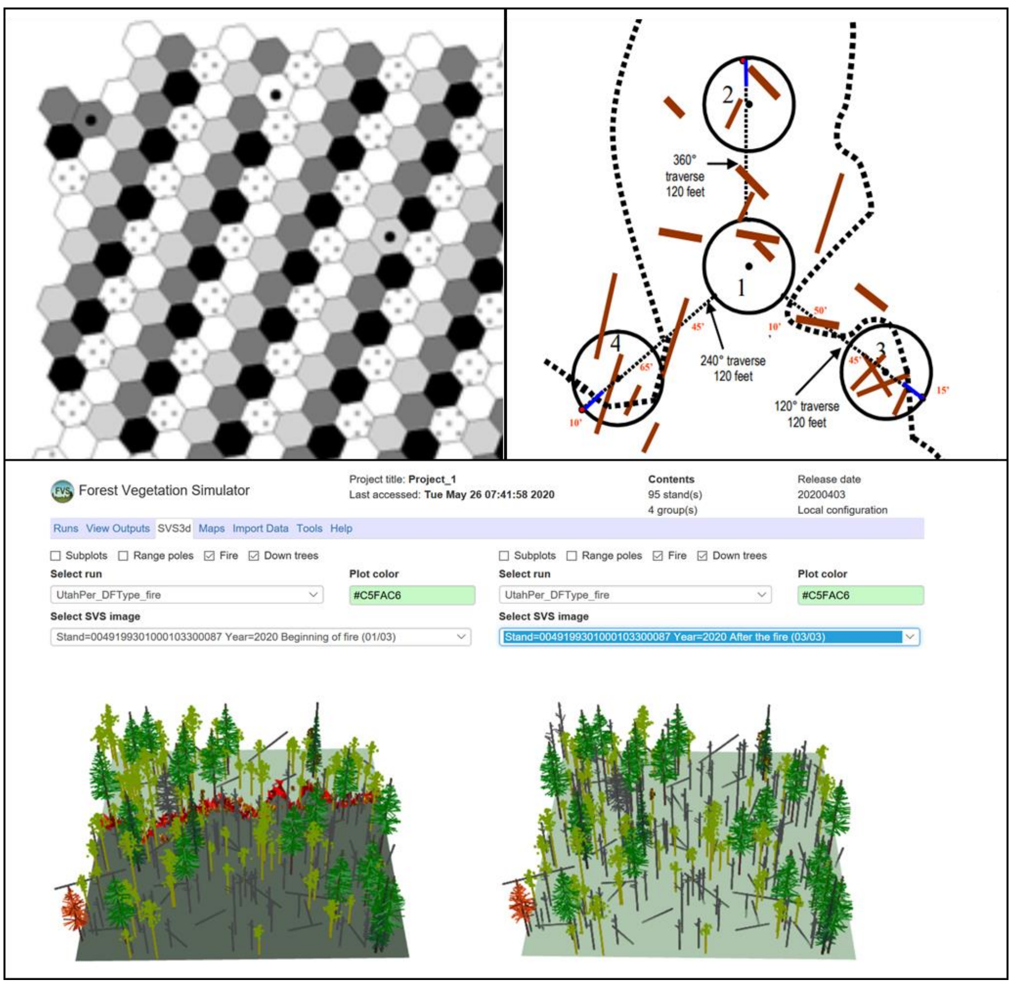
United States Department of Agriculture

Quick-Start Guide to Forest Inventory and Analysis Data in the Forest Vegetation Simulator

John D. Shaw
Aaron Gagnon



Forest Service
Rocky Mountain Research Station

September, 2020

Quick-Start Guide to Forest Inventory and Analysis Data in the Forest Vegetation Simulator

John D. Shaw, USDA Forest Service, Rocky Mountain Research Station (FIA) (john.d.shaw@usda.gov)
Aaron Gagnon, USDA Forest Service, Forest Management Service Center (FVS) (sm.fs.fvs-support@usda.gov)

Background

Shortly after the initiation of annualized inventory by the Forest Inventory and Analysis program, the first FIA plot data were made available through the FIA Mapmaker application in FVS-ready format (Miles 2008). These data were quickly adopted by FIA and FVS users, but they were provided in the original format used by the FVS Suppose interface (Crookston 1997), where the data for each stand was contained in a separate file. Although this was manageable using Suppose, the rapid addition of new FIA plot data over time would make the approach increasingly unwieldy. When the Mapmaker interface was retired, there was no replacement for delivery of FVS-ready FIA plot data.

In response to user demand created by the brief availability through Mapmaker and general user interest in running FIA plot data through FVS, the Forest Management Service Center developed FIA2FVS using specifications for translating FIA variables into FIA variables (Shaw 2009). FIA2FVS was originally an end-user application that could be used to process FIA database tables (FIADB) available on the FIA Datamart (https://apps.fs.usda.gov/fia/datamart/) in CSV file format.

FIA2FVS converted the CSV-formatted data from FIADB into a single Microsoft Access database, which was accessible in Suppose through the addition of the FVS Database extension feature (Crookston et al. 2003). FIA2FVS corrected several issues that were present in the Mapmaker data, but some of the complexities of translating site index from the equations used by FIA to the equations expected by FVS remained. Because FIA2FVS was developed as an end-user application, constant updates to the FIA database required FIA2FVS to be updated frequently. In addition, users attempting to translate data could use many different version combinations of FIADB and FIA2FVS, leading to an increasing support burden.

As FIA moved into a second decade of annualized inventory it became apparent that FIA data at a state level would soon exceed the 2Gb size limit of Access databases. As the FIADB set of tables for the first states approached this limit, Postgres was adopted as a stopgap measure and a possible long-term successor to Access as delivery database, while still providing the CSV option on the FIA Datamart. About the same time, the FVS program had decided to adopt SQLite as the standard database underlying the successor interface to Suppose. Because of the effectively unlimited capacity of SQLite databases (140Tb), the replacement process for FIA2FVS could take advantage of the opportunity to house both FIADB and FVS-ready tables in a single delivery database. Given the multiple advantages, FIA adopted SQLite as the alternative to Access.

In a joint effort between the Rocky Mountain Research Station FIA program and the Forest Management Service Center, plans were made to develop an update to FIA2FVS (here after FIA2FVS 2.0). Several major changes were planned for the update (for details see Shaw and Gagnon 2019; https://www.fs.usda.gov/treesearch/pubs/59245), but the most important were:

- FIA data would be pre-translated within the FIA program, removing the translation burden from the user. This would introduce better data quality controls, reduce the user support burden, and ensure that all users could easily obtain up-to-date data.
- Issues remaining after the last update to FIA2FVS 1.0 would be resolved. Notably, site index equations expected by FVS would be applied to FIA site tree data.
- Data would be presented in multiple formats within one database to cater to different users' needs – FIA plot data could be used in FVS simulations at the FIA condition, whole plot, or subplot levels.

