User's Guide to the Forest Vegetation Simulator (FVS) and Suppose Interface

Robert Seymour
Univ. Maine School of Forest Resources
SFR 409 – Forest Ecology and Silviculture Lab
Updated September 2003 2007, 2010, 2012 by R. Seymour
Revised 2013 by R. Seymour to incorporate the Database Extension
Revised April 2014 to clarify FVSStand and SVS postprocessors
Further extensive additions in 2016, 2017; eliminated instructions for non-database usage
Comments are welcome; please email to rseymour@maine.edu

Introduction

The Forest Vegetation Simulator (FVS) has become the national standard forest modeling system, and has undergone extensive development by the USDA Forest Service. Regional variants are currently available for anywhere in the United States, and are continuously being modified and improved. We will use the ***Northeast Variant*** developed in the late 1980s by scientists at the US Forest Service Northeastern Forest Experiment Station and revised many times since then. FVS can be used to predict growth of individual stands or large forest areas, depending on the objective

Information on FVS is available on its website, *http:// www.fs.fed.us/fmsc/fvs/index.shtml* from which all the software and much documentation can be downloaded. The model was developed in the Rocky Mountain region and was known originally as PROGNOSIS. See also "Essential FVS: A Users Guide to the Forest Vegetation Simulator" by Gary Dixon, a 228 page pdf file, that covers all workings of the model quite thoroughly, stored in the Class Folder for SFR 409.

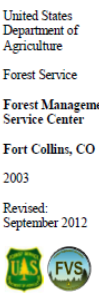
General Guidelines

File Management

There are two basic and fundamentally different ways to run FVS: (1) Creating input text files using the built-in editors, or (2) via the "Database Extension" where all data are stored in tables within a Microsoft Access database. The Forest Service has stopped supporting the first option, and strongly urges all users to migrate to the Access approach. This manual includes instructions to do so, and assumes that users have use of the Access software. This is not essential; one can still use the "old fashioned" process to create and edit the input files. However, for brevity, I have deleted the "old pre-Access instructions; if you absolutely need to work without the Access interface, I can send you an old version of this manual.

The Database Extension is fully described in the following manual, available at the website given earlier:

Crookston, Nicholas L.; Gammel, Dennis L.; Rebain, Stephanie; Robinson, Donald; Keyser, Chad E. 2003 (revised August 19, 2011). 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. 56p.


United States Department of Agriculture
Forest Service
Forest Management Service Center
Fort Collins, CO
2003
Revised: September 2012
FVS

Users Guide to the Database Extension of the Forest Vegetation Simulator Version 2.0

Regardless of how you create and store your input data, simulations are assembled and executed with the **SUPPOSE** graphical interface, using ***keywords***.

Despite the appearance of being a fully functional Windows application, Suppose has a few artifacts remaining from its original MS-DOS heritage. The most serious one is that it cannot handle blanks in file names or subfolder names. ***Never include spaces in the name of any file or subfolder***. Try to keep names short, and use the underbar character ("_") if you must break up long names into intelligible segments. If you encounter problems with tree lists that seem to be blank, then it's likely that you've got a space somewhere in a subfolder name or file name. Suppose will read only to the blank space, and won't recognize the portion of the file name afterwards.

Downloading the FVS Software from the FVS Website (checked Sept. 15, 2016)

Steps:

1. Open a web browser and navigate to **http://www.fs.fed.us/fmsc/fvs/**

2. Click on the "Software" heading.
3. Click on the "Complete Package" link.
4. Under "Download", simply click on the FVS_Setup.exe, and save this Download to your desktop. Double-click this .exe file and hit "Run" to install the program code and related files. If you have an older version of the software, the Install will write over it.
5. There should now be a "FVS-Suppose" icon on your desktop.

When you download the FVS software, two folders are created:

C:\FVSbin where all the program files are stored; and

C:\FVSData, where your input and output data are stored. Always keep your working database and simulation file (the *.key) in this folder and you will avoid problems with file management.

Because I use FVS for multiple classes and projects, I make subfolders under FVSData to keep files separated. If you do this, make sure that the subfolder names do NOT contain blanks!

The software installer will also create a folder "Blank-Database" in the FVSdata Folder. In this folder you will find the MS Access database "Blank_Database.accdb" along with the file "Suppose.loc". (The necessary FVS ***"Locations File").*** The instructions that follow will use this Blank Database as a starting point.

By default, output files will be sent to the folder where the input files are stored. This is governed by the subfolder that holds the ***Simulation File (*.key).*** The Simulation file is created when you click on the Run Simulation bar, at which point you can give it a unique name and subfolder. Simulation files are discussed in more detail below.

Quick Overview and Checklist for Running an FVS Simulation

This list is meant to be a quick reminder of the steps involved with using FVS to project the growth of a stand. Each item is covered in more detail in the text; follow the hyperlink to learn more about each item. This list assumes you've already created the Tree List, Stand List, and Locations Files using the various editors under the *Data Preparation* menu, or have successfully entered your data into the predefined tables in a Microsoft Access database. (Instructions later in this document.)

1. Close your MS Access database with the TreeInit and StandInit tables.
2. Open Suppose.

3. Open an existing Simulation file (.key), if one exists (File, Open). Skip to Step 4.
4. If you're creating a new simulation, then open the appropriate Locations File *(File, Select Locations File*). This will automatically link the Stand List and Tree List files, assuming the file references are correct. If using the Database Extension, this will also give FVS the database name and add SQL code (under the "From Database" keyword) to read the tree and stand data.
5. Click the Select Simulation Stands menu and make sure the correct stands are in the simulation. (If you have problems here, edit the Locations and Stand List files to ensure they refer to the correct subfolders and file names.)
6. Click Set Time Scale to verify that the simulation length and output intervals are OK.
7. If you want output tables, click Select Post Processors to make sure you've selected FVSStand (and SVS, if you want to generate visuals from FVS).
8. Click Use FVS Keywords to select FVSStand, along with any other keyword you wish to invoke. Select the NoTriples keyword to avoid tripling tree output records.

9. If you want to run SVS as a stand-alone application, make sure to select the TreeList and CutList keywords multiple times (every future output date) to generate the necessary tree output records. (This is *not* necessary just to run FVSStand.)

10. Click Management Actions and specify the silvicultural treatments you want to apply.

11. Click Edit Simulation File and verify that everything is correct.

12. Click Run Simulation to invoke the FVS simulator and post-processors.

13. Examine and print (if desired) the output tables from FVSStand or other post-processors.

Output files should all be written to the subfolder where the Simulation file was saved, and to the Output Database if using this option. To avoid confusion, review the FVS file types and their extensions, and set your computer to recognize them in the Explorer Details View.

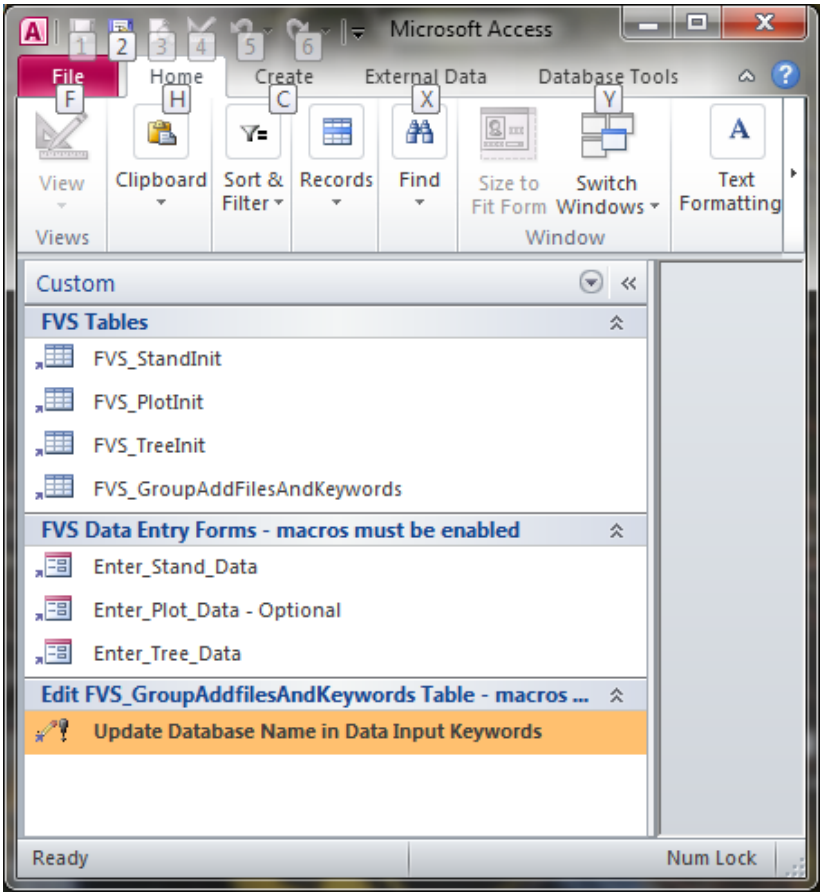
Creating and Editing Input Data

Create the Tree Data Table (Database) or "Tree List File " (*.fvs text file)

The Tree List File is the place where all the tree measurements taken in the field are stored. At minimum, FVS needs a tree number, plot number, species, and dbh. Users can also input total height, live crown ratio, recent dbh and height growth, various condition codes, and other variables if these are measured or estimated. The model will accept trees as small as 0.1 inches dbh, and assumes English units.

Option A: Using the Database Extension via Microsoft Access

Open the Blank_Database and you should see the following:



If you're new to Access, then a brief lesson in Objects is in order. Access stores all its data in TABLES, which appear here under the custom bar "FVS Tables". Tree data are stored in the ***FVS_TreeInit*** table; stand data are in the ***FVS_StandInit table***. Both are required. The ***FVS_PlotInit*** table is optional, and used only if you want to "grow" each plot separately in the simulation. ***The FVS_GroupAddFilesAndKeywords*** table contains the essential SQL programming code which tells the Suppose program where to locate tree and stand data. The Update Query ***Update Database Name in FVS_Groups Table*** inserts the name of your specific database in place of the "Blank_Database" names in the SQL code.

