed by running a separate, 1-yr "base year" projection to generate a POTFIRE table without management and that output was then prepended to the POTFIRE table output from the main simulations as year 1, with the year attribute incremented by 1 for all rows in those tables and FVS output loading was modified accordingly. Though effective, this workflow was confusing and difficult to explain, and was only implemented for POTFIRE (not CARBON, FUELS or any of the other FFE tables) so we no longer recommend it. Instead, we recommend the throwaway year (1 year grow-only projection before commencing management simulation), which is effective for all FFE tables. The legacy BASE YEAR approach remains temporarily as a legacy option that is lightly documented in the appendix for those with legacy BioSum projects on which BASE YEAR was implemented.

Simulating silvicultural sequences (Rx Packages) in FVSONline

After FVS variants have been assigned, prescriptions and sequences defined, tree species mapping confirmed, FVS Input Data created, FVS calibration/regeneration considered, and POTFIRE base year values generated (if applicable), it is finally time to run the FVS growth projections for each silvicultural sequence and each variant in a project. Each combination of silvicultural sequence (aka Rx package) and variant, is referred to as a "run" in FVS terminology. This section describes major steps and best-practices associated with FVSONline that are relevant to BioSum users. It is not an exhaustive guide to the FVSONline interface for parameterizing and producing FVS outputs.

Loading Data

Once an FVSONline project has been created, use the heading tabs to navigate to Manage Project < Import input data < Upload inventory database. Click the browse button to navigate to the location of your BioSum generated FVS input file (project/fvs/variant/FVSin.db) then click the "Install uploaded database" button. If your project includes multiple variants, additional variant specific FVSin files can be loaded into the project one at a time. To do so, navigate to the location of the next FVSin file using the browse button, and this time click the "Add new database to existing database" button instead. Continue this process for any additional variant

specific FVSin files. You can always check what data is loaded into FVS at any time by consulting the View and edit existing tables tab.

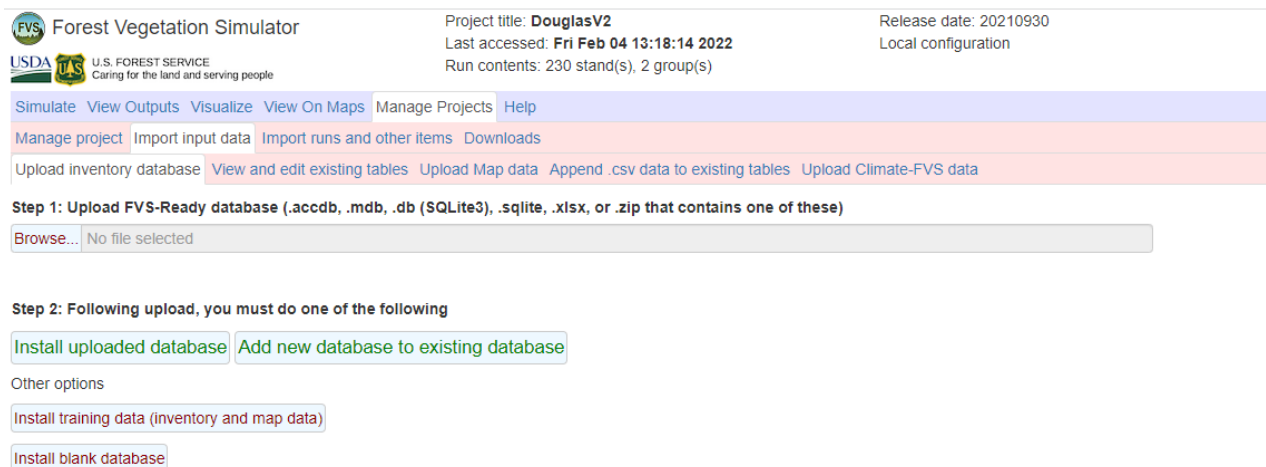


Figure 4.22 - The FVSONline interface used to load (import) FVS input databases generated from BioSum

Crafting and Simulating FVS Runs (Variant-Specific Rx Packages)

To generate FVS outputs that can be loaded into and used by a BioSum project, FVS runs must be crafted and simulated for each variant/Rx package combination, to match the Rx packages defined in your BioSum project. This process is completed in the “Simulate” tab of FVSONline.

Forest Vegetation Simulator
 U.S. FOREST SERVICE
 Caring for the land and serving people

Project title: **DouglasV2**
 Last accessed: **Fri Feb 04 14:11:54 2022**
 Run contents: 129 stand(s), 2 group(s)

Release date: 20210930
 Local configuration

Simulate View Outputs Visualize View On Maps Manage Projects Help

Selected run
 FVSOUT_CA_P001-999-999-999-999
 New Reload Save Duplicate Delete

Run title
 FVSOUT_CA_P001-999-999-999-999

Run Contents Full run Just groups

00412010060102019513461
 > Grp: All_FIA_Conditions
 -> Kwd: From: FVS_GroupAddFilesAndKeyv
 -> Kwd: Editor: Grow Only
 00412010060102019513463
 > Grp: All_FIA_Conditions
 -> Kwd: From: FVS_GroupAddFilesAndKeyv
 -> Kwd: Editor: Grow Only
 00412010060102019515461
 > Grp: All_FIA_Conditions
 -> Kwd: From: FVS_GroupAddFilesAndKeyv
 -> Kwd: Editor: Grow Only
 00412010060102019515462
 > Grp: All_FIA_Conditions
 -> Kwd: From: FVS_GroupAddFilesAndKeyv
 -> Kwd: Editor: Grow Only
 00412010060102019542501
 > Grp: All_FIA_Conditions
 -> Kwd: From: FVS_GroupAddFilesAndKeyv
 -> Kwd: Editor: Grow Only
 00412010060102019542502
 > Grp: All_FIA_Conditions
 -> Kwd: From: FVS_GroupAddFilesAndKeyv

Edit Change to freeform Cut/Delete Copy

Paste item selected below

Components available to paste
 Editor: Grow Only

Find stand:
 Find

Stands Time Components Select Outputs *Run*

User data installed Quick link to: Upload inventory database

Inventory Data Tables
 FVS_StandInit_Cond

Variants
 ca: Inland CA, Southern Cascades

Groups
 All_FIA_Conditions

Stands must be in any or all selected groups Any All

00412010060102019513461
 00412010060102019513463
 00412010060102019515461
 00412010060102019515462
 00412010060102019542501
 00412010060102019542502
 00412010060102019551851
 00412010060102019619111
 00412010060102019631671
 00412010060102019771481
 00412010060102019894801
 00412010060102019906411
 00412010060102019979981
 00412011060202019565071
 00412011060202019595011
 00412011060202019595012
 00412011060202019640031
 00412011060202019765564

129 Stand(s) in 1 Group(s)

Number of replicates of each added stand 1

Relative weights of each replicate 1

Add selected stands Add stands in selected groups

Find stand(s):
 Find

Figure 4.23 - The Simulate tab of FVSOnline, where runs (i.e., the variant/Rx package combinations defined in BioSum) are crafted and simulated.

Naming Runs

How runs are named in FVS is extremely important for BioSum post-processing and must follow a specific naming convention, else you will incur errors when attempting to load FVS output data back into BioSum. The naming convention is as follows:

FVSOUT_(variant)_P(Rx package #)-(cycle 1 treatment #)-(cycle 2 treatment #)-(cycle 3 treatment #)-(cycle 4 treatment #)

A valid naming example would look like: FVSOUT_CA_P001-110-999-999-999

Where CA refers to the CA variant, P001 refers to Rx package 001, 110 refers to a thin-from below treatment in cycle 1, and 999 refers to do nothing (grow-only) for the remaining cycles 2, 3, and 4.