Introduction

This guide is intended to give users who are not yet familiar with the latest translation of FIA data to FVS format an overview of the current database structure and basic instructions for using these databases in FVS. In general, users needing help should use the FIA contact email above for any questions regarding FIA data and how they are converted into FVS-ready tables by FIA2FVS 2.0. Basic variable conversion specifications are also provided in Appendices A and B. For questions specific to running simulations in FVS, users should refer to other documentation and training modules available on the FVS web site (https://www.fs.fed.us/fvs/index.shtml) or use the FVS support contact email above.

Obtaining and Importing FVS-ready Tables

FVS-ready tables containing FIA plot data are currently provided in state-level SQLite databases on the FIA Datamart (https://apps.fs.usda.gov/fia/datamart/). Each SQLite database contains six tables that are used by FVS, as well as a large number of tables that constitute the public FIA database (FIADB). Although FVS does not yet use the FIADB tables directly, the convenience of having all tables in a common database provide users with the capability to query FIADB to conduct analyses that can help in the development of simulations. Examples of this capability, which has not been feasible previously, will be included in a forthcoming user's guide.

State databases can be downloaded by clicking on the state of interest (fig. 1) and saving the ZIP file to a convenient location (e.g., C:\FVS). It is not necessary to unzip this file to load it into FVS, but if there is a need to read or manipulate the database outside of FVS it will need to be unzipped. Using the Import Data tab in FVS, navigate to the location of the zipped or unzipped version of the downloaded database to complete Step 1 of the loading process. During normal processing, the blue candy-striped progress bar should be active, although it may pause at times. If the stripes turn red or there is some other sign of a program hang-up, such as the FVS screen turning grey, there could be a problem with the FVS installation (e.g., out-of-date version) or possibly the file was corrupted during download. Every SQLite database is checked for load ability after it posted to the FIA Datamart, so there should be no errors associated with the database itself. However, some users may experience a failure to read in the database when the file is very large (>2Gb) in unzipped form. In these cases, the zipped file should be imported into FVS and the unzipped version can be used for other purposes.

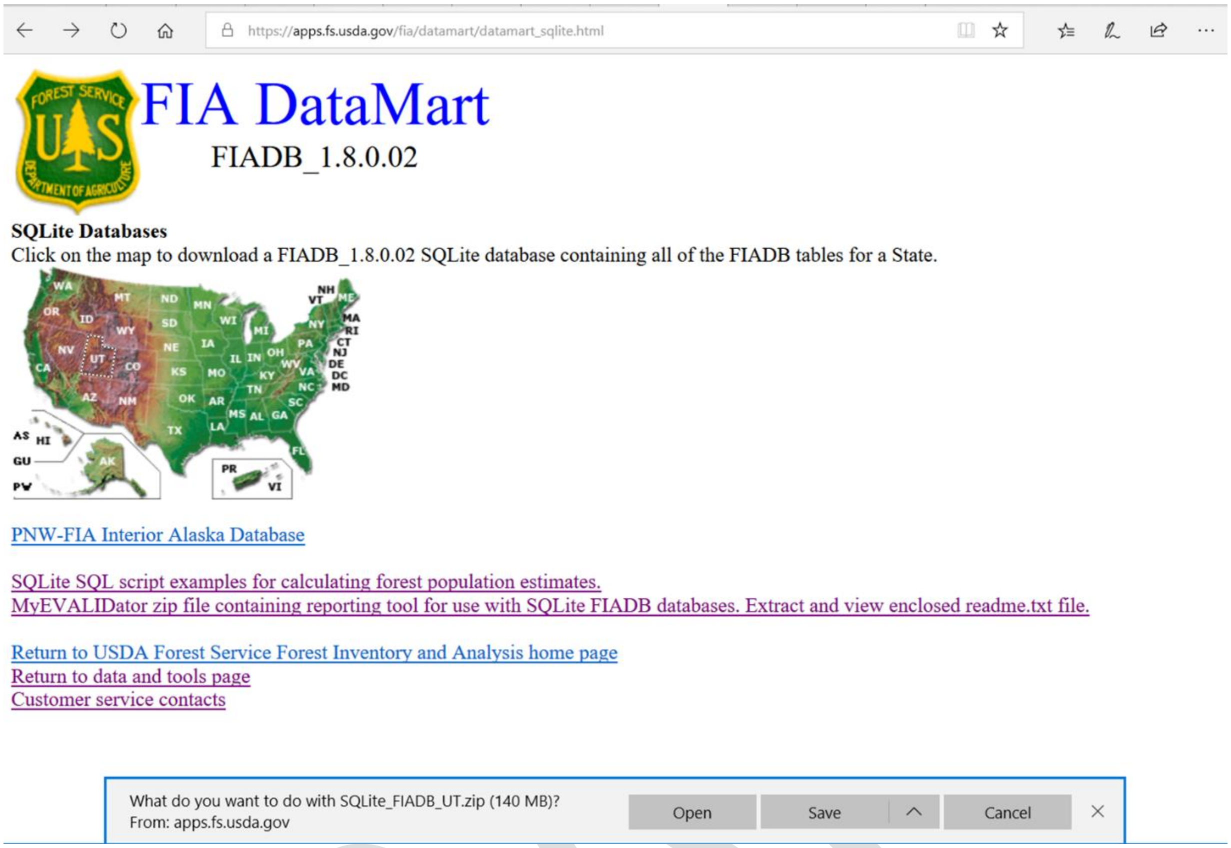


Figure 1. FIA DataMart web page for SQLite databases. State-level FIA databases can be downloaded by clicking in the state of interest.

After Step 1 is successfully completed, the FVS window should look similar to the screen shot in figure 2 (some rows omitted for clarity). Users who are familiar with the standard FVS table scheme will notice the additional tables and somewhat different table names for FVS tables when using FIA data. This is to accommodate the FIA concept of a “condition” which, for practical purposes, is what would ordinarily be considered a “stand” in the field. Although FVS data are also built around the concept of a “stand”, the way ordinary stands and FIA conditions are sampled requires that they be represented separately in the FVS tables.

Therefore, the representations of “typical” FVS tables to equivalent FVS tables with FIA data are:

- FVS_StandInit_Cond: Equivalent to FVS_StandInit, with all data from the same “stand” as identified in the field.
- FVS_StandInit_Plot: No exact equivalent in FVS, since one FIA plot can sample multiple forest conditions, or a combination of forest and nonforest. However, for single-condition plots (i.e., where all FIA subplots sample only one stand), FVS_StandInit_Cond and FVS_StandInit_Plot records will be identical and the stand projections will behave similarly.
- FVS_PlotInit_Plot: No exact equivalent in FVS, but FVS_PlotInit_Plot is to FVS_StandInit_Plot what FVS_PlotInit is to FVS_StandInit in standard FVS tables – i.e., FVS_PlotInit_Plot allows FIA subplots to be run as separate units within a whole FIA plot.