Do NOT rename these tables! Names are used for other programming tasks which won't work if you change them.

The Forest Service has developed **Access FORMS** which use macros to allow you to enter data conveniently. If you're starting from scratch with a small dataset, then these forms can be useful and we recommend starting with them as you're learning.

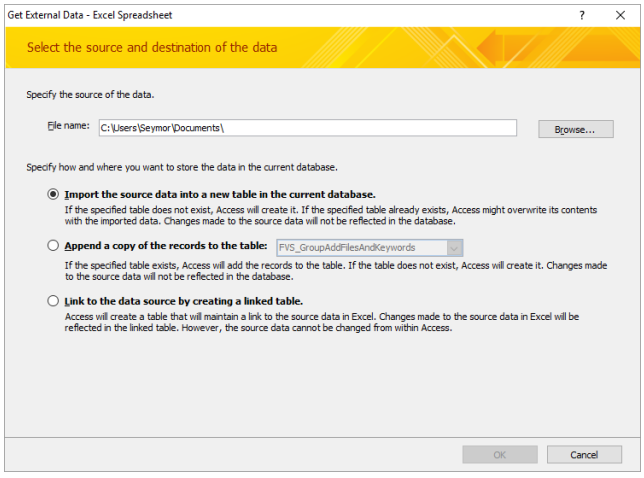
Importing the Tree List from EXCEL

If you already have your data in some kind of table (like an Excel Worksheet), then you can **IMPORT** this file to the database using Access's ***External Data*** function, renaming it as appropriate, and bypassing this built-in editor. If you import a table from Excel or other source, then the column headings (the variable names used by the FVS model) must be exactly those specified in the FVS_TreeInit table. One way to ensure that this happens is to EXPORT the FVS_TreeInit table as an Excel file, then enter your data as you would in any other analytical project using Excel. Once the data are entered, you can then re-IMPORT the populated table back into Access. ***Important note:*** Sometimes exporting, then re-importing, creates formatting problems with these tables that are not obvious and difficult to correct. Usually this is because a column that must have numeric input will be incorrectly formatted as text, and Suppose will not read the correct data. So, while it may work, I do not recommend this method.

Linking to stand-alone EXCEL Worksheets – HIGHLY RECOMMENDED to avoid format issues!

IMPORT/EXPORT problems can be eliminated by keeping your data in an EXCEL file with a "FVS_TreeInit" worksheet, and simply LINKING it to ACCESS, not IMPORTING it.

1. Open Access. On the main ribbon, click "External Data", then "Excel" and you should see:



2. In the Browse box, navigate to the Workbook that contains your FVS_TreeInit worksheet (created previously). Note that the default is the first option (Import into a new table….).

3. Check the ***bottom*** button ("Link to the data source…"), then proceed through the next several prompts which ask you to select the correct worksheet in the target Workbook, check the "Column Headings" box, and verify the Linked Table Name (which must be FVS_TreeInit).

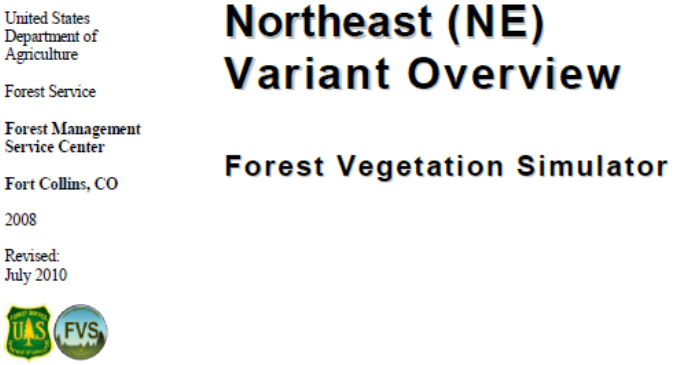
4. If this fails, you may need to first delete the existing FVS_TreeInit file from the Access Database, then repeat the steps above.

Once you have this Excel worksheet linked to Access, ***it must remain in the same drive and folder on your computer!*** However, this also means that you can change just the EXCEL tree list file, and the changes will automatically be available to your Access database, with no need to move files in and out of Access.

Tree Data

The Table 3.0.2 on the next page (from the Database manual) shows all the possible variables than can be recorded for each tree in the sample. In practice, only a few are required: the Stand_ID, Tree_ID, Species, and DBH. Other fields are optional and can be left blank or even omitted. All should be self-explanatory. Some may not even be used or available for some FVS regional variants. For example, the Northeast variant does not currently use the CrRatio field, unlike many of the other variants. Most of these variables are probably self-explanatory and are the same as used in the "old" tree list file. See Appendix A in this document, or the discussion below (Option B) for a further elaboration on what these variables mean.

Make sure to use the correct two-letter species codes, which may differ among regions. These can be found in the Enter_Tree_Data form, or in Table 3.2.1 in the documentation manual for the Northeast Variant:


United States Department of Agriculture

Forest Service

Forest Management Service Center

Fort Collins, CO

2008

Revised: July 2010

Northeast (NE) Variant Overview

Forest Vegetation Simulator

Keyser, Chad E., comp. 2008 (revised July 29, 2010). Northeast (NE) Variant Overview – Forest Vegetation Simulator. Internal Rep. Fort Collins, CO: U. S. Department of Agriculture, Forest Service, Forest Management Service Center. 36p.

Table 3.0.2 – The predefined table structure used for initializing tree information when using the TREESQL keyword.

Column Name	Data type	Description
Stand_CN	Char(40)	Same as Stand_CN in Table 3.0.1. Not read by FVS, but may be used for querying purposes.
Stand_ID	Integer or Char(26)	Same as Stand_ID in Table 3.0.1.Not read by FVS, but may be used for querying purposes.
StandPlot_CN	Char(40)	Same as StandPlot_CN in Table 3.0.1. Not read by FVS, but may be used for querying purposes.
StandPlot_ID	Integer or Char(26)	Same as StandPlot_ID in Table 3.0.1. Not read by FVS, but may be used for querying purposes.
Tree_ID	Float	Tree Identification Code
Plot_ID	Float	Plot Identification
Tree_Count	Float	Tree Count
History	Float	History Code 0-5 are live trees, 6 and 7 died during mortality observation, 8 and 9 died before mortality observation period
Species	Char(8)	Tree Species Code, can be the FVS alpha code, FIA code or USDA plant symbol.
DBH or Diameter	Float	Diameter at Breast Height in inches. Diameter is an alias for DBH in this version.
DG	Float	DBH Growth in inches (not tenths of inches).
Ht	Float	Height in feet
HtG	Float	Height Growth in feet
HtTopK	Float	Height to top kill is the height to the point of the tree of top kill in feet
CrRatio	Float	If the number is 0-9 then it is considered a crown ratio code, according to the FVS documentation. If the number is 10-99 the value is considered a percent live crown.
Damage1	Float	Damage Code, see the FVS documentation for details
Severity1	Float	Severity Code corresponding to damage code 1
Damage2	Float	Second damage code.
Severity2	Float	Second severity code.
Damage3	Float	Third damage code.
Severity3	Float	Third severity code.
TreeValue	Float	Tree Value Class Code 1 for desirable, 2 for acceptable, 8 for non-stockable and any other number represents a live cull
Prescription	Float	Prescription code
Age	Float	Age of the tree record
Slope	Float	Slope Percentage on the plot where the tree was located
Aspect	Float	Aspect in degrees on the plot where the tree was located
PV_Code or Habitat	Integer or Char(10)	The potential vegetation code on the plot where the tree was located (see notes in table 3.0.1 for the stand).
TopoCode	Float	Topography Code 1=bottom, 2=lower, 3=mid slope, 4=upper slope, and 5=ridge top, on the plot where the tree was located
SitePrep	Float	Site Preparation code 1=none, 2=mechanical, 3=burn, and 4=road cuts/road fills/stockable road beds, on the plot where the tree was located

Important! The key linkage between the Tree Table and the Stand Table (described below) is the *Stand_ID* field in both tables. This is the first box in the input form, labeled "Stand Identifier". This is read as text, even if you use numbers, so make sure you use exactly the same characters as you did in this field of the FVS_StandInit table.

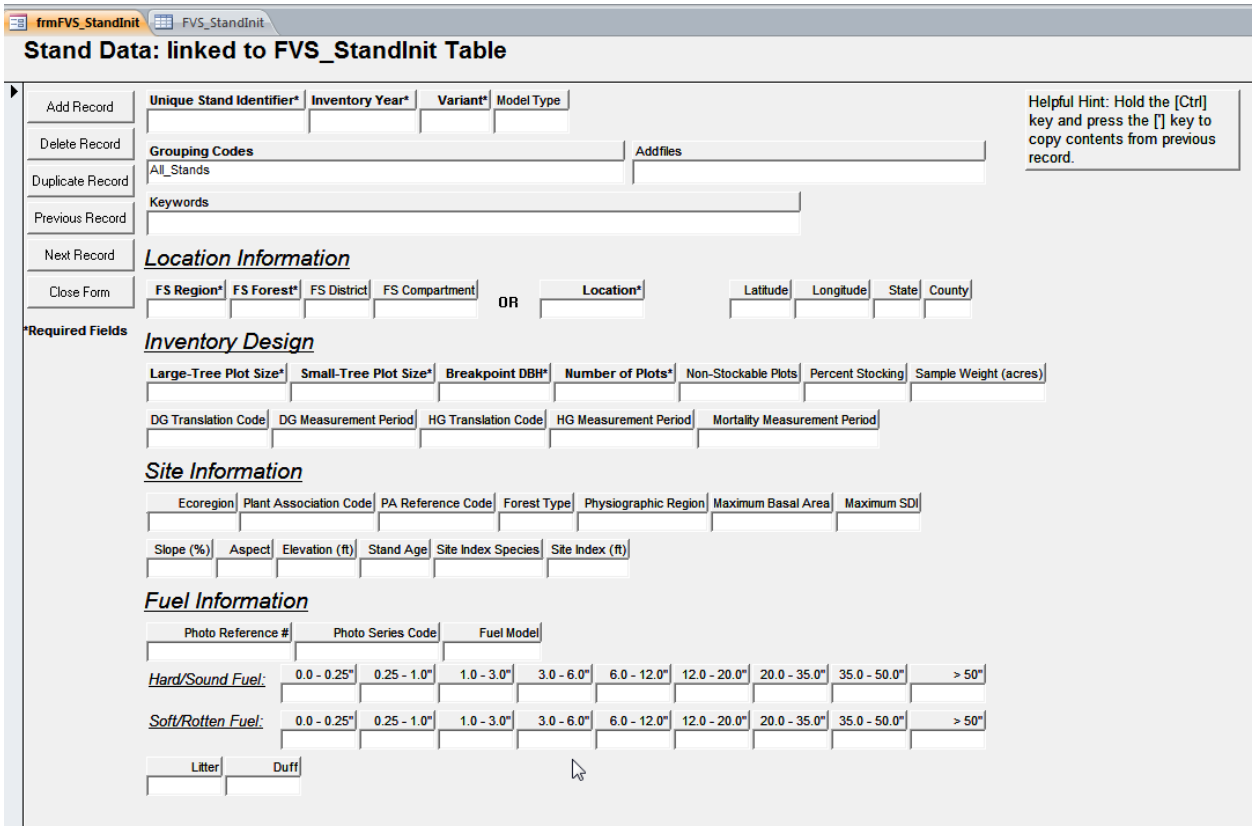
Create the Stand Data Table (Database) or "Stand List File" (*.slf text file)