Type the run title name under “Run title” and click the “Save” button. A new run has been added to your FVS project and is ready to have input data appended to it. To add stands to a run, select the desired variant from the “Variant” drop-down menu, ensuring “All_FIA_Conditions” from the Groups box is highlighted. All stands (conditions) associated with the selected variant will be populated into the Stands table. Highlight specific stands you want to import to the run and click the “Add selected stands” button. The specified stands will be added to the Run Contents table. You are now ready to craft and append FVS keywords to stands that simulate the Rx treatments associated with your FVS run. You will also need to append a basic set of BioSum-related keywords to maintain compatibility of the FVS outputs with BioSum.

Adding Keywords to Runs

FVS keywords can be added to stands using two methods, (1) via GUI-guided command tables in the Components < Management, Modifiers, Event Monitor, Economic, and Keywords tabs or (2) via the Components < Editor tab, which allows for free-form FVS keyword scripting. Before applying Rx-related keyword to scripts, ensure BioSum-related keywords are first appended via the Keyword Editor. Click on the Components < Editor tab and click the <Browse> button under the “Upload an existing Keyword component file (KCP), or Keyword component archive (FVS_kcps.Rdata) header. Navigate to and select the BioSum Keywords KCP file saved in your project directory: project/fvs/data/BioSum_Keywords.kcp, then give the Component Title a name (e.g., BioSum_Keywords). The keyword component can be saved for later (useful when appending the same keyword script to many different runs) by clicking the <Save in component collection> button. The component can then be copied to all stands, by first ensuring all stands are highlighted from the Run Contents window, then clicking the <Save in run> button.

Forest Vegetation Simulator
 Project title: **Douglas_Rx_Test** Release date: 20230106
 Last accessed: **Thu Mar 23 12:31:01 2023** Local configuration
 Run contents: 129 stand(s), 2 group(s)

Simulate View Outputs Visualize View On Maps Manage Projects Help

Selected run
 FVSOUT_CA_P003-003-003-003
 New Reload Save Duplicate Delete

Run title
 FVSOUT_CA_P003-003-003-003

Run Contents Full run Just groups

1201941010201900669520001
 > Grp: All_FIA_Conditions
 -> Kwd: From: FVS_GroupAddFilesAndKeyv
 -> Kwd: Editor: BioSum_Keywords.kcp
 1201841010101900964790001
 > Grp: All_FIA_Conditions
 -> Kwd: From: FVS_GroupAddFilesAndKeyv
 -> Kwd: Editor: BioSum_Keywords.kcp
 1201941010201900946500001
 > Grp: All_FIA_Conditions
 -> Kwd: From: FVS_GroupAddFilesAndKeyv
 -> Kwd: Editor: BioSum_Keywords.kcp
 1201941010201900766090001
 > Grp: All_FIA_Conditions
 -> Kwd: From: FVS_GroupAddFilesAndKeyv
 -> Kwd: Editor: BioSum_Keywords.kcp
 1201841010101900689310001
 > Grp: All_FIA_Conditions
 -> Kwd: From: FVS_GroupAddFilesAndKeyv

Edit Change to freeform Cut/Delete Copy

Paste item selected below
 Components available to paste
 Editor: F-KLSC-EA1-003.kcp

Find stand: Find

Existing component collection
 BioSum_Keywords.kcp
 New Append selected component from run Save in run Save in component collection Delete Download (KCP)

Component Title: BioSum_Keywords.kcp
1.....2.....3.....4.....5.....6.....7.....8

```

  !!*****
  !!**Code generated by FIA Biosum
  !!*****
  !!**TIME: Number Of Cycles and Years Between Cycles
  !!*****
  !!The default TIME-related keywords below are parameterized
  !!for 10-year cycles and to accommodate the inclusion of
  !!pre-treatment values in Cycle 1 for FFE tables (e.g., POTFIRE) that
  !!report only after-treatment stand attributes. See the
  !!BioSum User Guide FVS chapter for additional guidance on
  !!alternative TIME keyword configurations.
  InvYear      1
  Timeint     1      1
  Timeint     2      1
  Timeint     3      8
  
```

Math:
Variables:
Species codes:
FVS Functions:

Description: This Editor menu allows you to utilize the advanced features of the freeform text format for creating custom component sets by directly adding & editing keyword records and Event Monitor functions. You can upload an existing keyword component file (.kcp), or keyword component archive (FVS_kcps.Rdata) and then save it into the Run Contents window on the left (Save in run), and also save it in the component collection (Save in component collection). You can also create your own component sets by appending items from the Run Contents on the left (Append selected component from run) and then saving them into your component collection (Save in component collection). Finally, you

Figure 4.24 - Using the keyword editor to copy BioSum keywords to all stands for a single run.

IMPORTANT: Specifying time correctly via select keywords is extremely important for BioSum post-processing, yet how it is specified will vary depending on the project, Rx packages, and whether POTFIRE tables are required. Consult the **Strategies for Specifying FVS Cycles and Intervals** section above. Those requiring documentation of legacy BASE YEAR implementation are advised to consult **BioSum User Guide Appendix D** and edit the “TIME: Number Of Cycles and Years Between Cycles” keywords as necessary. By default, the time keywords in the Biosum_Keywords.kcp file specify a 40-year simulation (four 10-year BioSum cycles). In this case, treatments should occur specifically in FVS cycles 2, 5, 8, 11 (simulation years 2, 12, 22, 32; BioSum cycles 1, 2, 3, 4).

Full documentation regarding FVS keywords and parameterization can be referenced in the Keyword Reference Guide for the Forest Vegetation Simulator (Van Dyck and Smith-Mateja 2000). FFE related keywords can be referenced in the Fire and Fuels Extension to the Forest Vegetation Simulator (Rebain 2010)

Specifying Output Tables

For each run, you will need to specify what information (tables) are output by FVS. This can be done by editing Output Table Keywords in the BioSum-Keywords.kcp. BioSum requires the FVS_Cutlist and FVS_Summary tables – these tables are produced by default. Consult FVS documentation on whether other output tables are useful for your analysis and edit the BioSum_Keywords.kcp file “Output Table Keywords” section appropriately. FVS output table keywords can be referenced in the User’s Guide to the Database Extension of the Forest Vegetation Simulator (Crookston 2003). Keep in mind that as more tables are selected for export, the larger the final FVS output file will be. This can drastically increase post-processing time spent in BioSum and occasionally lead to system memory issues when Access files become too large (e.g., >2gb). Thus, it is good practice to determine what tables (information) are needed for a BioSum analysis beforehand to prevent the export of unneeded tables and data.

Set additional harvest cost flags in FVS via the FVS_COMPUTE table (optional)

As explained in Step 2d of the treatment definition workflow and in the section on Additional CPA at the beginning of this chapter, when coding silvicultural treatments (Rx) in FVS via .kcp keyword files, any treatment components that incur operational costs that are not accounted for by OPCOST will need flags set (one per activity) in the FVS_COMPUTE table during the FVS simulation for BioSum to correctly track those costs. The FVS_COMPUTE table contains all user-defined and calculated variables. Examples of additional harvest costs that may need to be tracked include those associated with surface fuel treatments (such as mastication, prescribed fire or pile burning); mechanical site prep; application of herbicide, fertilizer, lime or biochar; water-barring skid trails; and planting or seeding. Recall that if additional CPA flags were specified with the treatment when it was defined, the activity flags were associated with the treatment as 8-character labels (that avoided spaces and special characters for compatibility with FVS conventions for variable names in COMPUTE).