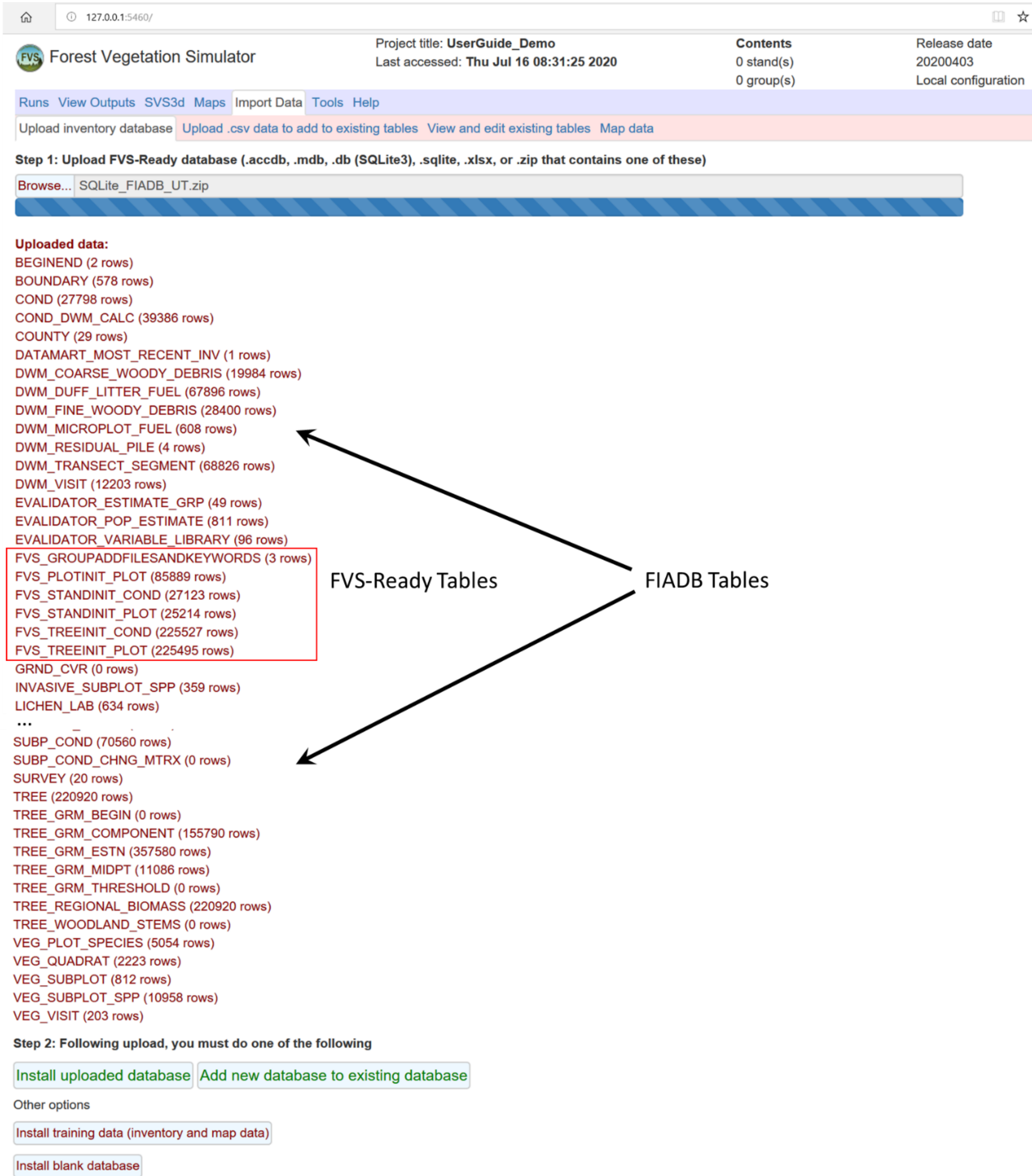


Figure 2. Example results from Step 1 on the “Upload Inventory Database” tab in FVS.

Both FVS_TreeInit_Cond and FVS_TreeInit_Plot are the equivalent to FVS_TreeInit, and are commonly identical in number of records. There are occasionally differences in the numbers of records, owing to the differences in record selections between plots and conditions. The differences are typically small, such as the 32-tree difference seen above. The main difference between the two tables lies in the control numbers (Stand_CN), which link tree records to the appropriate records in FVS_StandInit_Cond and FVS_StandInit_Plot.

If the numbers of rows present in the tables appear correct, proceed with Step 2. The option to "Add a new database existing database" to combine FIA data from two or more states has not yet been thoroughly tested, but conceptually it should pose no problem. The most complicating factor would likely be the inclusion of plots covered by two or more variants when multiple states are loaded (see "Variant Assignment to Plots" section below). For many western states this situation already occurs within a single state, and there are options for dealing with multiple variants when running FVS simulations over multi-variant areas. Combinations of several eastern states may be less likely to result in multiple variants in the data, since many groupings of states can fall under a single variant (e.g., NE, LS, SN). The ability to download multi-state databases directly may be accommodated in future releases of the FIA Datamart, depending on how user demand evolves.

The condition/plot/subplot arrangement is recognized by FVS when the database is installed in step 2, and this can be seen when comparing typical FVS data and FIA data in FVS format under the "Stands" tab in FVS (fig. 3). Standard FVS databases will provide users with two choices (Stands and Plots) in the Inventory Data Tables drop-down under the Stands tab, whereas data translated by FIA2FVS will provide three choices (FIA plots, FIA subplots, and FIA conditions). FIA conditions are most closely matched to stands that are delineated on the ground, but for the majority of FIA plots measured using the current mapped plot design there is no practical difference between plots and conditions – i.e., the FIA plot footprint is located such that all four subplots sample the same condition, and therefore a whole FIA plot and the corresponding FIA condition are coded similarly. See Shaw and Gagnon (2019; https://www.fs.usda.gov/treesearch/pubs/59245) for more information on how FIA conditions, plots, and subplots relate to each other.

Basic Characteristics of the Data

Most variables in the FVS-ready tables have been transferred directly from equivalent variables in FIADB, have been recomputed to have appropriate values depending on whether the FIA plots are single- or multi-condition, or have been cross walked from FIA codes to equivalent FVS codes. Some of these variables, such as the FVS variant and Location codes were populated in FIADB for the primary purpose of used in FIA2FVS translation. This approach simplified translator programming, and provided a side benefit of being able to use certain FVS variables for non-FVS use in the analysis of FIA data. Other variables are composites of multiple FIA variables, such as when Stand_ID is created from state and county codes, inventory year, and plot number. A summary of the translation specifications for all variables in FVS-ready tables are provided in Appendix A (StandInit and PlotInit tables) and Appendix B (TreeInit tables). Information on the source FIA variables can be found in the FIA Databases users' guides (see https://www.fia.fs.fed.us/library/database-documentation/index.php).