The Stand List file tells FVS what sort of sampling method was used to collect the tree data, and records other stand-level information such as site index, topographic variables, and geographic location. The Stand List File can also be used to combine individual plots or stands into user-defined groups, thereby

avoiding the need to combine simulation output manually from several stands. A Stand List file may include a reference to only a single tree list (one stand) or dozens of tree lists (stands) that users may wish to combine in various ways. If desired, Suppose will "grow" and "manage" each stand (Tree List) separately, then combine the output into a single set of tables. A common application would be a large woodlot with several different stands that warrant separate silvicultural treatments and growth projections, but that you would eventually want to combine into a single set of output tables.

Option A: Using the Database Extension via Microsoft Access

Here we must populate the Access table "FVS_StandInit". As with the TreeInit table, you can use the "Enter_Stand_Data" form, or create the table yourself in Excel and import it into the database. Because the number of stands is usually small, sometimes only one, it arguably makes more sense to just use the form below to enter the data. Just open your Access database and double-click on the "Enter_Stand_Data" Form, and you will see the following:



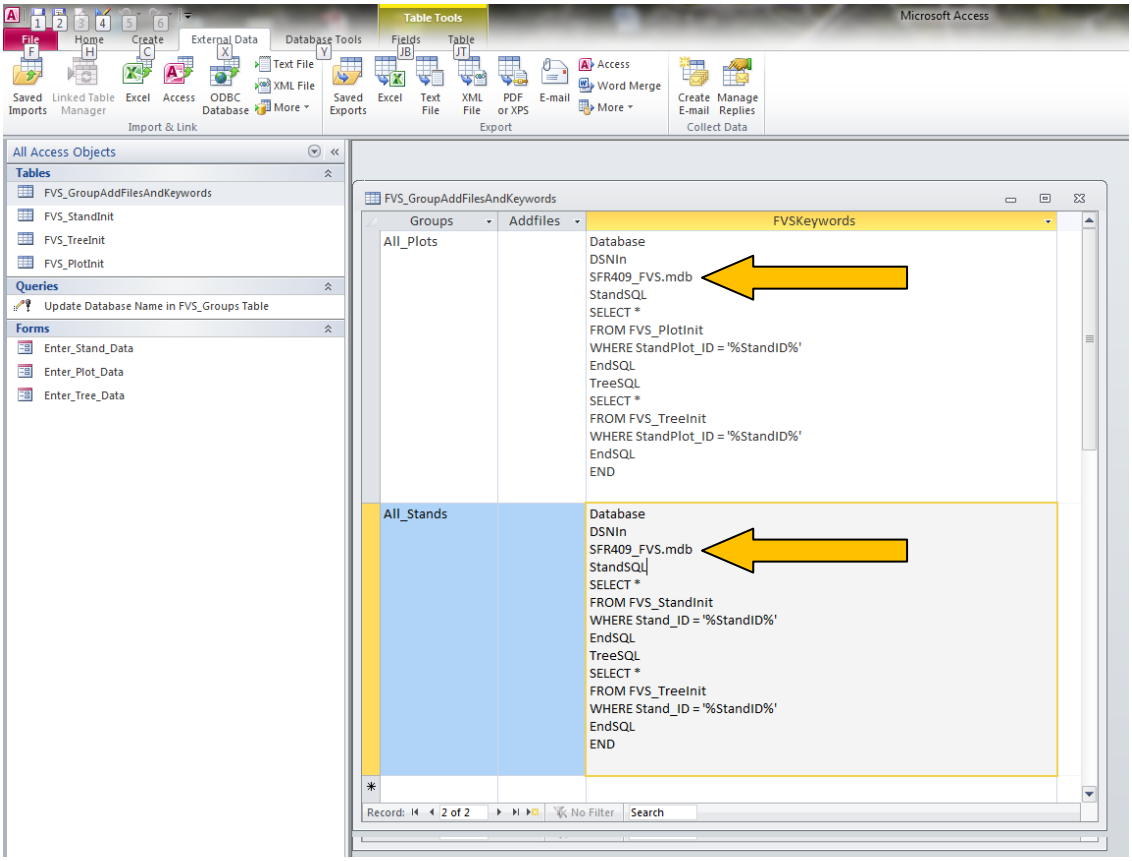
Fill in the required fields (**in bold-faced text**) along with any other fields for which you have data (e.g., Grouping Codes or Site Index and the Site Species). As with the tree data, many fields are optional. Note that you can hover the cursor over the labels above the boxes and see an explanation for each field.

Important! The key linkage between the Stand Table and the Tree Table is the Stand_ID field in both tables. This is the first box in the input form above, labeled "Unique Stand Identifier". This is read as text, even if you use numbers, so make sure you use exactly the same characters as you did in this field of the TreeInit table.

Also important: Do NOT delete the line that contains the text "All_Stands" in the Groups column! If you do so, Suppose will not automatically create the "From Database" keyword (really a compute statement) when you open the locations file and select stands for the simulation. The "From Database" keyword contains the SQL language Suppose needs to "find" and read the Tree and Stand tables in your database.

Add your database name to the SQL query language

Once you have your tree and stand data successfully entered into the respective Access Tables, save the database using the ***File, Save Database As*** commands on the main menu bar. The final step is to change the "blank_database" name to your specific database name in the **FVS_GroupAddFielsAndKeywords** table. Here, just double-click on the Update Query (**Update Database Name in FVS_Groups Table**) and you will first get a warning (respond "yes") and a notification that you're updating two rows in the table. Open the **FVS_GroupAddFielsAndKeywords** table and you should see the following, where your specific database name (in the case **SFR409_FVS.accdb**) has been substituted (at the arrows).



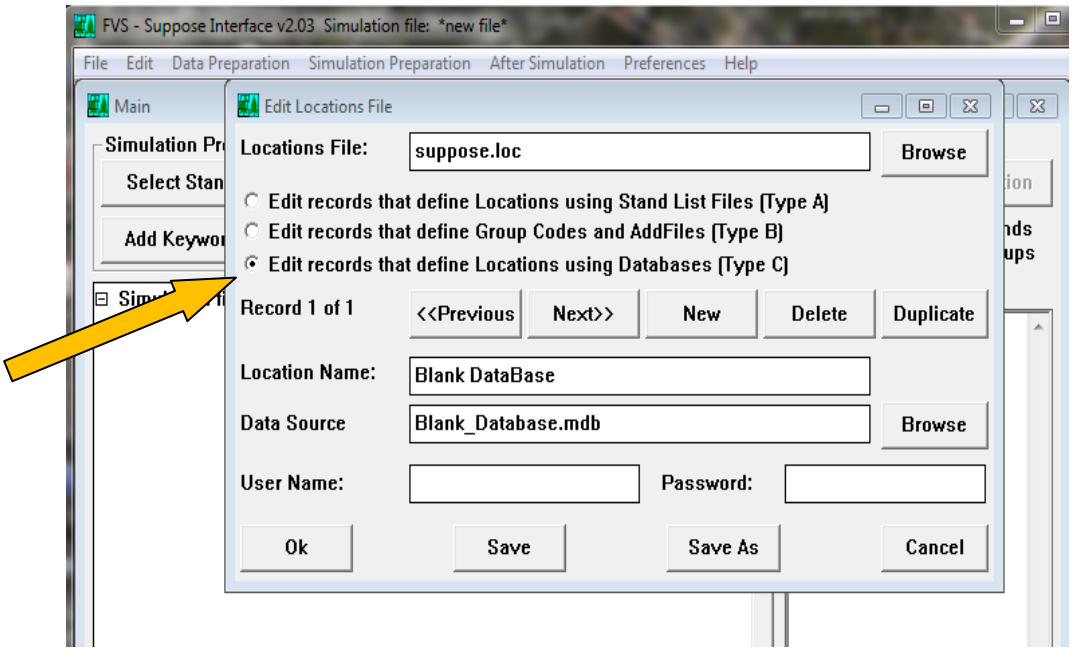
Create the Locations file (*.loc)

When using the Database extension, the Locations File is necessary to "tell" FVS that you're using a database as input (***Type C*** records). The default is Type A records, which refer to the old text file structure.

Option A: Using the Database Extension via Microsoft Access

When the FVS software installs on your hard drive, it creates folder **FVSData.** Within this you will find a folder **Blank-Database**, which contains an Access data base with unpopulated tables, and the file **Suppose.loc.** Copy both of these files to your working folder (where you're storing input and output data for your particular project).