These flags need to be set to 1 in the FVS_COMPUTE table for the record corresponding to the cycle/year in which an activity occurs, and to 0 otherwise (if the cost should not be incurred at that cycle). This is best accomplished via keyword statements integrated into the treatment keyword (kcp) scripts and/or as individual keyword addfiles appended to the simulation. At its most basic, such kcp code looks like:

```
COMPUTE          0
VARNAME = 1
END
```

Such flag-setting must be applied for each treatment component that incurs additional costs per acre (in essence, all of the additional CPA components defined for each Rx that is a component of the silvicultural sequence [package] represented by the kcp code being modified).

Below is an example where the component flag (broadcast burn; BRDBURN) keywords have been integrated into a treatment keyword (kcp) script. BRDBURN values are only set to 1 when the treatment is conditionally executed.

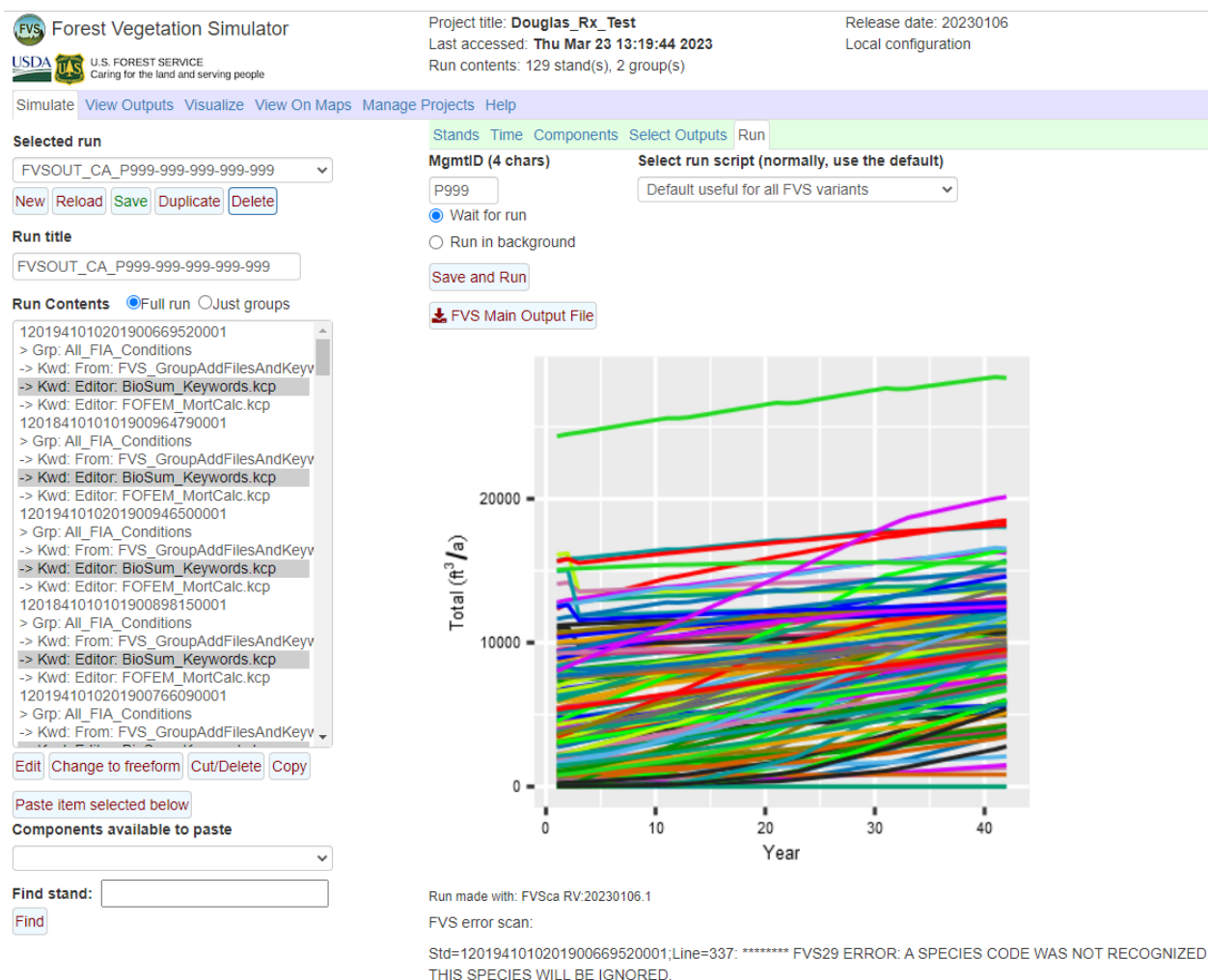
```
*Conditional statement triggers treatment if FVS cycle = 2
IF
CYCLE EQ 2
THEN
*The thinning (harvest) component of the treatment
YardLoss          0      PARMS(.05, 1, 0.1)
ThinSDI           0      PARMS(200, 1, All, 7, 30, 1)
*The broadcast burn component of the treatment
FMIN
SimFire           0      PARMS(3, 2, 55, 1, 100, 4)
FlameAdj          0      PARMS(1.0, 0.2)
FMortMlt          0      PARMS(0, All, 10, 999)
END
*Sets the BRDBURN variable value to 1 in the FVS_COMPUTE
table when treatment is applied
COMPUTE           0
BRDBURN = 1
END
ENDIF
```

Important Timing Note: The FVS_COMPUTE flag associated with a treatment component that incurs additional harvest costs must be set only for FVS cycles that correspond to BioSum cycles 1-4 in which management activity is scheduled to occur. These are the FVS cycles for which output is loaded into the pre-treatment registers of the Summary PREPOST table during the FVS Output Load using sequence number assignment). For example, if FVS cycles 2, 5, 8, and 11 are selected as the pre-treatment sequence numbers for the Summary output for BioSum cycles 1, 2, 3, and 4, the flag for broadcast burn must be activated in FVS cycles 2, 5, 8, or 11 (if, in fact, the additional CPA should be incurred for the associated BioSum cycles). The broadcast burn treatment itself can be programmed to occur in a different FVS cycle, however (e.g., the year/cycle immediately after tree harvest, or even later—what’s essential is that the flag be set to 1 for a pre-treatment cycle).

Executing Run Simulations

Once a run is correctly named, has stands appended to it, and has Rx-related and BioSum related keywords appended to stands, the run is ready for simulation execution. Navigate to the *Run* tab. Ensure you save run parameters by clicking the Save button under the Selected Run

dropdown menu. Optionally, give the run a Management ID (MgmtID) if desired. Runs can either be executed one at a time or batch processed. To run one at a time, ensure the “Wait for run” radio button is selected and click the Save and Run button. When this option is checked, the run will take longer to process, but will provide a list of potential FVS errors that occurred. This is useful for diagnosing whether FVS correctly read all input keywords and/or database values. To batch process, ensure the “Run in background” radio button is selected, select how many CPU cores should be used while processing (i.e., parallel processing) and click the Save and Run button. This option can significantly reduce processing time by utilizing parallel processing, assuming the user’s computer has many CPU cores. With this option selected, potential FVS errors will not be displayed, thus it is pragmatic to first use the “Wait for run” option to initially diagnose errors, then the “Run in background” option to maximize processing speed.