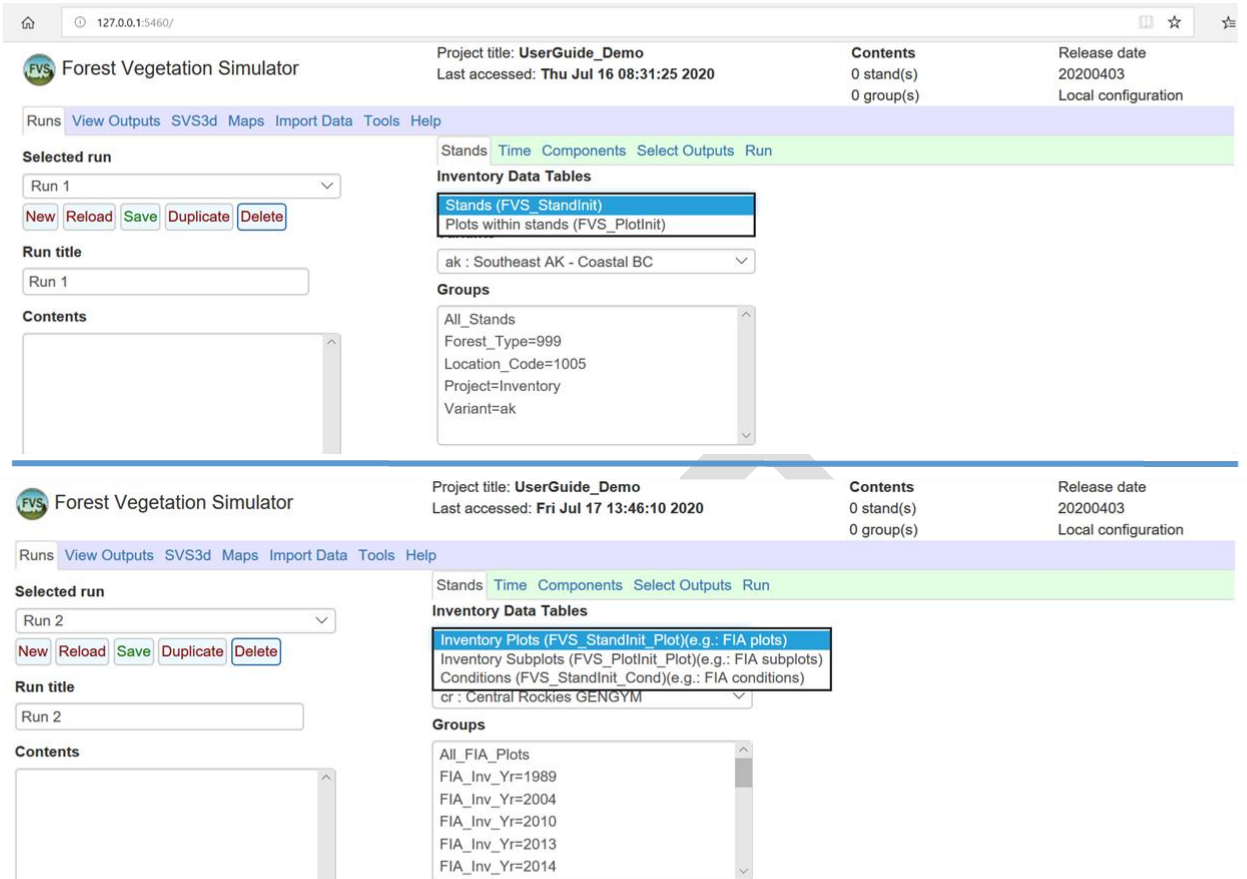


Figure 3. Comparison of the Inventory Data Tables dialog under the FVS Stands tab. Standard FVS databases will present two choices – running whole stands or running plots separately. Databases resulting from FIA2FVS translations will present three choices – whole FIA plots, FIA subplots, or FIA conditions.

Things to Watch Out For

Inventory and Measurement Years

The initial translation of FIA plots into FVS-ready format covered over 1.6 million plot visits, some dating back to the 1960s. This includes data from annual inventory, which started in the late 1990s and was implemented over time, with Wyoming being the last whole state to start in 2011, and Interior Alaska anticipated to be added in late 2020. Inventories done prior to the start of annual inventory are referred to as periodic inventories, and their designs varied widely across time and among states. At the time of development of the modern FIADB, some of these early inventories had properties that didn't allow them to fit cleanly into the new database structure, so not all states will necessarily include all of the same data or table structure as in annual inventories. FIA inventory years of periodic and beginning annual inventory are reproduced from Appendix L in the FIADB User Guide (https://www.fia.fs.fed.us/library/database-documentation/index.php) here in Appendix C.

This is a good place to note the relationships between FIA inventory year (INVYR), measurement year (MEASYEAR), and FVS inventory year (Inv_Year). For periodic inventories, which commonly took several years to complete, FIA inventory year is usually the final year of measurement or the year during which most of the plots were measured. For example, the 1999 inventory of Arizona includes data from plots measured between 1984 and 2000, with nearly 80% of the plots being measured between 1995 and 1998. In periodic inventory, plots with a common inventory year are used together to produce population estimates.

Under annualized inventory, inventory year is assigned to plots in a panel scheduled to be visited in that year, and a full inventory cycle typically includes 5 to 10 panels. For various reasons, such as active fire, early fall snowfall, or temporarily denied access, not all plots from an annual panel are done on schedule. A matrix of inventory year vs. measurement for the most recent full inventory cycle of Arizona shows this “creep” (done ahead of schedule) and “lag” (done behind schedule) (Table 1). Note that in 2015 an unusual number of plots were measured from inventory years 2009-2014. These “catch-up” plots may have been skipped for a variety of reasons during their scheduled years, but circumstances were favorable (landowner permission granted, sufficient budget/staffing, long field season, etc.) during 2015 to allow them to be measured.

Table 1. Inventory years and measurement years for a full annualized inventory cycle (2009-2018) for Arizona. Cells on the diagonal (green) represent numbers of plots measured during their scheduled inventory year.

Inventory Year	Measurement Year											
	2007	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	Total
2009	27	1192	19					3					1241
2010	28		1147	50				3					1228
2011				1191	17		22	5					1235
2012					1212	9	9	20					1250
2013						1127	29	27					1183
2014							1226	12	1				1239
2015								1211	15	8			1234
2016								3	1156	81	1		1241
2017										1225	10	1	1236
2018											1184	3	1187
Total	55	1192	1166	1241	1229	1136	1286	1284	1172	1314	1195	4	12274

FVS inventory year (Inv_Year) is set to FIA measurement by FIA2FVS because FVS expects the actual year of measurement. This allows for stand age, for example, to be accurately represented in simulations. As a result, it is not possible to look at Inv_Year in the FVS StandInit tables and know what set of plots constitute a complete FIA inventory cycle, should it be desired to project a full cycle of plots with no overlap of previous cycles. However, both the FIA inventory and measurement years are included in the Groups column, so all conditions, plots, or subplots can be selected by selecting the “Any” button for “Stands must be in any or all selected groups”, and highlighting all inventory years belonging to a full cycle (fig. 4).