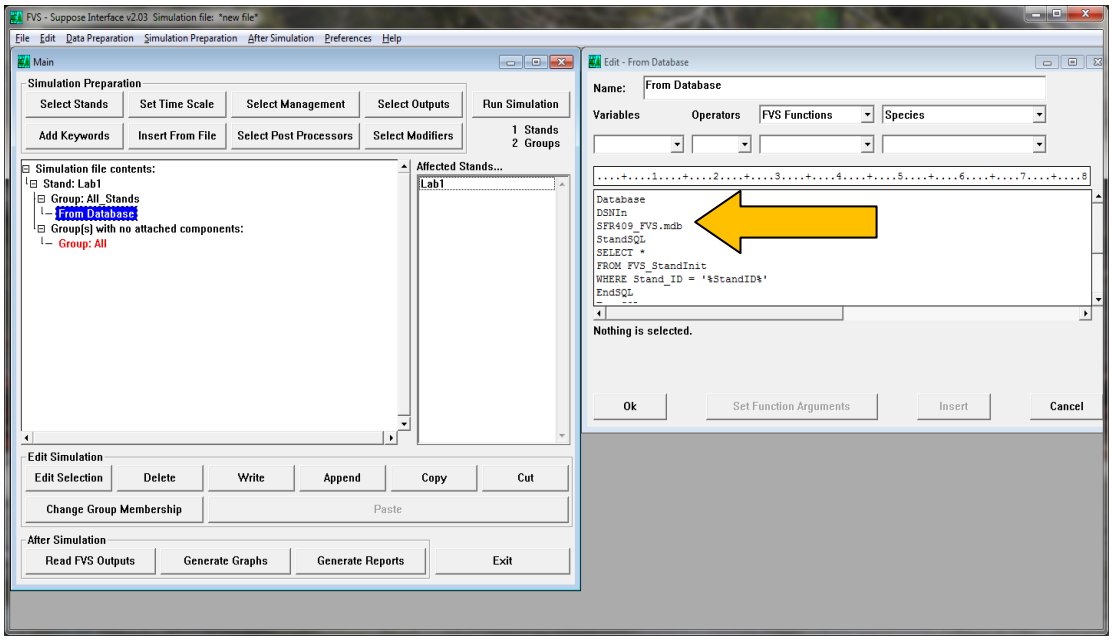
Next, open the Suppose Interface program. On the main menu bar, click "Data Preparation", then "Modify/Create Locations File". Check the third radio button in the list just below the "suppose.loc" box, labeled "Edit records that define Locations using Databases (Type C). You will see the following window:



In the "Location Name" box, enter any descriptive term(s) that identify your project, typically a forest or woodlot that contains multiple stands (e.g., **SFR409_Labs**). Next, rename, or browse for, the database that contains your input data, and make sure the "Data Source" box contains this file name. It's a good idea to include the path here too, especially if you're using multiple folders. You may also want to rename the location file itself by changing the generic "suppose.loc" to something like **SFR409.loc**. (Do not change the loc suffix!). When you're done editing/entering names, hit the Save or OK button and the Location File should now be specific to your input database. ***Important: This is the file that you will open first to start a new simulation.***

To verify that the Locations File is working, close and reopen Suppose. Click ***File, Select Locations File*** and browse to select it. Suppose will open the **Select Simuation Stands** window. The left-hand box should show your **Location Name**. If you click on this and follow the procedure below under **Select Simulation Stands**, you should see a **From Database** keyword in the left-hand simulation file window once a stand has been selected. Double-click on this line and Suppose will open a window to the right with the SQL

statements required to read your database which include the StandSQL and TreeSQL keywords. Here the database is named SFR409_FVS, shown immediately below the DSNIn line:



If you get this far, your simulation should work fine.

If the From_Database statement does not appear, this likely means that the All_Stands group code is missing from the Stand_Init file (as described in the warning above). Evidently this is what triggers Suppose to add the SQL text when you add a stand.

Running a Simulation

Select Simulation Stands

Once you've created the Tree List, Stand List, and Locations Files, or created these tables in your Database, you're ready to run a simulation. First, you must tell Suppose which Stand or Stands to include in the simulation, as follows:

1. Open Suppose.

2. If you have previously created a Simulation File (*.key) to store various run options, then open this first by clicking on File, Open, and navigating to the appropriate file. [Note that the Open, Save, and Save As commands on the File menu apply to the Simulation file, not to the Locations File which has its own "open" command.] If you're constructing a new simulation, skip this step; Suppose will allow you to save the simulation file just before you click "Run Simulation."

3. Under File on the Menu bar, click **Select Locations File**.

4. Navigate to the subfolder that holds the correct Locations file, and double-click to activate.

5. Click on the Select Simulation Stands button and you should see three columns. The left column should now be headed by the Location Name you assigned to this Locations file.

6. Click the All Stands button at the bottom of the third column. The box above should now be filled with the Stand Names that are linked to each Tree List file. (There may be only one, if that's all you've entered in the Stand List file.)

7. In the right-hand column, highlight a Stand or Stands that you want to include in the simulation.

8. On the lower left part of the screen, click Add *n* Stand *(n* is the number highlighted).

9. The message above this box should now read: Contents: *n* Stands …. where *n* is the number of stands included.

10. Click Close to take you back to the main Suppose menu.

11. The only way to see a list of stands selected for simulation in this menu is to click the "Delete Stand" button (!), then click the down arrow on the drop-down list box. This will show all stands that will be grown by FVS. Delete any that you don't want, then Close.

You're now finished communicating Stand Data to Suppose. You still need to specify other options such as how long you want the simulation to run, what sort of output is needed, what management treatments to apply, and other factors. Each of these has its own box:

Set the Time Scale

1. On the main Suppose screen, click the Set Time Scale box.

2. The Common starting year should be the same as the Inventory year in the Tree List file.

3. The Common cycle length determines how often output tables and graphs are produced. Generally, 10 years is the best choice here, although there may be times when you want to specify 5-year or even 1-year output.

4. The Common Ending Year determines how long the simulation will run. This should be some multiple of the Cycle Length. For example, if you want to make a 50-year simulation beginning in 2000, then enter a Cycle Length of 10 years and 2050 for the ending year.

5. Hit OK to return to the main menu screen.

Select Management

This is where you indicate what treatments, if any, are to be applied to the Stands. The best single reference here is *Evaluating Stand Density Management Alternatives Using the Forest Vegetation Simulator (Dixon 1998)*. If you invoke treatments via the "Management Actions" button, you will get more descriptive titles for the various options. These can also be invoked directly by the keyword itself if you know what you want to do. For example, if you click on the Select Management button, then "Thinning and Pruning Operations", you will see (for example) "Thin throughout a diameter range"; using the "Add Keywords" button, you need to know that this is the "ThinDBH" keyword.

In the Northeast, our stands are generally so species-diverse that the simple prescriptions (e.g., Thin from Below, BA) cannot capture many refinements needed in a typical treatment. Generally, we have found the most flexible options to be:

1. Thin from a specific diameter range" [keyword **ThinDBH**]. This option can replicate any possible combination of prescription elements, by allowing the user to harvest a defined percentage (the "cutting efficiency" in FVS) of a dbh-class range by individual species, or species groups created by the user. This is the ideal way to implement a prescription that involves a target diameter structure. If you want to cut remove 40% of the large hemlock sawtimber 16-24" dbh, then you can do so easily. You can also cut diameter ranges to fixed densities, either basal area or trees per acre. If you specify both a cutting efficiency and a residual BA, then the cutting efficiency is ignored.

   If you have "standard" prescriptions you want to use often in multiple scenarios, you can save them as Keyword Component Files (*.kcp) using the "Write" button. To bring them into a simulation, use the "Insert from File" button, and navigate to the kcp file(s).

2. "Thin individually marked trees" [keyword: **ThinPRSC**]. This is very handy if your data are from a stand that has already been marked for a treatment, or if you mentally do a "cut/leave" tally while collecting the plot data. Here, you use the "Prescription" column of the TreeList table, where up to 9 different codes can be used.

   This keyword also provides a way to harvest by tree criteria which are in the tree list, but FVS provides no direct way to harvest according to them. A good example here is the Tree Value Class (TVC). Say you code all the unacceptable growing stock as TVC = 2 in the tree list file, and you want to cut them all. There is no such keyword. In Excel, you could use a simple "IF" formula in the Prescription column to set a unique code if the TVC =2, then use this code in a Prescription Thinning. [This would only work if you used Linked Excel worksheets for your input.]

FVS has recently added a new keyword **"SpLeave**" which allows the user to prevent any species or species group from being harvested. This must precede the actual prescription keyword. The ability to keep a species from being harvested is very useful, primarily in cases where you want to treat most of the stand similarly, but keep all rare species regardless of quality, such as an overstory removal with reserves in the final step of a shelterwood sequence.

Another very useful feature applicable to all keywords is the ability to create species groups [keyword: **SpGroup**]. For example, if you create an "Aspen" group, and assign quaking and bigtooth aspen to it, you can then make prescriptions or direct output by using this group. Once you create groups, they will then show up in the menus of all the various prescriptions options that include species as a variable – very handy!

If you have more than one stand in a simulation, Suppose will apply any keywords, including management actions, to ALL stands unless you direct it to do otherwise. ***To apply a treatment to a specific stand, first right-click on it in the Simulation File window, then select the keyword(s) and enter its parameters. When you right click to select a stand in this fashion, its name will turn RED in the Edit Simulation Screen, indicating that any keywords now apply just to it. Right click again to de-select.***

If you're experimenting with various silvicultural treatments, then consider making only a 2-year simulation with a 1-year output interval. This way, the FVSStand tables produced will show the actual trees and volumes harvested, and the post-harvest tables will match the residual stand. Once you get the prescription exactly the way you want it, then increase the simulation length and output intervals to longer periods. If you use 5- or 10-year output, then all the harvest data (in FVSStand) will be annual averages and you'll need to multiply them by the length of the period to get the correct totals. Also, the post-harvest stand will include some growth on the residual trees, so you'll never know exactly what the residual BA or TPA was, for example.