Forest Vegetation Simulator
 Project title: Douglas_Rx_Test
 Release date: 20230106
 Last accessed: Thu Mar 23 13:19:44 2023
 Local configuration
 Run contents: 129 stand(s), 2 group(s)

Simulate View Outputs Visualize View On Maps Manage Projects Help

Selected run
 FVSOUT_CA_P999-999-999-999-999
 New Reload Save Duplicate Delete

Run title
 FVSOUT_CA_P999-999-999-999-999

Run Contents Full run Just groups

1201941010201900669520001
 > Grp: All_FIA_Conditions
 -> Kwd: From: FVS_GroupAddFilesAndKeyv
 -> Kwd: Editor: BioSum_Keywords.kcp
 -> Kwd: Editor: FOFEM_MortCalc.kcp
 1201841010101900964790001
 > Grp: All_FIA_Conditions
 -> Kwd: From: FVS_GroupAddFilesAndKeyv
 -> Kwd: Editor: BioSum_Keywords.kcp
 -> Kwd: Editor: FOFEM_MortCalc.kcp
 1201941010201900946500001
 > Grp: All_FIA_Conditions
 -> Kwd: From: FVS_GroupAddFilesAndKeyv
 -> Kwd: Editor: BioSum_Keywords.kcp
 -> Kwd: Editor: FOFEM_MortCalc.kcp
 1201841010101900898150001
 > Grp: All_FIA_Conditions
 -> Kwd: From: FVS_GroupAddFilesAndKeyv
 -> Kwd: Editor: BioSum_Keywords.kcp
 -> Kwd: Editor: FOFEM_MortCalc.kcp
 1201941010201900766090001
 > Grp: All_FIA_Conditions
 -> Kwd: From: FVS_GroupAddFilesAndKeyv

Edit Change to freeform Cut/Delete Copy

Paste item selected below
 Components available to paste

Find stand:
 Find

Run made with: FVScal RV.20230106.1
 FVS error scan:
 Std=1201941010201900669520001;Line=337: ***** FVS29 ERROR: A SPECIES CODE WAS NOT RECOGNIZED THIS SPECIES WILL BE IGNORED.

Figure 4.26 – Executing a grow-only Rx package for the CA variant from the *Run* tab of the FVSONline interface. In this case, a keyword error (species code not recognized) was detected and logged.

Exporting Output Data

Once all FVS runs (Rx package and variant combinations; POTFIRE base year and variant combinations) in a project have been simulated, the user is ready to export the FVS output data that will be loaded into BioSum. Navigate to the Manage Projects < Downloads tab and click the “Output data base (.db, all runs)” button. Navigate to the project/fvs/data folder of your BioSum project. The output file must be named “FVSOOut” and be saved in the project/fvs/data folder. Click save.

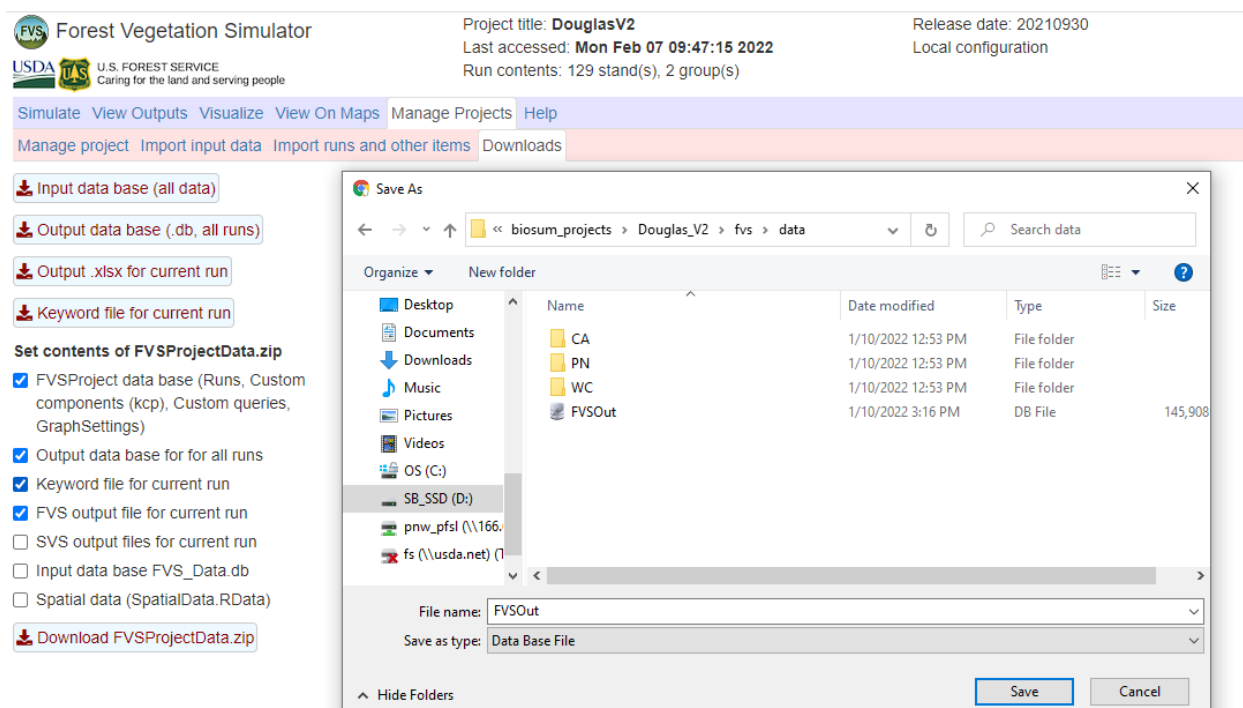


Fig 4.27 - Location of the “Output data base (.db, all runs)” button used to export the FVS output file and location where the FVSOOut.db file must be saved within the BioSum project directory.

FVS Output Data

The last task in the BioSum’s FVS module reformat FVS output as input to the BioSum Processor and Treatment Optimizer modules. The “cut list” data (list of trees harvested in management operations) are used to calculate potentially available biomass and merchantable wood volume and value, and costs of implementing treatment packages; the other FVS outputs provide pre- and post-treatment stand attributes that can be a basis for evaluating management effectiveness. With the project open in BioSum Manager:

1. Click <FVS> on the left side-bar, then <Create FVSOUT MDBs >.
2. From the newly opened FVS: Create MDBs window, click the Create MDBs button. This process will write data from the single FVSOut.db file produced in the previous section into many .mdb (Access) databases. A .mdb database will be written and populated with data from each run executed in FVSONline. Databases are stored in the project/fvs/data/variant subfolders. If this process has already been run previously in a project, databases will be overwritten rather than appended, preventing duplication of outputs. The process may take 1-30 minutes (or longer) to complete, depending on the size of a project (i.e., number of Rx packages, variants, and stands). The dialog box will actively track execution of SQL statements used to write databases; you may close the window when the dialog box displays “Done”.