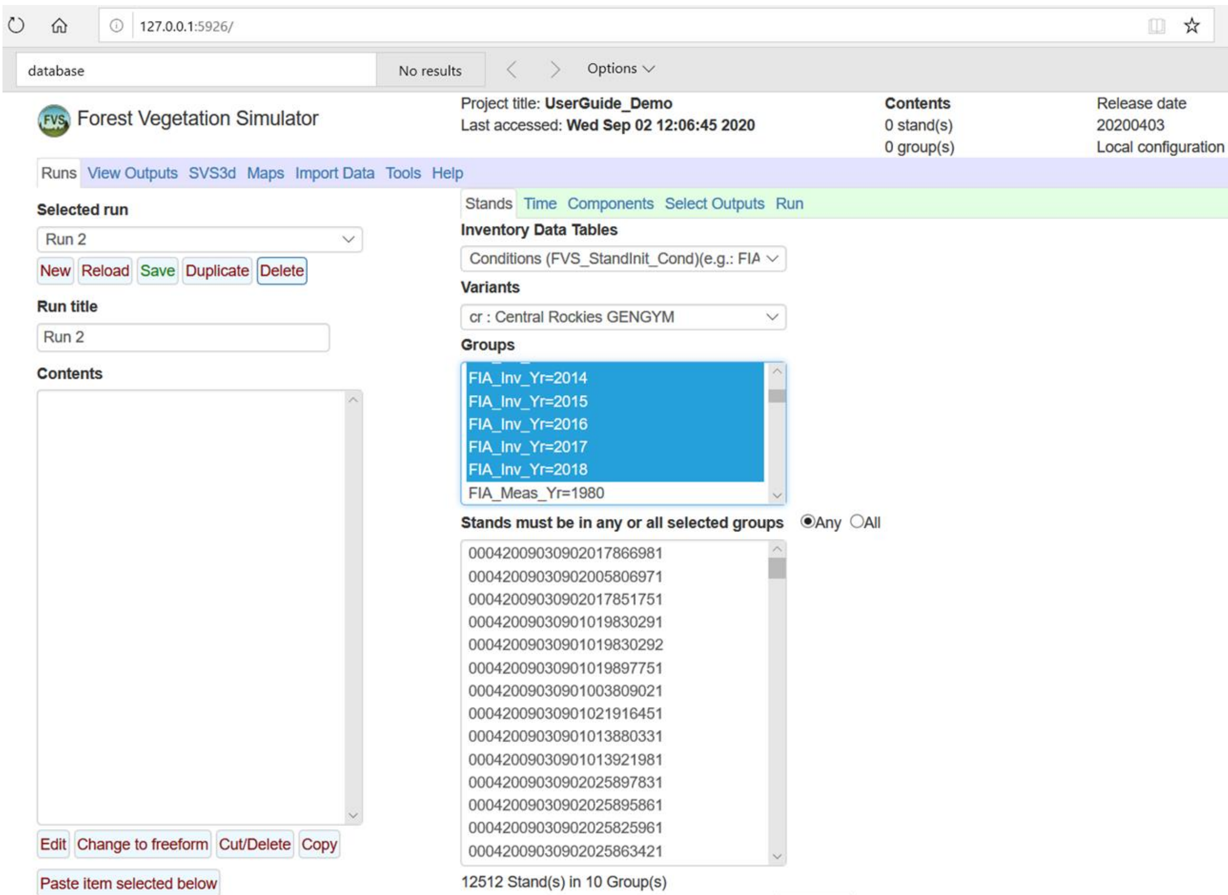


Figure 4. Selecting multiple FIA inventory years to access all conditions in the most recent inventory cycle for Arizona, based in Group membership. Note that the total number of conditions (12,512) exceeds the number of plots (12,274) in table 1, due to the fact that some plots have multiple conditions present.

Plot Designs and Number of Subplots

Annual inventory plots and some late periodic-era plots are based on a 4-subplot design, and are generally consistent across the U.S. However, during translation more than four subplots were processed for some annual-design plots. These "extra" subplots plots were carried over from periodic inventories to account for mortality. Their presence may or may not have an effect on FVS simulations, depending on user objectives. These plots occur in most states covered by the Southern FIA Region (see https://www.fia.fs.fed.us/about/about_us/#regional_offices), but only for the first one to several years of annual inventory, and no later than 2008 in any of those states.

Standard annual, 4-subplot design plots are coded in two different ways, depending on how they are used and how many conditions are present. For plots where only one condition is present and the optional FIA macroplot has not been used, plot design information is preserved in the FVS data tables (table 2). These variables represent the 1/24 acre subplots, 1/300 acre microplot, and the 5-inch breakpoint diameter. In the FVS TreeInit tables, the corresponding Tree_Count is 1 for trees 1 inch, and the actual count by species for seedlings.

For all other situations it is necessary to use the FVS design codes to treat tally data on a per-acre basis. The reasons for this vary by situation. In cases where macroplots are used (only California, Oregon, and Washington), the use of two breakpoint diameters (5 inches between subplots and microplots, and 24 or 30 inches between subplots and macroplots) cannot be coded in FVS design variables. In multi-conditions plots, conditions that occupy less than the entire plot footprint also cannot be represented in FVS because they can occupy fractions of subplots. For periodic inventories, the ability to represent design codes in FVS varies widely among states and inventories, so during the development of FIA2FVS 2.0 a decision was made to code all periodic data on a per-acre basis. In all of these cases, Tree_Count in the FVS TreeInit tables is populated with the appropriate number of trees per acre, based on the FIA plot design.

Based on user demand, FIA data can be projected in FVS at the FIA subplot level. This is relatively straightforward with annual design plots. However, due to the conversion of periodic inventory data into the current FIADB structure, subplot-level data could not be represented properly in FIADB for some periodic inventories. Therefore, there are no subplot entries in FIADB for these inventories, and no corresponding entries in the FVS PlotInit_Plot table. Inventories without subplot data are primarily periodic inventories of the 1980s and 1990s in states covered by the Northern FIA Region (see https://www.fia.fs.fed.us/about/about_us/#regional_offices).

Questions about plots with non-standard plot characteristics should be directed to contacts at the FIA program responsible for the states where they occur, because these special plots are not covered in standard FIADB documentation. For Regional contact information, see: https://www.fia.fs.fed.us/tools-data/customer-service/.