Add Keywords

FVS is an extraordinarily flexible program that can be customized extensively through the use of Keywords. The Keyword Manual is nearly 100 pages long and contains seven categories of keywords:

- Program Control
- Stand and Tree Information
- Growth and Mortality Modifiers
- Thinning
- Volume Controls
- Input/Output Control
- Extensions

The "Add Keywords" box uses 9 similar categories to help narrow down the number of options. (There is some overlap among these categories, and with the Management Actions box on the main menu.) Clicking on a keyword in the right column gives a short description of its purpose, but you must read the more detailed description in the manual to understand fully how to apply it. Often, the input screen for the specific keyword is self-explanatory.

A few commonly used keywords (in addition to the post-processors below) include:

NoTriple – suppresses the automatic tripling of tree output records (something the program does automatically if the tree list is small, to avoid certain problems with random processes.) Invoke this to keep your tree list the same size as it is at the beginning.

NoAutoES – suppresses automatic regeneration establishment. Currently in the NE variant, this includes only sprouting of certain hardwoods. If small-dbh hardwoods mysteriously show up in the future tree lists after cutting, but you don't want them to appear, then invoke this keyword to suppress them.

BFVolume and Volume – redefines the merchantability limits for total and board-foot volume tables. This can be valuable for softwood species which tend to be used for sawlogs at much smaller diameters in Maine than in many other places.

TreeList and CutList – produces an output file with the entire tree list (living trees or harvested trees) at the date(s) specified. Necessary for using SVS, Stand and Stock tables, and other custom post-processing.

Select Post Processors - FVSStand

FVSStand produces stand and stock tables, growth, harvest, and mortality summaries by species for any future date. This is the main program by which you learn what happened to the original stand and how any silvicultural prescriptions were implemented. The FVSStand program runs automatically if you activate it here prior to the simulation, or it can be run as a stand-alone program after the FVS simulation is completed.

To run FVSStand as part of an FVS Simulation you must both add it as a postprocessor and add its keyword, as follows:

1. Click on the Select Post Processors box.

2. Scroll down the “Available post processors” window until you see “FVSSTAND Alone: Generate Dynamic Yield Reports”.

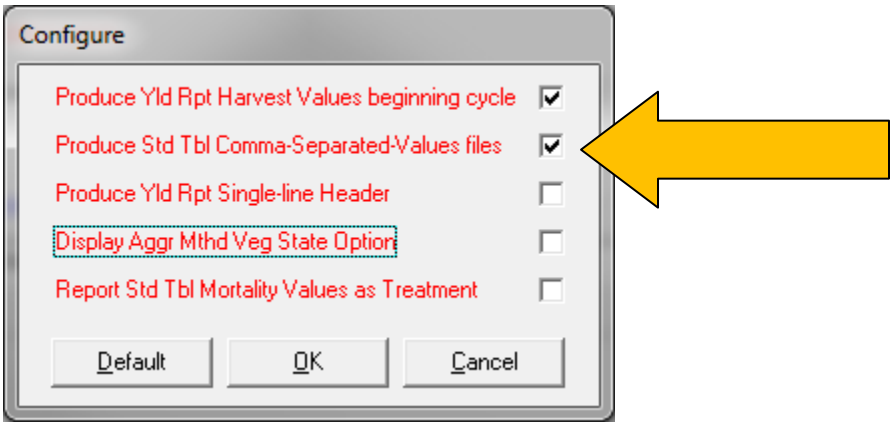
3. Double-click this line and it should now be listed in the middle window under “included post processors:” Click Close to return to the main screen.

4. Click on the Use FVS Keywords box.

5. In the far right column under “keyword”, scroll down until you see FVSStand. Double-click.

6. If you want output only for a specific year, then enter this year in the box. If you want output at *all* future years (the usual option), enter a zero in the year box (as instructed in the first line of the screen). Click OK to accept.

7. If you want to save the output tables as *.csv files (very useful in Excel), then when the program first executes, click the “File” command, then “Preferences, Configure” and check the second box as show below. This will remain in effect in future applications unless you change it back.

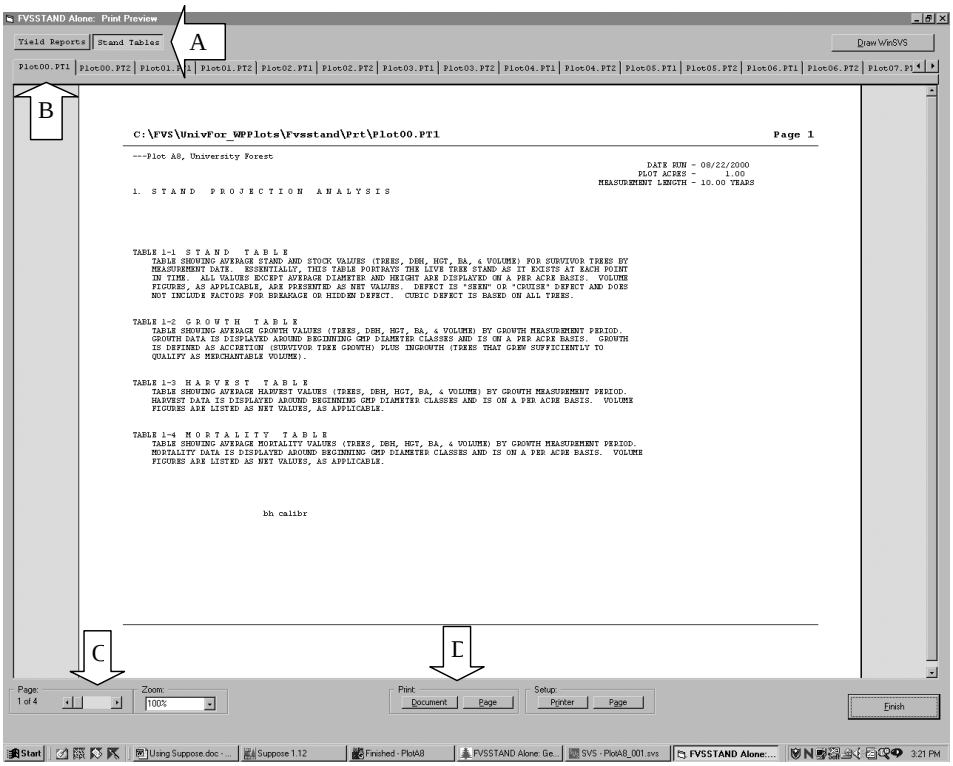


FVSStand creates a subordinate subfolder “Fvsstand” under the folder that contains the keyword file set used to run the simulation. Within the FVSStand folder, three sub folders are created. For stand and stock data, we want the **PRT** folder. Here we have Stand and Stock Tables and growth summaries which are editable and can be copied to a spreadsheet or other application. If you chose to generate these data in *.CSV spreadsheet-input files, then these will also be created in the PRT folder as well.

There are two types of “Stand Tables”, labeled with file extensions .PT1 and PT2. Notice that these repeat *in pairs* on the line across the top of the page. Each pair represents a year of simulation output, beginning with the initial conditions (the “00” labels) at the far left. Because the leftmost tab on the second row is highlighted (arrow B), the view above shows the contents of the 4-part, *.PT1 tables:

- Stand and stock tables (2” dbh classes)
- Growth
- Harvest (may be zero or missing if no cutting is done)
- Mortality.

Output screen from FVSSTAND alone.



There will be one table for each species present if you checked this option on the FVSStand Wizard (the default).

To scroll through all the tables, use the slider bar at the lower left corner of the screen (arrow C).

You can print any or all tables using the print buttons at the bottom middle (arrow D).

The *.PT2 tables are organized similarly, but information is grouped by broader dbh classes and many species can appear on a single page. In other words, the PT2 output is more summarized than the PT1 tables, which use 2” dbh classes.

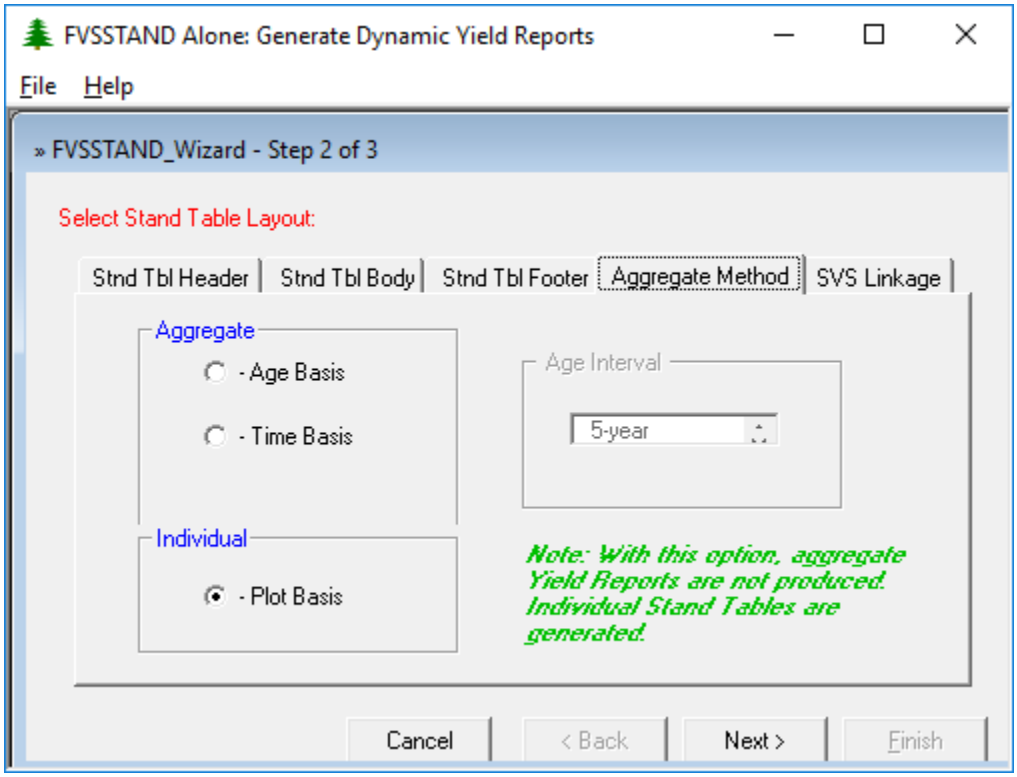
When you’re done with this FVSStand table generator, close it by clicking the upper right.