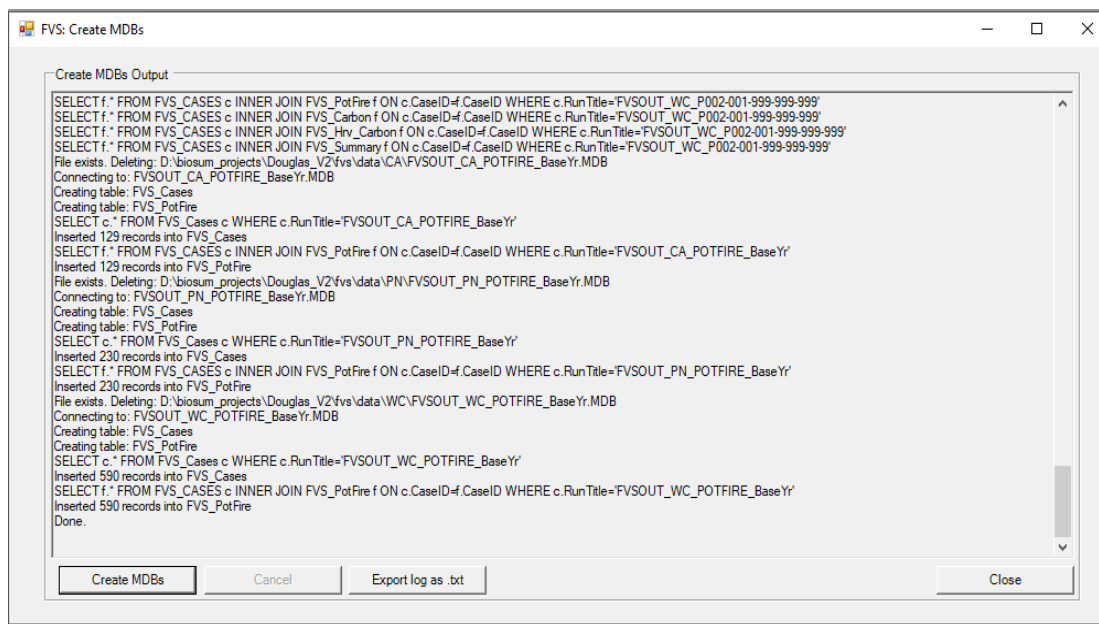
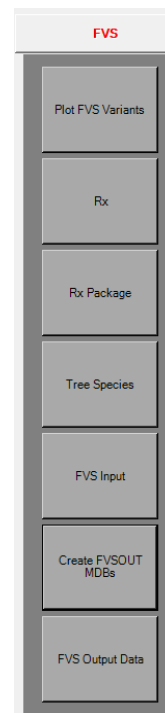


Figure 4.28 - The Create FVSOUT MDBs window in the BioSum interface with completed processing shown in the dialog box.

3. Next, click the **<FVS Output Data >** button from the left side of the BioSum interface; The Join and Append FVS Out Data window opens showing a list of all variant/Rx package combinations that have been modeled in FVS, similar to the window displayed when creating FVS input files. Additional information is also provided (viewable by scrolling to the right), such as the output database file name, Rx package specifics, whether the file was successfully found, and the number of records in the FVS summary, cut tree, standing tree, and potential fire tables.

There are 4 required and 2 optional tasks managed here, with the required tasks to be executed in sequence. Only Steps 2-4 and the pre- and post-append and audit tables and logs buttons respect the status of the checkboxes adjoining FVS variant and package identifiers—the other selections in the drop down “step list” apply to all variants and packages.

Defining the sequence number assignments (Step 1) builds the configuration tables (in `[project_name]\db\fvsmaster.mdb`) that control FVS output loading. See “Strategies for Specifying FVS Cycles and Intervals” earlier in this chapter for details on how to assign sequence numbers. Variant-package combinations showing an “a” next to the checkbox have not yet been appended or the source data in what was exported for that combination from FVSOOut contains a more recent time stamp than what is currently loaded in the BioSum data structures.

Manually select output databases to be appended or click **<Check All>** to select all variant/Rx package combinations, then click **Execute Step** (when Step 1 is highlighted).

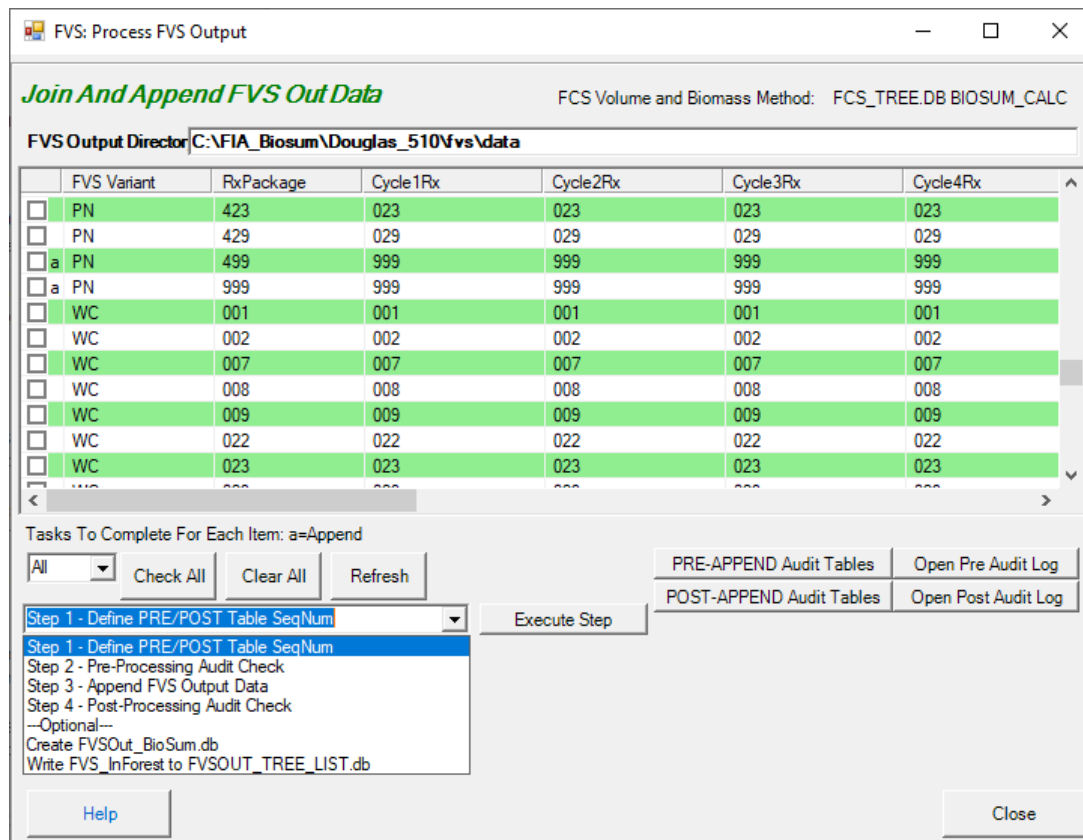
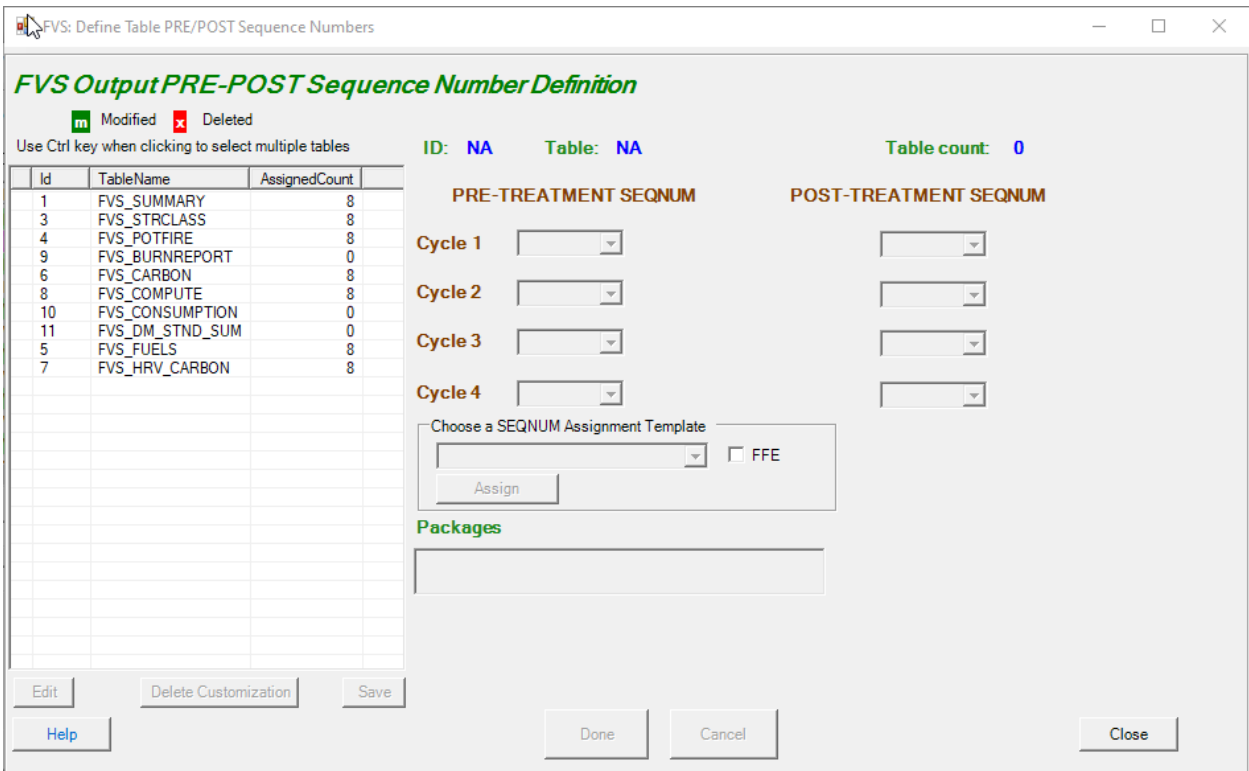