Table 2. Coding of FVS design variables based on FIA plot and condition situation.

	FVS Design Variables in StandInit_Cond and StandInit_Plot			
FIA Data Case	**Basal_Area_ Factor**	**Inv_Plot_Size**	**Brk_DBH**	**Num_Plots**
Plot – annual design, single-condition, macroplot not used	-24	300	5	4
Plot – annual design, single-condition, macroplot is used	0	1	999	1
Plot – annual design, multi-condition, macroplot not used	-24	300	5	4
Plot – annual design, multi-condition, macroplot is used	0	1	999	1
Periodic Plot – single-condition*	0	1	999	1
Condition – annual design, 1 condition present on plot, macroplot not used	-24	300	5	4
Condition – annual design, 1 condition present on plot, macroplot is used	0	1	999	1
Condition – annual design, >1 condition present on plot	0	1	999	1

*Periodic plot designs were not mapped, and always sampled a single condition

Variant Assignment to Plots

FIA2FVS uses FVS variant and location codes from the FIADB PLOTGEOM table, which in turn is populated by intersecting FIA public plot coordinates with the FVS Variant Map (fig. 5), which can be obtained here: https://www.fs.fed.us/fvs/documents/index.shtml. In keeping with the data privacy requirements of the 1998 Farm Bill, FIA does not publish exact plot coordinates, but "fuzzes" their location by up to 0.5 miles. Fuzzing will not ordinarily result in a plot crossing over a state line, since that would result in a conflicting data situation in FIADB. However, during the process of intersecting the map, it was found that the resolution of the base map was too coarse in some areas and did not match some boundaries exactly. For example, the centerline of the Ohio River is imprecise, resulting in the possibility that some plots that are located very close to the river may be assigned to the variant located on the opposite side. Many of these map errors have been found and corrected, and refinement of the map is a continuing process.

Periodic inventories from California (1994), Oregon (1999), and Washington (2001) have no entries in PLOTGEOM, and therefore have no FVS variant assignments for those inventories. Hawaii and the U.S. Territories do have entries in PLOTGEOM, but have no variant assignments because there is currently no coverage of those areas by FVS variants.

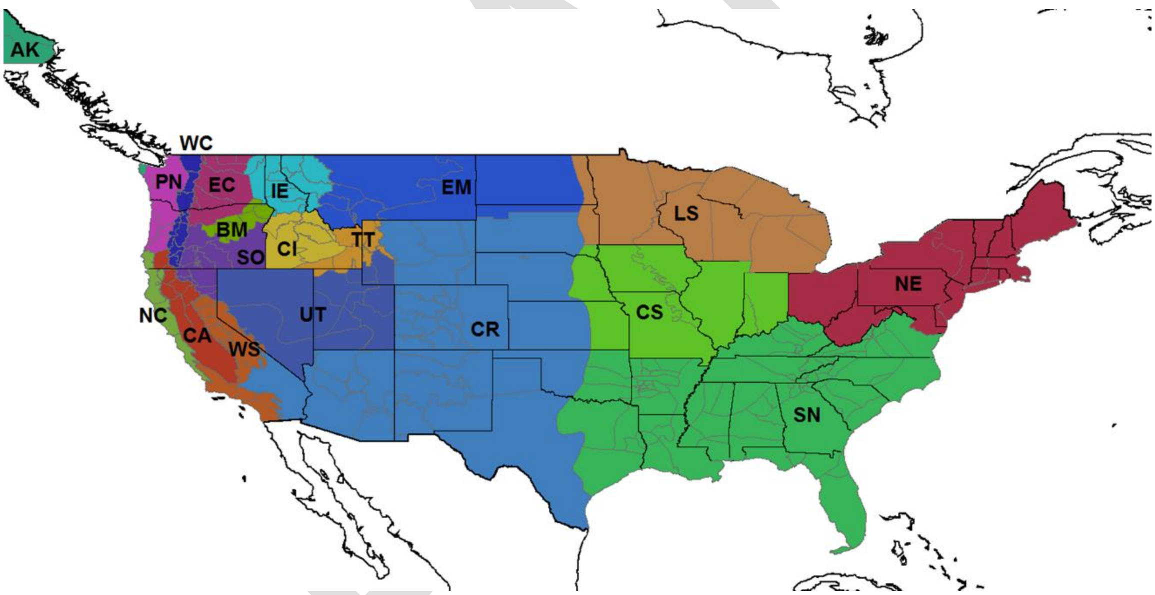


Figure 5. The FVS variant map. The boundaries separating the western EM and CR variants from the eastern LS, CS, and SN variants are generalized due to their location in mostly non-forested land.

A similar situation with the assignment of FVS location codes (fig. 6). In the past, the FVS variant map did not appear as in figure 5, and the location map did not exist at all. In the western states, FVS variant coverage appeared as "islands" centered on National Forest lands. FVS location codes were, and still are, primarily associated National Forests, and different FVS model parameters are associated with different location codes. Therefore, FVS users entering data from lands outside the National Forests were instructed to select the National Forest most representative of their land – usually the nearest, but sometimes based on similar landforms, soils, and other characteristics.

Given this history, and the fact that the location boundaries are somewhat approximate with respect to where one set of location parameters is better representative of forest characteristics vs another in the boundary area, in many areas the location boundaries can be considered suggestions. Since every available FIA plot location was intersected with the variant map to populate PLOTGEOM, FVS location codes have been assigned strictly according to the map in figure 6. However, note that because FVS locations are nested within variant boundaries, the same cross-boundary issues noted above will apply to some location assignments.

In situations where FVS simulations include plots that are near variant or location boundaries, users are encouraged to evaluate the default variant and location assignments and consider making changes where appropriate. Having all FIA data easily accessible in FVS-ready format is anticipated to accelerate FVS validation work, so in the future there may be additional guidance with respect to location assignment, or even some realignment of location boundaries. Meanwhile, users should use their best judgement when considering variant or location changes.