Don’t worry about closing this window prematurely; you can always get this information back. These tables are stored in editable formats in the FVSStand subfolder that is now attached to the subfolder where your data files are stored. (FVSStand creates this subfolder automatically.) As described above under the FVSStand post-processor, the PRT subfolder of FVSStand now contains these stand and stock tables, labeled just as they are in the original view. If you wanted to make a graph of the stand table in Excel, for example, this is where you would go to copy the relevant data.

Getting Data for Individual Stands

The default for FVSStand is to combine all data into a single set of averages. If you have a woodlot with multiple stands, this is probably not what you want. To get individual stand data, you must click on the “Aggredate Method” tab in Step 2 of 3, then check the “Invdividual Plot Basis” radio button when

FVSStand executes. This will produce separate summaries for each stand which are stored in the FVSStand Folder, but no printed tables will be displayed on the screen. Screenshot below:



FVSStand CSV Files

Through much experience, we have found that the best way to manage output is to create the CSV files, then extract what you need using Excel's versatility. [By all means, look at the screen and print output as the program executes to verify accuracy, but you don't need to write any of this down or copy it from there.] Each output interval (typically a decade) will be in a separate file, numbered ****00, ****01, ****02, etc, corresponding to the intervals used. **** is the first 4 letters of your keyword file, so named when you run a simulation. Obviously if you're doing multiple scenarios that you must keep separate, don't use the same first four letters for the different key files or else they'll look the same in the PRT folder.

Having all these separate files is awkward if you want all your output in one excel file. I could not find any simple command using Excel or Access to append all these CSV files into a single one, but there is a very clever, if ancient, way to do this quite simply:

1. Open the old DOS Command Prompt. (You'll see the old C:\ prompt on a black screen. Very cool. This still works in Windows 10.)

2. Invoke a series of "cd" commands (Change Directory) until you're in the folder where the CSV files are stored. Typically this would be the folder: C:\FVSData\FVSStand\Prt.

3. Type "Copy *.csv AllYears.csv and the system will copy all the CSV files in that folder to a new file named AllYears.csv. Amazing what old technology will do!

Now you can open the combined file with Excel, turn the entire contents into a Pivot Table, and quickly create whatever custom cross-tabulations you might need for reports or graphing.

For every output interval, FVSStand will create four kinds of data, labeled in the "Component" column of the CSV file: Live_Tree, Grow_tree, Mort_tree, and Harv_tree.

Very Important: The values in the Grow_Tree rows, or those in the Growth tables of the printed output, are ACCRETION only. Do not use these values if you need the Net Growth (PAI) of the stand! Accretion is just the positive increase in tree BA and volume owing to diameter and height growth. To get net growth of the stand, we must subtract the Mortality (the rows labeled Mort_Tree). As far as I can tell, FVS does not do this anywhere, except perhaps in the TOSS tables.

Also Very Important: Harvest values in FVSStand are not totals, but annual averages for the growth period, generally 10 years unless you change it. I am not sure why the developers chose to do this (perhaps to equate with growth?), but it means you must multiply the Harv_Tree data by 10 (or other length of the growth period) to derive the volume cut. The Stand and Stock Tables Post-processor reports the total harvest, and requires no further conversion.

If you get a program execution error while FVSStand is running, it may be because you have another window of FVSStand open from a prior run. If this is not the problem, just delete the entire FVSStand folder and re-run the simulation.

PostProcessors - TOSS

Everything that happens during a FVS run is written more or less sequentially to the Output File, which has the *.out suffix. This file will appear in your FVSData directory after a run. It is an old-fashioned text file that one must manually scroll through to find a particular piece of output info. It is a good habit to routinely inspect these *.out files, especially if you're having strange results from a simulation, because the contents will document everything that happens during the simulation.

TOSS simplifies this process. Invoke TOSS as the postprocessor, and you can select whatever part of the Output file you need in a separate report that opens as a text file on the screen during a run. (Unlike FVSStand, you don't need any keywords.) By default you get the general stand summary that appears in the black DOS window. Also, this is where you can print out and edit a Carbon report from the Fires and Fuels Extension, diameter growth calibration factors (if you have repeat measurements), statistics from the cruise if your tree list includes multiple plots, and many other specific items of interest to modelers.

PostProcessors – Stand and Stock Tables

This is a useful postprocessor if you want nice ready-made tables of stand and stock data. Data are organized by species within a given output interval, and totals are also provided. Output is also saved as text files which are automatically stored in your data folder and can be loaded into MS Word and reprinted (after some formatting to fit on the page).

To run the Stand and Stock postprocessor, you need to add the TreeList and CutList keywords, because this program uses these output files to do its computations. Make sure to create lists for all output years when invoking these keywords.

Output Tree and Cut Lists

If you want to track how your trees are growing over time individually, or see the exact dimensions of trees that are harvested, then use these keywords. You will get simple text files similar (though not identical) to the input tree list that can be easily brought into Access or Excel using the Text-to-Columns function if necessary.

If you're working with a fairly small tree list but still want to track individual trees, then make sure to invoke the NoTriples keyword which prevents FVS from creating an expanded tree list.

Select Post Processors – Stand Visualization System

SVS, the Stand Visualization System, allows you to display the tree list in various one or three-dimensional views. This can be very helpful in visualizing stand development over time, or displaying the visual consequences of harvesting and silvicultural treatments. It contains an interactive tree-marking window with a little paint-gun too, and even has a tree designer to create your own custom tree forms!

SVS will run as an integral part of Suppose, or as a stand-alone application. If you run it as part of Suppose, it can greatly slow down the program execution. If all you want is some tabular output, or if you're still experimenting with a variety of silvicultural treatments where visuals don't help, don't bother with it. Plus, under the automatic option, there is no way to change certain SVS parameters such as the plot size (default is 1.0 acre) or the spatial patterns, both of which can be done with the stand-alone program.

Creating SVS Output automatically with Suppose: If you want to run SVS at the same time as FVS through Suppose, then follow the steps above (for FVSStand) for invoking post-processors. Just select SVS in the Post processor window, and then turn on its Keyword in the Use FVS Keywords box. Files will be created automatically and stored as described below.

Creating SVS Output manually from output Tree Lists: *[Note: I learned in 2014 that the Forest Service has stopped supporting the conversion program, although it still works fine if you know what to do.]*

A more flexible way to run SVS, once you've settled on a particular scenario that you'd like to illustrate, is as a stand-alone application. SVS will convert FVS output tree lists to its own format, and in the process, you can change the plot size, views, and other things that can't be done via Suppose. If you do plan to use SVS as a supplementary tool, you must tell FVS to produce output tree lists for every time in the future that you want SVS to "visualize." To do this, use the TreeList Keyword, as follows:

1. In Suppose, click on the Use FVS Keywords box.

2. Scroll down and double-click on TreeList.

3. Type the year that you want the TreeList in the Select Year box. If you want all future years, just enter zero in the "year" box.

4. You can leave all the other options with the default values unless you need to change them for some specific reason.

5. If you've implemented any harvesting treatments, you should also generate a CUTLIST (list of cut trees) for the years when treatments are implemented. This will include the harvested trees in the overall output TreeList, and allows SVS to display them as stumps or horizontal stems. Follow the procedure for adding the TreeList Keyword, only select CutList from the keyword list.

Once you've finished a simulation, you can then open SVS and convert this output tree list to SVS format. Follow these steps:

1. On the SVS Main Menu bar, click File, New Stand.

2. Click the fourth box (Read FVS tree list file). The program will probably open your main FVSbin folder; navigate to where your *.trl files are stored, and select the one to convert. These are the output tree lists generated by the TreeList Keyword.

3. Click the Convert file button in the lower left of the "Convert an FVS tree list file…" box.

4. Now, you'll see a box headed "FVS2SVS options". The default plot size and spatial distribution of trees can be changed under the spatial pattern parameters options; there are many other possibilities here.

5. Click OK to complete the conversion. SVS should now run and generate the first view on the screen.

SVS Output Files. When you run SVS *automatically* as part of an FVS simulation, it will create tree list images for every output interval (without specifying the TreeList keyword). Each of these will be a separate file, with extension .SVS. During the run, FVS creates a new subfolder that has the same name as the simulation file, and stores the individual image files there. This subfolder is attached to the original parent folder where the TreeList, Simulation, and other files are stored. In this original parent folder, you will now find a file named "*name*_index.svs" where *name* is the name of the simulation (.key) file. If you open this *index* file with SVS, it will automatically index all the images for a particular simulation and allow you to display them in any order.

Files are handled and stored differently when you convert an FVS tree list to SVS *manually* using the Tree List Conversion option in SVS. Here, the individual image files will have extensions of ".001, .002," etc, corresponding to each output interval. The overall index file used to access these images has the same name as the output tree list (.trl file) used to generate it, but with the .SVS file extension. (The letters "*index*" are not part of the file name, as when FVS creates the file.) All these files will be in the same subfolder that contains the original FVS output tree list. These .SVS files are not really images, but text files that contain the tree data necessary to create them. (You can see their format and content if you open a .SVS file with NotePad.)

If you want to save any image created by SVS, for example for use in a PowerPoint presentation, just click on File, Save Image As… and select the desired options (save to file, Windows bitmap, etc.) In effect, this permanently converts the tree data to a picture that no longer requires SVS to portray.

SVS Tree Designer. If you attempt to create your own trees designs (the last option under "SVS Options" on the main SVS menu), you must be sure to edit the correct Tree Design file (*.trf). The Forest Service now supports only an "Eastern.trf" and "Western.trf" which are stored in C:]FVSbin. For some reason, when I convert *.trl FVS tree list files (in 2014), the SVS text file refers to the "NE.trf" designer file which has not been supported for many years. To overcome this, just copy the Eastern file and rename it NE.trf if you need to. Load the desired *trf file (bottom left), then find the alpha code for the species you're changing on the drop-down menu. Important: set the tree class to 99, which is the wild card; others are for shorter heights and may not be selected. Edit the crown parameters and you'll have some interesting trees in your visualizations!