Figure 4.29 – Join and Append FVS Out Data window. Displays a list of all variant and package combinations.

The table grid displays names of all tables found in the FVSOut.db in the project_name\fs\data folder and the number of sequence number assignments (AssignedCount) currently specified for each table. Those with zero will not be brought forward into PrePost databases.



The fact that sequence numbers are assigned (i.e., a non-zero count) does not necessarily make them the right sequence numbers and at some point before proceeding to load data, every table should be checked. To view the current assignments, select a table name and press “edit”. The assignments populate and can be edited, if needed. In this assignment for Summary:

FVS: Define Table PRE/POST Sequence Numbers

FVS Output PRE-POST Sequence Number Definition

Modified Deleted

Use Ctrl key when clicking to select multiple tables

ID: 1 Table: FVS_SUMMARY Table count: 1

Id	TableName	AssignedCount
1	FVS_SUMMARY	8
3	FVS_STRCLASS	8
4	FVS_POTFIRE	8
9	FVS_BURNREPORT	0
6	FVS_CARBON	8
8	FVS_COMPUTE	8
10	FVS_CONSUMPTION	0
11	FVS_DM_STND_SUM	0
5	FVS_FUELS	8
7	FVS_HRV_CARBON	8

PRE-TREATMENT SEQNUM POST-TREATMENT SEQNUM

Cycle 1 2 3

Cycle 2 5 6

Cycle 3 8 9

Cycle 4 11 13

Choose a SEQNUM Assignment Template

Assign FFE

Packages

001,002,003,004,007,008,009,010,011,020,022,023,025,026,029,030,101,102,103,104,107,108,109,110,111,122,123,125,126,

Edit Delete Customization Save

Help Done Cancel Close

The cycle 1 Pre(treatment) value of 2 and the cycle 1 post of 3 tell us that for each stand, for each package, BioSum will take the 2nd output row in the summary table as the Pre value and the 3rd as the Post value, where table output rows are sorted by year within stand and package. This will be the case whether YEAR holds values like 2020, 2001, 2022, 2030, 2031, 2032, etc. as they would if the plot was sampled in 2020 and if InvYear 1 had not been specified in the FVS simulation or values like 1, 2, 3, 11, 12, 13, etc. if InvYear was set to 1 (the recommend option!). This assignment reflects the recommended practice of modeling a grow-only year before the first treatment can occur, via a 1 1 8 1 1 8 1 1 8 etc. pattern of projections which enables the specification of a true pre and post treatment year for each cycle for both FFE timed tables and non-FFE ones (Summary is an example of the latter).

These assignments can be made manually, but it is easier and more consistent to use the templates available in the template drop down.

Choose a SEQNUM Assignment Template

Mgt4X_PrePost
Mgt1X
Mgt4X_WtdMean

ackages

The assignments displayed are patterned on the Mgt4X_WtdMean template (management at up to 4 cycles with a weighted mean style analysis which compares a treatment trajectory to a base trajectory such as grow-only). Selecting that template and choosing Assign will set the exact same assignments seen in this figure. The Max4_PrePost is a very similar template differing only in what is assigned for Cycle 4 Post—Sequence number 12, which pulls the record for immediately following treatment in Cycle 4 rather than a later value that allows for a weighted mean that extends further into the 4th decade. The Mgt1X template is designed for a BioSum simulation which treats only at cycle 1, then follows the outcome for 3 more cycles. The same assignments make sense in that case whether doing a pre-post or a weighted mean style analysis.

To end assignment for a table, click Done. If changes were made, an “m” for modified will appear on that table’s line in the table grid. The choices made will not take effect in the FVSOUT configuration until a Save is executed (with a button click). Closing without saving abandons the changes.

STRCLASS, another non-FFE table, has essentially the same assignments as SUMMARY except that one has the option of selecting the before or after tree removal cases (there are two lines in that output table for every simulated year). In almost all cases, you will want the before tree removal cases.

FVS: Define Table PRE/POST Sequence Numbers

FVS Output PRE-POST Sequence Number Definition

Modified Deleted
Use Ctrl key when clicking to select multiple tables

ID: 3 Table: FVS_STRCLASS Table count: 1

Id	TableName	AssignedCount
m 1	FVS_SUMMARY	8
3	FVS_STRCLASS	8
4	FVS_POTFIRE	8
9	FVS_BURNREPORT	0
6	FVS_CARBON	8
8	FVS_COMPUTE	8
10	FVS_CONSUMPTION	0
11	FVS_DM_STND_SUM	0
5	FVS_FUELS	8
7	FVS_HRV_CARBON	8

PRE-TREATMENT SEQNUM

Cycle 1: 2 | 0 = Before Tree Removal

Cycle 2: 5 | 0 = Before Tree Removal

Cycle 3: 8 | 0 = Before Tree Removal

Cycle 4: 11 | 0 = Before Tree Removal

POST-TREATMENT SEQNUM

3 | 0 = Before Tree Removal

6 | 0 = Before Tree Removal

9 | 0 = Before Tree Removal

13 | 0 = Before Tree Removal

Choose a SEQNUM Assignment Template

Mgt4X_WtdMean FFE

Assign

Packages

001,002,003,004,007,008,009,010,011,020,022,023,025,026,029,030,101,102,103,104,107,108,109,110,111,122,123,125,126.

Edit Delete Customization Save

Help Done Cancel Close

COMPUTE is a non-FFE table that must match the sequence numbers entered for the SUMMARY Table. If the sequence numbers between the two tables do not match, cost accounting for packages and treatments that include supplemental harvest costs (Additional CPA) will not work properly. Ensure that the FFE box is not checked when setting the COMPUTE table's sequence numbers.