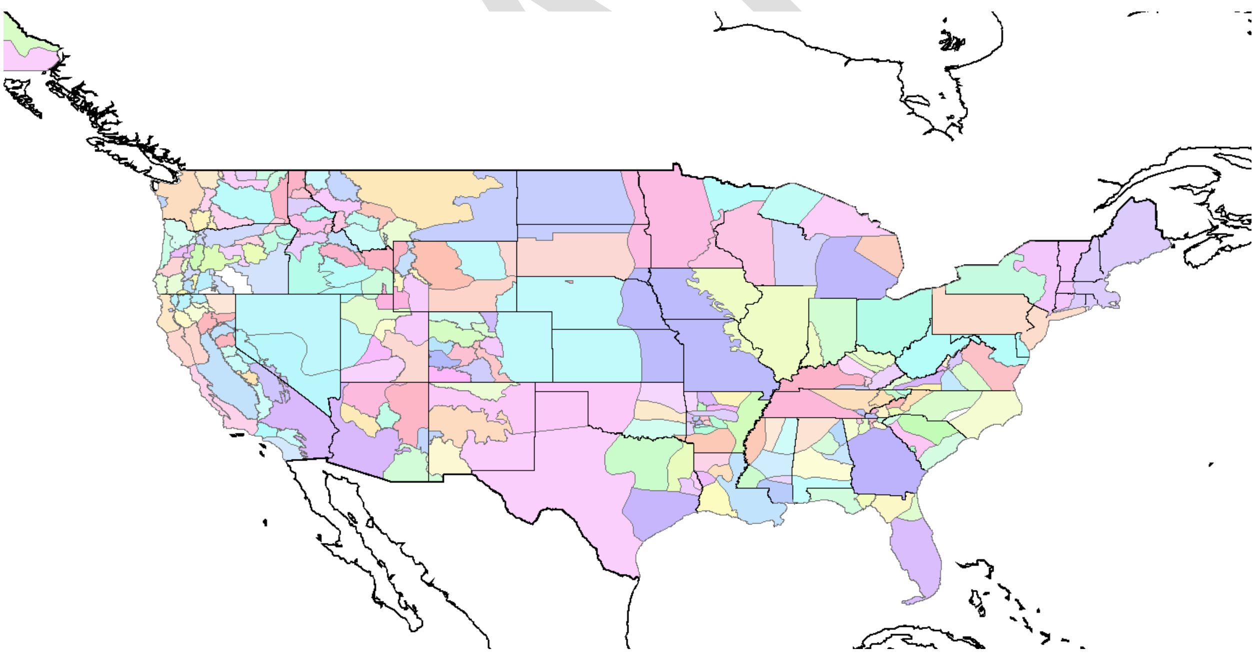


Figure 6. Within each variant, submodels are parameterized by species and location. Most locations are based in National Forest units, but other local parameterizations exist. For example, the sliver in central North Carolina is the Fort Bragg military installation location, which includes Fort Bragg and the surrounding Sandhills environment. It was parameterized using local inventory data.

Conclusion

This guide is intended to give users who are already familiar with FVS some basic guidance on obtaining and importing FIA data, and to highlight some important differences in the way FIA data are tabled in FVS as compared to “traditional” stand exam-type data. To make the most effective use out of FIA data in FVS, users should become familiar with FIA variable definitions, as they are described in the FIADB user’s guide (https://www.fia.fs.fed.us/library/database-documentation/index.php), and the large library of documentation available from the FVS web site (https://www.fs.fed.us/fvs/index.shtml).

Other documents supporting the use of FIA data in FVS are planned, such as an expanded user’s guide (similar to Dixon’s (2002) “Essential FVS”) and topic-driven documents, such as FVS variant validation or sensitivity analysis of certain FVS submodels. In addition, FVS-based projection examples are now being included in FIA state reports, starting with Nevada (Witt in press) and Utah (Menlove in press). These state report sections not only provide examples of how FIA data can be used in FVS, but also how information derived from past FIA can be used to set up effective and informative FVS simulations.

Acknowledgements

FIA2FVS 2.0 is the latest in a series of efforts that began in the early 2000s with Pat Miles (NRS-FIA) first making large quantities of FIA data available in FVS format through the FIA Mapmaker interface. Decommissioning of Mapmaker, which was forced by changes in Forest Service processing capabilities, left users who had started to learn about and use this valuable resource with a data gap. Don Vandendriesche, then a member of the FVS Staff, filled that gap for several years by writing FIA2FVS 1.0 and providing considerable user support. This support was maintained by the FVS staff long after Don had moved on to other positions, and lasted until a plan could be put into place for a more comprehensive, collaborative solution between the FIA and FVS programs. The plan for FIA2FVS 2.0 was championed and initially funded by the RMRS-FIA program, under Program Manager Michael Wilson, with the initial project management conducted by Information Management Team leader Troy Heithecker, and later by Doug Jacobs. Additional funding for programming work was provided by the FIA Washington Office. Starting with the FIA2FVS 1.0 effort and still continuing today, FVS Group leader Mike VanDyck has been a major supporter on the FVS side of the effort, as have Chad Keyser, Erin Smith-Mateja, and other members of the FVS staff.

References Cited

Crookston, N.L., 1997. Suppose: an interface to the Forest Vegetation Simulator. In: Teck, R., Moeur, M., Adams, J. (Comps.), Proceedings: Forest Vegetation Simulator Conference, February 3–7, 1997, Fort Collins, CO. Gen. Tech. Re INT-373. U.S. Department of Agriculture, Forest Service, Intermountain Research Station, Ogden, UT, pp. 7–14.

Crookston, N.L.; Gammel, D.L.; Rebain, S.; Robinson, D.; Keyser, C.E.; Dahl, C.A. 2003. (revised March 2015). 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. https://www.fs.fed.us/.ftproot/pub/fmsc/ftp/fvs/docs/gtr/DBSUserGuide.pdf

Dixon, G.E. (Comp.), 2002. Essential FVS: A User's Guide to the Forest Vegetation Simulator. Internal Report. U.S. Department of Agriculture, Forest Service, Forest Management Service Center, Fort Collins, CO, 189 pp. https://www.fs.fed.us/.ftproot/pub/fmsc/ftp/fvs/docs/gtr/EssentialFVS.pdf

Menlove, J.; Shaw, J.D.; Goeking, S.A.; Bakken, J.L. In progress. Utah's Forest Resources, 2008-2017. Resour. Bull. RMRS-RB-XXXX. Fort Collins, CO: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station, 000 p.

Miles, P.D. 2008. Forest inventory and analysis data for FVS modelers. In: Havis, R.N.; Crookston, N.L., comps. 2008. Third Forest Vegetation Simulator Conference; 2007 February 13–15; Fort Collins, CO. Proceedings RMRS-P-54. Fort Collins, CO: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station. p. 125-129. https://www.fs.usda.gov/treesearch/pubs/30974

Shaw, J.D. 2009. Using FIA data in the Forest Vegetation Simulator. In: McWilliams, W.; Moisen, G.; Czaplewski, R., comps. Forest Inventory and Analysis (FIA) Symposium 2008; October 21-23, 2008; Park City, UT. Proc. RMRS-P-56CD. Fort Collins, CO: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station. 16 p. https://www.fs.usda.gov/treesearch/pubs/33330

Shaw, J.D.; Gagnon, A. 2019. Field note: A new conversion of Forest Inventory and Analysis data for use in the Forest Vegetation Simulator. Journal of Forestry 118(3):307-312. https://www.fs.usda.gov/treesearch/pubs/59245

Witt, C.; Goeking, S.A.; Shaw, J.D.; Bakken, J.L.; Menlove, J. In press. Nevada's Forest Resources, 2009-2018. Resour. Bull. RMRS-RB-XXXX. Fort Collins, CO: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station, 000 p.