Edit and Save the Simulation File

This feature is a handy way to check exactly what options are selected in a particular simulation, and to delete options that you don't want. This screen is confusing, but it does show the stands in the simulation and the keyword options that have been selected. Before making a run, it's always a good idea to look here first, and delete any stands, options or keywords that you don't want. If you're not sure what a particular line in the window refers to, highlight it and click on Edit Selection at the bottom left of this screen. This will take you to the input menu for that item.

This information is all stored in the Simulation File (*.key). This file can be saved anytime under the File, Save or Save As Menu, or just before running the simulation when prompted by Suppose. ***All output files are written to the same folder that the Simulation File is stored in***, so make sure to specify the same subfolder as your input files (tree list, etc.) when prompted by Suppose. Also, you should always specify a new Simulation File name if you make changes and want to save the output files from previous runs; otherwise, FVS will overwrite them.

The Sequence of Actions while a Simulation Executes

To run a simulation, click the Run Simulation box. Depending on what options you've selected, several things will then happen:

1. The FVS growth simulator will run in a MS-DOS Window. No input is required; just close the DOS window when the simulation is done. If you want to capture this in a file, add the TOSS postprocessor.

2. If specified, the SVS program will start, and begin to "grow" trees in a window at the upper left of the screen. No input is required; just allow this to finish. You can come back later and use the Next/Previous buttons to see how the stand will look in the future.

3. If specified, the FVSStand Wizard will open in yet another window, and will ask you to specify options for the output tables. Here, type in a descriptive title, select whether you want output for individual species, etc. Just look at each tab and respond accordingly. Click Next and do the same for the Yield table output. (Just leave these boxes blank if you don't plan to use these yield tables later.)

4. FVSStand will then execute, generating tables for each output interval. When this is done, close this window by clicking the X in the upper right.

5. You should now see a screen that allows you to view and print the output tables generated by FVSStand.

6. If you're using the Database Extension, you output some standard files to your input database, or another output database, using the DSOut keyword. These options are covered in detail in the Database User's Manual.

How FVS Labels and Calculates Tree Volume Units

In FVSStand's CSV tables, there are four volume headings, defined as follows:

ALL_CU/Ac: this is the stemwood cubic volume (ft3) of the MERCHANTABLE STEM only, trees 5" dbh class and bigger, usually to a 4-inch top (these can be changed with the volume keywords). It does NOT include

the stump or tip of the tree smaller than the merchantable top size, or any stemwood volumes of saplings (1-4" trees).

SAW_CU/Ac: this is the cubic volume (ft3) of the sawlog portion of any merchantable tree, as defined in board-foot volume definitions. Normally we would not need this for anything; however, it is used to calculate the topwood volumes (next).

TOP_CU/Ac: this is the cubic volume of the upper portions of sawlog-sized trees, from the upper sawlog top dib to the 4" merchantable upper limit. This is what you would use to calculate pulpwood volumes resulting from harvesting sawlog trees. [Note: this is not reported in the PT1 or PT2 tables, nor the screen output, when FVS stand executes, so this is one example of where using the CSV files comes in very handy.]

BD/Ac: this is the gross (no defect deduction) board foot volume of sawlog-sized trees, using assumed taper built into the standard equations. For the NE variant, hardwoods must be 11" dbh, to a 10" top dib; softwood board footages are calculated for trees 9" dbh and up, to a 6" top dib. If these are not accurate, they can be changed with the BFVolume keyword, a very flexible tool to accommodate specialty markets. For example, you can tell FVS to calculate board foot volumes for paper birch boltwood trees 8" dbh and larger, to a 7" top, if those are the mill specs in effect.

Important: regardless of how you categorize trees in the TreeValue column of the tree list, FVS will compute board foot volumes for them if they are large enough. This is true in the Stand and Stock and FVSStand Postprocessors, and probably others. If you have many non-growing stock TreeValue code 2 (UGS) or 3 (cull) trees, the summary board-foot volumes will be gross overestimates of the true board-foot volumes. This can be overcome, but requires customized Compute Statements using the SpMcDBH keyword from the Event Monitor (described below).

The Stats Keyword

If you want to see the accuracy of your cruise data, invoke the **Stats** keyword. The only variable is the significance level for calculating confidence limits; the default is p = 0.05. You can view this table in the *.out file, or invoke the TOSS postprocessor and check the box Stats: Stand Attributes when TOSS runs. You will see means, standard deviations, confidence limits, and percent accuracy for basal area, trees per acre, cubic volume, and board-foot volume.

Keyword Component Files (*.kcp)

If you make frequent use of certain keyword combinations (as in a complex harvest prescription, or elaborate Compute statements), you can save these from the Suppose menu to a simple text file, then import it into another simulation where you require the same actions or output. Just click on the keyword(s) to save, click on the "Write" button at the bottom, and it will be saved to your data directory. To bring it into a simulation, just click the "Insert from File" button on the second row at the top.

Customizing the Names/Titles of Keywords

This is an optional, but very handy step that really helps keep elaborate simulations organized and understandable. For any keyword, you can just type in a descriptive title in the "Name" box at the top,

and it will appear this way in the Simulation file window. For example, you could change: "Base FVS system: BFVolume" to "Compute Boltwood Volumes for 8-10" white birch trees", or "Base FVS system: ThinDBH" to "Cut all red maple between 8 and 14 inches dbh". Major improvement!

The SPMCDBH Function (Event Monitor Function)

FVS provides the ability to define your own variables using "Compute" statements that employ many different functions in the so-called Event Monitor (described in Crookson, GTR Int-275, 1990). I have found the SpMcDBH function to be of tremendous value in calculating various stand metrics that I would otherwise need to derive from tree lists or other output.

Here is the description of this keyword, directly from its Users' Guide:

SPMCDBH Returns the trees, basal area, total cubic volume, merchantable cubic foot volume, total board foot volume, quadratic mean diameter, average height, percent cover, average mistletoe rating, stand density index, or average diameter growth per acre for a given species, tree- value class, tree status, and range of diameter and height. The first three arguments are required.

Argument 1 = code indicating which measurement is desired, as follows:

1 = trees per acre,
2 = basal area per acre,
3 = total cubic volume/acre, or
4 = total board foot volume/acre
5 = quadratic mean diameter
6 = average height
7 = percent cover
8 = average dwarf mistletoe rating
9 = merchantable cubic foot volume/acre
10 = average diameter growth for the cycle
11 = stand density index

Argument 2 = alpha or numeric species code where 0 (or All) indicates that the sum over all species is desired. Numeric species codes are variant specific.

Argument 3 = tree-value class (IMC codes 1,2, or 3; see Wykoff and others 1982, p. 16) where 0 indicates that the sum over all tree-value classes is desired.

Arguments 4 and 5 = diameter range for trees included in the sum. The default values are 0.0 and 999.0.

Arguments 6 and 7 = height range for trees included in the sum. The default values are 0.0 and 999.0.

Argument 8 = tree status for trees included in the sum, where 0 indicates live trees, 1 indicates recent mortality trees, and 2 indicates harvested trees. The default value is 0.

Examples: SPMCDBH(1,LP,0) returns the trees per acre that are lodgepole pine; SPMCDBH(2,0,3) returns the basal area/acre that is in IMC class 3; SPMCDBH(3,3,1,10) returns the total cubic volume/ acre that is Douglas-fir, IMC class 1, and in trees greater than 10 inches d.b.h.; SPMCDBH (3,3,1,10,20)

is just like the previous example except that the trees must be greater than 10 inches and less than 20 inches d.b.h.; and SPMCDBH(5,0,0,10,999,40,999,1) returns the quadratic mean diameter of recent mortality trees which are at least 10 inches d.b.h. and at least 40 feet tall.

I use this function to make sure FVS calculates board-foot volumes only for those trees which have been tallied as Tree Value Class = 1 (growing stock, or AGS) in the tree list. First add the "Compute Keyword" (either as a database extension or base system option), then define whatever variables you choose. The example below shows a Compute statement that calculates board-foot volumes (first argument = 4) for various species. The first would return the total board-foot volume of white ash sawlogs, for tree value class = 1 only, with dbh = 10 inches or larger. Note the red spruce function that drops the minimum dbh to 8". (You would also need to add this to the BFVolume keyword.) The white pine function adds all the volume of TVC = 1 trees plus half the volume of TVC = 2 trees, on the grounds that most poorer-quality pines will still yield some sawlog material even though weevilled when young and somewhat deformed.

```
Compute              0
WALogs = SpMcDBH(4,WA,1,10)
YBLogs = SpMcDBH(4,YB,1,10)
RMLogs = SpMcDBH(4,RM,1,10)
SMLogs = SpMcDBH(4,SM,1,10)
BTLogs = SpMcDBH(4,BT,1,10)
RSLogs = SpMcDBH(4,RS,1,8)
BFLogs = SpMcDBH(4,BF,1,8)
WPLogs = SpMcDBH(4,WP,1,10)+ (0.50)* SpMcDBH(4,WP,2,10)
End
```

The zero in the first line tells FVS to calculate these variables for all output years (typical). Don't forget the End statement.

These variables are output in the *.out files and also to an output database or spreadsheet if you have designated one. They are formatted as one long line of data, first with the simulation year, followed by all the variables in the order you define them.

Give these Compute statements a descriptive name, and save them as KCP files so you can use them in any simulation without having to re-create them.

Summary of Default File Types and Their Extensions

For reference, here is a list of all files types used by or created by FVS with their default file extensions:

.FVS	FVS Input Tree List (pre-Access)
.SLF	FVS Stand List File (pre-Access)
.LOC	FVS Locations File
.KEY	FVS Simulation (Keyword) File
.TRL	FVS Output Tree List
.OUT	FVS Output File (Detailed summary of EVERYTHING!)
.PT1	FVSStand Output1
.PT2	FVSStand Output2
.SVS	SVS Tree List (open with SVS)

Appendix A: The Tree Data Editor Help File

The FVS Tree Data can be viewed, edited, or both using the Edit FVS Tree Data File window.