POTFIRE is an FFE table and has subtle differences in the assignments relative to the non-FFE tables owing to it displaying only post-harvest values in each record. FFE style tables need to have the FFE box checked in order for the Assign button to make the right assignments (e.g., for the CARBON and FUELS tables). BioSum knows that POTFIRE is an FFE table so will only assign it an FFE sequence (that relies on grow only years to have the pre-treatment values at each cycle) regardless of whether or not you check the FFE box for this table.

FVS Output PRE-POST Sequence Number Definition

Modified Deleted
Use Ctrl key when clicking to select multiple tables

Id	TableName	AssignedCount
m 1	FVS_SUMMARY	8
m 3	FVS_STRCLASS	8
4	FVS_POTFIRE	8
9	FVS_BURNREPORT	0
6	FVS_CARBON	8
8	FVS_COMPUTE	8
10	FVS_CONSUMPTION	0
11	FVS_DM_STND_SUM	0
5	FVS_FUELS	8
7	FVS_HRV_CARBON	8

ID: 4 Table: FVS_POTFIRE Table count: 1

PRE-TREATMENT SEQNUM **POST-TREATMENT SEQNUM**

Cycle 1 1 3

Cycle 2 4 6

Cycle 3 7 9

Cycle 4 10 13

Choose a SEQNUM Assignment Template
Mgt4X_WtdMean FFE
Assign

Packages

001,002,003,004,007,008,009,010,011,020,022,023,025,026,029,030,101,102,103,104,107,108,109,110,111,122,123,125,126.

Edit Delete Customization Save

Help Done Cancel Close

The Packages list is display only and not changeable. As of BioSum version 5.10.0, all packages must be assigned the same sequence number pattern—as we've concluded that any other more flexible arrangement only guarantees more confusion than value.

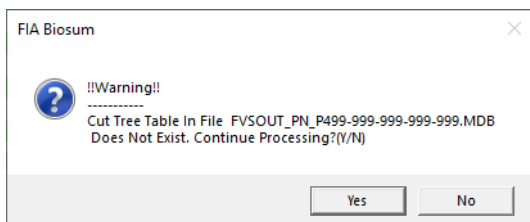
If after assigning sequence numbers to a table, you decide not to load that table's data, simply select the table and choose "Delete Customization".

After all customizations are complete (or at any time in the process), click “Save” to save your work. Selecting “Close” will return you to the main Join and Append dialog.

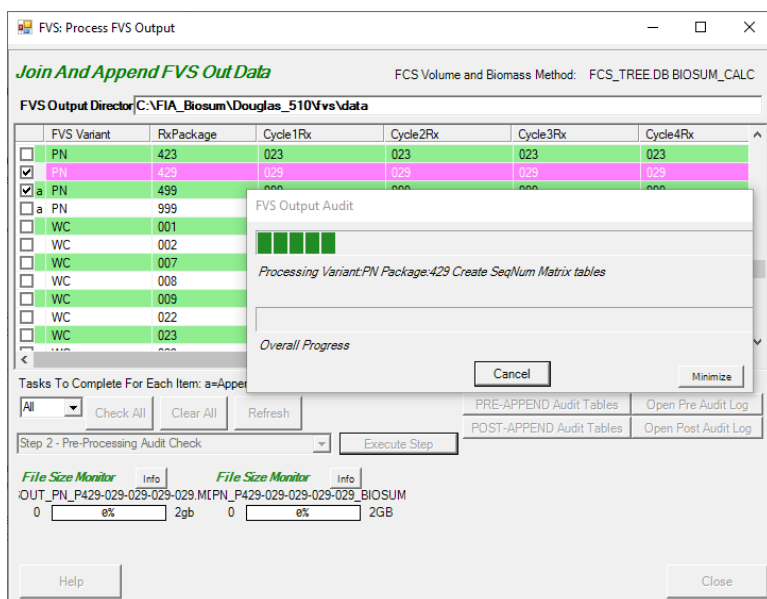
Steps 2 and 4 in the Join and Append Dialog are audit steps that assess data readiness and perform QA chores on the package-variants with checked boxes.

Select Step 2 <Pre-processing Audit Check> and “Execute Step” to run an audit on the output databases. During this audit each table in the FVS output is analyzed and PRE and POST year is assigned to each cycle. The analysis is written to several tables with the table name format of audit_pre_post_rx_year_fvsoutputtablename. Additionally, the data from the FVS_summary table is analyzed to determine if the year in the table is represented in the other tables.

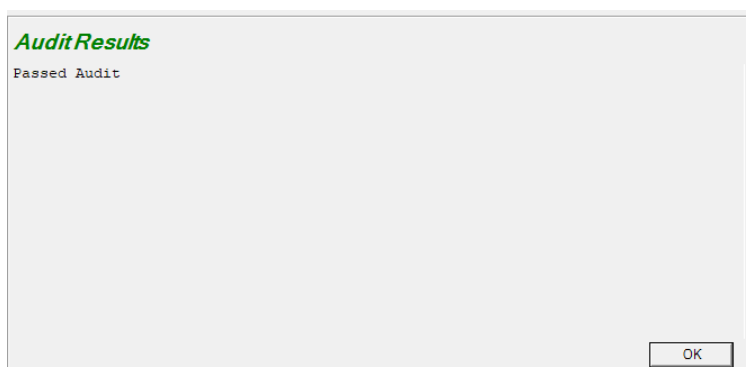
If BioSum detects an error or inconsistency in the data, a pop-up window will appear. If there are errors not significant enough to affect processing, the audit will pass with a warning. It is up to the user to determine if warnings require action. More significant errors will return a failed audit result. The user must address the specified error before continuing. Packages that don’t contain mechanical thinning (such as a grow-only simulation) and therefore do not contain a CUTLIST, will receive a message asking if you want to continue without a cutlist. Select <Yes> to continue. Packages without a CUTLIST will technically fail the audit, but if other issues aren’t present in the audit results log, the audit is functionally successful.



The audit operation builds and analyzes temporary databases that live within your windows\user\[username]\appdata\temp folder. The sizes of these databases as the audit proceeds are monitored with progress bars as a diagnostic in case a glitch occurs owing to exceed the MS Access maximum database size allowed (2 GB).



When the audit concludes, a dialog appears returning an audit status.



Detailed information about any issues that arose can be viewed by summoning the pre-append audit log and/or pre-append audit tables

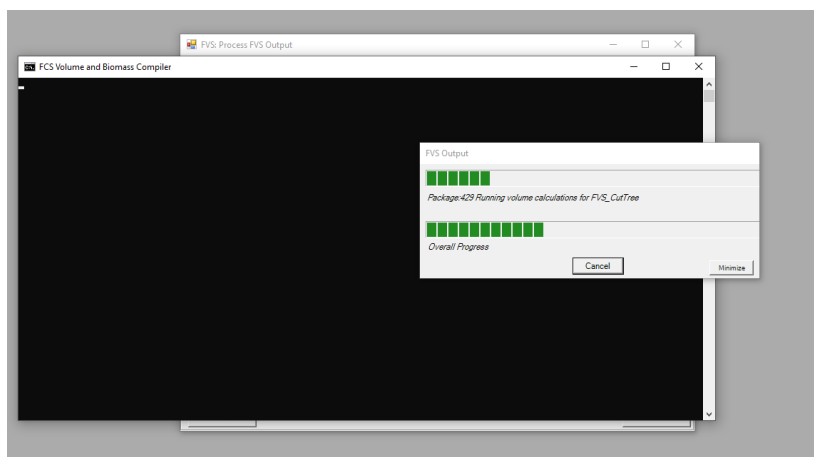


Step 3 loads the FVS output data into BioSum PrePost databases (in the project's fvs\db folder) and the cut tree list in the FVSOUT_TREE_LIST.DB (in the project's fvs\data folder). Execute Step 3 (<Append FVS Output Data>). Pre and post record values for all FVS output tables (except the atrList, treeList, and cutList tables) are appended to tables in the project's fvs\db folder. For example, pretreatment SUMMARY records are written to PRE_FVS_SUMMARY table and post treatment records to POST_FVS_SUMMARY

table based on the sequence number assignments selected in the previous steps. The BioSum Optimizer module then uses these tables to define treatment effectiveness.