The contents of the tree data file are related to a stand by placing the name of the tree data file in Field 3 of Type A stand list file records. A stand list file can be edited using the Edit Stand List File window.

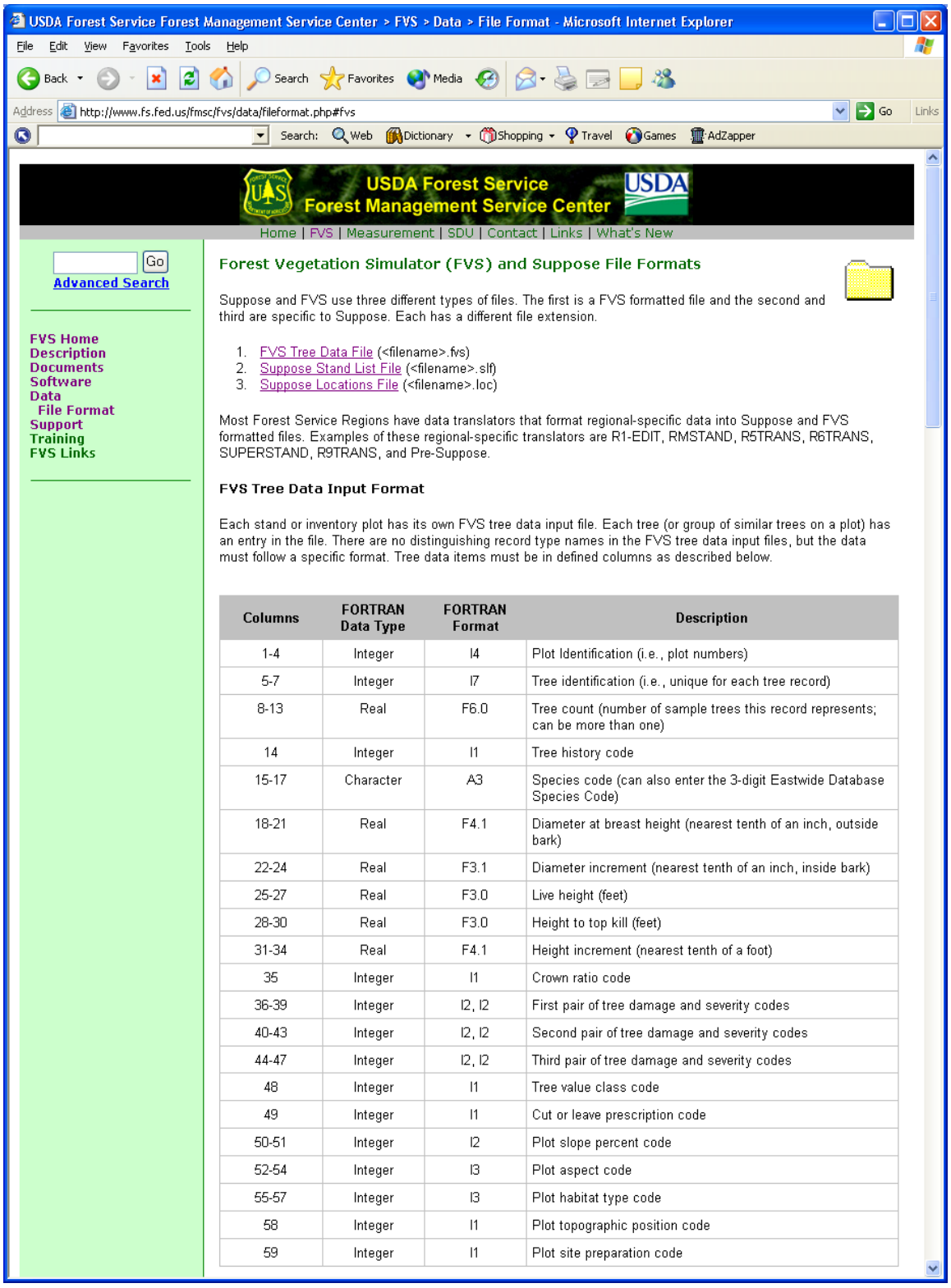
Here is the format of the tree data records (col = columns):

Plot ID (col 1-4):	A 4-digit plot identification; no decimal allowed.
Tree ID (col 5-7):	A 2-digit tree identification; no decimal allowed. FVS reads the plot and tree ID fields together as the tree ID.
Tree count (col 8-13):	A 6-digit real number is the number of trees represented by this data record (also known as a tally); decimal is allowed.
Tree history (col 14):	A 1-digit tree history code as follows: 0, 1, 2, 3, 4, or 5 are live trees. Code 6 or 7 trees died during mortality observation period; code 8 or 9 died prior to mortality observation period.
Species (col 15-17):	A 3-character, left justified, species identification code. The codes are different for each FVS variant.
DBH (col 18-21):	A 4-digit real number is the DBH in inches (metric variants code this value in cm). If NO decimal is coded, the value is interpreted as being in tenths of inches.
DBH Growth (col 22-24):	A 3-digit real number is the DBH in inches (metric variants code this value in cm). If NO decimal is coded, the value is interpreted as being in tenths of inches (or cm).
Height (col 25-27):	A 3-digit real number is the tree height in feet (meters).
Height to topkill (col 28-30):	A 3-digit real number is the tree height to the point of top kill in feet (decimeters). This value is only used if there is a tree damage code of 96 or 97 coded for the tree.
Ht Growth (col 31-34):	A 3-digit real number is the height growth in feet (meters). If NO decimal is coded, the value is interpreted as being in tenths of feet (or decimeters).
Crown ratio (col 35):	A 1-digit number is the crown ratio code, as follows: 1=1-10%, 2=11-20%, 3=21-30%, 4=31-40%, 5=41-50%, 6=51-60%, 7=61-70%, 8=71=80%, 9=81-100%.
Damage code 1 (col 36-37):	A 2-digit number is damage code 1 (of 3). See below for a table of damage and severity codes.
Severity 1 (col 38-39):	A 2-digit number is the severity code that corresponds to the damage code.
Damage code 2 (col 40-41):	A 2-digit number is damage code 2 (of 3).
Severity 2 (col 42-43):	A 2-digit number is the severity code that corresponds to the damage code.
Damage code 3 (col 44-45):	A 2-digit number is damage code 3 (of 3).
Severity 3 (col 46-47):	A 2-digit number is the severity code that corresponds to the damage code.

	Damage and severity coding:
	Damage code 96=broken top and 97=dead top (no severity code is used). The point of broken top or top kill is coded in columns 28-30.
	Damage codes 61=Armillaria, 62=Phelinnus, 64=Annosus use these severity codes: 1=there is a tree within 30 feet that has a deteriorating crown or is killed by the disease; 2=the pathogen or a diagnostic symptom is detected; and 3=crown deterioration observed.
	Damage code 2=mountain pine beetle attack uses these severity codes: 1=unsuccessful bole attack (pitchout), 2=strip attack, 3=successful bole attack, 4=topkill.
	Damage code 3=Douglas-fir beetle attack uses these severity codes: 1=unsuccessful bole attack (pitchout), 2=strip attack, 3=successful bole attack, 4=topkill.
	Damage code 30=dwarf mistletoe in some species, 31=mistletoe in lodgepole pine, 32=mistletoe in larch, 33=mistletoe in Douglas fir, and 34=mistletoe in ponderosa pine. Use the Hawksworth rating (0-6) for severity.
Tree value class (col 48):	A 1-digit number is the tree value cod were 1=desirable tree, 2=acceptable tree, and 3=live cull. You can code an 8 to indicate that the plot is non-stockable and all other values are interpreted as 3.
Prescription code (col 49):	A 1-digit prescription code where 1 or 0 is leave and other values mark the tree for removal.
Plot slope (col 50-51):	A 2-digit slope in percent.
Plot aspect (col 52-54):	A 3-digit aspect in degrees.
Plot habitat/PA (col 55-57):	A 3-digit plot habitat code. If plant association is coded, an index value must be used.
Plot topo code (col 58):	A 1-digit code is the plot topographic position, where 1=bottom, 2=lower, 3=mid-slope, 4=upper slope, and 5= ridge top.
Plot site prep (col 59):	A 1-digit site preparation code where 1=none, 2=mechanical, 3=burn, and 4=road cuts, road fills, and other stockable road beds.

Appendix A: Tree List Format for FVS (http://www.fs.fed.us/fmsc/fvs/data/fileformat.php - fvs)

Note: This format is unnecessary if using the Database Extension, but the variables are all the same.



Columns	FORTRAN Data Type	FORTRAN Format	Description
1-4	Integer	I4	Plot Identification (i.e., plot numbers)
5-7	Integer	I7	Tree identification (i.e., unique for each tree record)
8-13	Real	F6.0	Tree count (number of sample trees this record represents; can be more than one)
14	Integer	I1	Tree history code
15-17	Character	A3	Species code (can also enter the 3-digit Eastwide Database Species Code)
18-21	Real	F4.1	Diameter at breast height (nearest tenth of an inch, outside bark)
22-24	Real	F3.1	Diameter increment (nearest tenth of an inch, inside bark)
25-27	Real	F3.0	Live height (feet)
28-30	Real	F3.0	Height to top kill (feet)
31-34	Real	F4.1	Height increment (nearest tenth of a foot)
35	Integer	I1	Crown ratio code
36-39	Integer	I2, I2	First pair of tree damage and severity codes
40-43	Integer	I2, I2	Second pair of tree damage and severity codes
44-47	Integer	I2, I2	Third pair of tree damage and severity codes
48	Integer	I1	Tree value class code
49	Integer	I1	Cut or leave prescription code
50-51	Integer	I2	Plot slope percent code
52-54	Integer	I3	Plot aspect code
55-57	Integer	I3	Plot habitat type code
58	Integer	I1	Plot topographic position code
59	Integer	I1	Plot site preparation code