Biomass and volume of the trees specified as harvested in the FVS CutList table are calculated using regional and species specific formulas. This process runs within Java. BioSum updates the fvs\data\FVSOUT_TREE_LIST.DB database with the calculated volumes and biomass. Calculated volumes and biomass are written, for example, to the VOLCFNET, VOLCFGRS, VOLCSGRS, DRYBIOM, and DRYBIOT columns in the CUTLIST table. These values are used to calculate harvest value and costs in BioSum Processor.

A warning dialog for packages with no harvested trees can be cleared—this is typically displayed when a package is grow-only (so no cut trees will be written to FVSOUT_TREE_LIST.DB) but such packages nonetheless must be loaded into BioSum (to populate the PrePost databases). When the loader calculates volume and biomass of the projected trees, you will see an empty black window appear for a short time.



Finally, once Step 3 is complete, run step 4 (<Post-Processing Audit Check>) to ensure all necessary data are present and formatted correctly. If errors exist, evaluate and rectify before moving on to the Processor module.

It is essential to complete the post-load audit to ensure data integrity, step 4. Audit results will post to the Audit Results dialog and be written to the log file that can be accessed via the “Open Post Audit Log” button. It is common for there to be warnings about species changes, almost always due to the mapping of species that must be specified in the Tree Species table to address the fact that some trees in an FVS variant will not be directly supported as the species that they are.

After successful loading and a clean bill of health from pre- and post-append audits for every variant package combination, one can proceed to the Processor module. There are

two optional tasks that can be completed here at any time—they source data from the FVSOOut.DB and do not depend on a successful append operation. They also act on all package combinations represented in the FVSOOut database (ignoring the checkboxes).

The "Create FVSOOut_BioSum.DB" task writes an SQLite database, sourced from FVSOOut.DB, that formats all data as data types that are fully compatible with MS Access. This can be very helpful if needing to work with FVSOOut data from MS Access, as the native data types in that database (such as LONG TEXT, which translates to Memo in Access) are nearly impossible to work with (e.g., sort, filter, etc.). As this is not a small file, it only makes sense to create it if you believe you will use it. Moreover, it can be created at any time, as it is not used in any BioSum module.

The "Write FVS_InForest to FVSOOut_TREE_LIST.DB" task copies all data from the TREELIST in FVSOOut.DB into the FVSOOut_TREE_LIST.DB (in the project's fvs\data folder). This can be useful if needing tree level biomass or carbon data, for example, at all time points in the simulation for analysis beyond what the BioSum framework manages directly. As this is not a small file, it only makes sense to create it if you believe you will use it. Moreover, it can be created at any time, as it is not used in any BioSum module.

Additional columns in the data grid (which can be scrolled to see) provide status information that is sometimes useful for diagnosing issues.

Advanced Topic: Creating FVS Input Databases using BioSum

Aside from the many potentially fruitful applications of a full BioSum analysis, this software can also be useful when there is a need to create a variant-specific FVS input database from data formatted consistent with FIADB but not downloaded from the Datamart. This generates an Access .ACCDB file (not an SQLite file) that can be read by either FVS Online or the deprecated FVS Suppose. This use case is outside of the BioSum scope documented in this User Guide, but workflow that has successfully accomplished this task for FIADB formatted spatially intensified off-grid plots that are not part of the FIA stratification, entailed the following steps:

1. Start with a DATAMART originated FIADB SQLite database, copying the table structures for PLOT, COND, TREE, SITETREE, and SEEDLING into a new SQLite database. This can be accomplished by copying just the data structures or copying the tables, then deleting all records in these tables to create a template.
2. Copy the table structures for the four POP_ tables (EVAL, ESTN_UNIT, STRATUM and PLOT_STRATUM_ASSGN from an existing BioSum project into your SQLite template).
3. Populate the PLOT, COND, TREE, SITETREE, and SEEDLING tables in the template via queries from your custom FIADB style data source.

4. Populate the POP tables to enable the data to load into BioSum so that an FVSIn.accdb can be created. The simplest POP table quartet that will enable a BioSum load won't necessarily generate valid area expansions for each plot, but if that is not important to the application for which the FVS input file is required, then that is not a problem. You'll need a single record in each of three of the tables (ESTN_UNIT, EVAL and STRATUM) – that is to say one estimation unit, one evaluation, and one stratum, and as many records in PLOT_STRATUM_ASSGN as exist in the PLOT table (referencing the one record in each of the other 3 POP tables).
5. When creating the FVSIn.accdb (which will appear in the variant folder, along with an empty FVSIn.DB), choose for the input database that contains FVS_STAND_INIT and FVS_TREE_INIT any FIADB database that does NOT contain records for your plots in those tables (e.g., could even be from another state). BioSum will then use its old workflow to populate the FVSIn.accdb without relying on FIADB INIT tables.

If contemplating this use case and in need of the most current guidance on this use of BioSum, contact BioSum support.

Literature Cited

- Crookston, Nicholas L.; Gammel, Dennis L.; Rebain, Stephanie; Robinson, Donald; Keyser, Chad E.; Dahl, Christopher A.; David, Lance R.; Shettles, Michael A. 2003 (revised November 2022).** Users Guide to the Database Extension of the Forest Vegetation Simulator Version 2.0. Internal Rep. Fort Collins, CO: U. S. Department of Agriculture, Forest Service, Forest Management Service Center. 60p
- Rebain, Stephanie A. comp. 2010 (revised February 1, 2022).** The Fire and Fuels Extension to the Forest Vegetation Simulator: Updated Model Documentation. Internal Rep. Fort Collins, CO: U. S. Department of Agriculture, Forest Service, Forest Management Service Center. 409p.
- Dixon, Gary E. comp. 2002.** Essential FVS: A user's guide to the Forest Vegetation Simulator. Internal Rep. Fort Collins, CO: U. S. Department of Agriculture, Forest Service, Forest Management Service Center. 226p. (Revised: August 24, 2022)
- Van Dyck, Michael G; Smith-Mateja, Erin E. (Editors). 2000.** Keyword Reference Guide for the Forest Vegetation Simulator. Fort Collins, CO: U.S. Department of Agriculture, Forest Service, Forest Management Service Center. 118p. (Revised: December, 2022)
- Vandendriesche, Donald A. 2009.** An Empirical Approach for Estimating Natural Regeneration for the Forest Vegetation Simulator. U.S. Department of Agriculture, Forest Service,

Rocky Mountain Research Station. Ogden, UT. Proceedings: National Silviculture Workshop; June 15-19, 2009. Boise, ID.

Vandendriesche, Donald. 2010. FVS out of the box-assembly required. In Jain, Theresa B.; Graham, Russell T.; and Sandquist, Jonathan, tech. eds. 2010. *Integrated management of carbon sequestration and biomass utilization opportunities in a changing climate: Proceedings of the 2009 National Silviculture Workshop*; 2009 June 15-18; Boise, ID. Proceedings RMRS-P-61. Fort Collins, CO: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station. 351 p.
http://www.fs.fed.us/rm/pubs/rmrs_p061/rmrs_p061_289_306.pdf