Appendix A.

Column list and translation logic for records in FVS StandInit and PlotInit tables

Column Name	GH #	StandInit_cond table	GH#	StandInit_plot table	GH#	PlotInit_plot table
n/a		PLOT = current record in the PLOT table, identified by PLOT.CN COND = current record in the COND table, identified by PLOT.CN and COND.CONDID		PLOT = current record in the PLOT table, identified by PLOT.CN COND = record in the COND table having the largest value of COND.CONDPROP_UNADJ where COND.FORTYPCD>0 , identified by PLOT.CN and COND.CONDID		PLOT = current record in the PLOT table, identified by PLOT.CN COND = record in the COND table having the largest value of COND.CONDPROP_UNADJ where COND.FORTYPCD>0 , identified by PLOT.CN and COND.CONDID
Stand_CN	1	StandInit_cond.Stand_CN = COND.CN	85	StandInit_plot.Stand_CN = PLOT.CN	168	PlotInit_plot.Stand_CN = PLOT.CN
Stand_ID	2	StandInit_cond.Stand_ID = concatenation of: STATECD(4) + INVYR(4) + CYCLE(2) + SUBCYCLE(2) + UNITCD(2) + COUNTYCD(3) + PLOT(5) + CONDID (1).	86	StandInit_plot.Stand_ID = concatenation of: PLOT.STATECD(4) + PLOT.INVYR(4) + PLOT.CYCLE(2) + PLOT.SUBCYCLE(2) + PLOT.UNITCD(2) + PLOT.COUNTYCD(3) + PLOT.PLOT(5).	169	PlotInit_plot.Stand_ID = concatenation of: PLOT.STATECD(4) + PLOT.INVYR(4) + PLOT.CYCLE(2) + PLOT.SUBCYCLE(2) + PLOT.UNITCD(2) + PLOT.COUNTYCD(3) + PLOT.PLOT(5).
StandPlot_CN	n/a		n/a		170	PlotInit_plot.StandPlot_CN=concatenation of Stand_CN+"_"+ SUBPLOT.SUBP
StandPlot_ID	n/a		n/a		171	PlotInit_plot.StandPlot_ID=concatenation of Stand_ID+"_"+ SUBPLOT.SUBP
Variant	3	StandInit_cond.Variant=PLOTGEOM.FVS_VARIANT	87	StandInit_plot.Variant= PLOTGEOM.FVS_VARIANT	172	PlotInit_plot.Variant= PLOTGEOM.FVS_VARIANT
Inv_Year	4	StandInit_cond.Inv_Year = PLOT.MEASYEAR	88	StandInit_plot.Inv_Year = PLOT.MEASYEAR	173	PlotInit_plot.Inv_Year = PLOT.MEASYEAR
Inv_Month	5	StandInit_cond.StandInit.Inv_Month = PLOT.MEASMON	89	StandInit_plot.StandInit.Inv_Month = PLOT.MEASMON	174	PlotInit_plot.StandInit.Inv_Month = PLOT.MEASMON
Inv_Day	xxx	StandInit_cond.Inv_Day = PLOT.MEASDAY	xxx	StandInit_plot.Inv_Day = PLOT.MEASDAY	xxx	PlotInit_plot.Inv_Day = PLOT.MEASDAY
Groups	6	StandInit_cond.Groups = Construct string: "FIA_Conditions LULC=[LULC$] FIA_Inv_Yr=[PLOT.INVYR] FIA_Meas_Yr=[PLOT.MEASYR] State=[PLOT.STATECD] Inv_Kind=[PLOT.KINDCD] Location=[COND.ADFORCD] For_Type=[COND.FORTYPCD] Field_Type=[COND.FLDTYPCD] Variant=[variant] NumConds=[NumConds$]" LULC$= IF COND.FORTYPCD>0, THEN "Forest", ELSE COND.PRESNFCD NumConds$= IF COND.CONDPROP_UNADJ = 1, THEN "Single", ELSE "Multiple"	90	StandInit_cond.Groups = Construct string: "FIA_Conditions LULC=[LULC$] FIA_Inv_Yr=[PLOT.INVYR] FIA_Meas_Yr=[PLOT.MEASYR] State=[PLOT.STATECD] Inv_Kind=[PLOT.KINDCD] Location=[COND.ADFORCD] For_Type=[COND.FORTYPCD] Field_Type=[COND.FLDTYPCD] Variant=[variant] NumConds=[NumConds$]" LULC$= IF COND.FORTYPCD>0, THEN "Forest", ELSE COND.PRESNFCD NumConds$= IF COND.CONDPROP_UNADJ = 1, THEN "Single", ELSE "Multiple"	175	StandInit_cond.Groups = Construct string: "FIA_Conditions LULC=[LULC$] FIA_Inv_Yr=[PLOT.INVYR] FIA_Meas_Yr=[PLOT.MEASYR] State=[PLOT.STATECD] Inv_Kind=[PLOT.KINDCD] Location=[COND.ADFORCD] For_Type=[COND.FORTYPCD] Field_Type=[COND.FLDTYPCD] Variant=[variant] NumConds=[NumConds$]" LULC$= IF COND.FORTYPCD>0, THEN "Forest", ELSE COND.PRESNFCD NumConds$= IF COND.CONDPROP_UNADJ = 1, THEN "Single", ELSE "Multiple"
AddFiles	7	Pass null value.	91	Pass null value.	176	Pass null value.
FVSKeywords	8	Pass null value.	92	Pass null value.	177	Pass null value.
GIS_Link	9	Pass null value.	93	Pass null value.	178	Pass null value.
Latitude	10	StandInit_cond.Latitude = PLOT.LAT from FS_FIADB	94	StandInit_plot.Latitude = PLOT.LAT from FS_FIADB	179	PlotInit_plot.Latitude = PLOT.LAT from FS_FIADB
Longitude	11	StandInit_cond.Longitude = PLOT.LON from FS_FIADB	95	StandInit_plot.Longitude = PLOT.LON from FS_FIADB	180	PlotInit_plot.Longitude = PLOT.LON from FS_FIADB
Datum	12	StandInit_cond.Datum = NAD83	96	StandInit_plot.Datum = NAD83	181	PlotInit_plot.Datum = NAD83
Region	13	StandInit_cond.Variant = PLOTGEOM.FVS_REGION	97	StandInit_plot.Variant = PLOTGEOM.FVS_REGION	182	PlotInit_plot.Variant = PLOTGEOM.FVS_REGION
Forest	14	StandInit_cond.Forest = PLOTGEOM.FVS_FOREST	98	StandInit_plot.Forest = PLOTGEOM.FVS_ FOREST	183	PlotInit_plot.Forest = PLOTGEOM.FVS_ FOREST
District	15	StandInit_cond.District = PLOTGEOM.FVS_DISTRICT	99	StandInit_plot.District = PLOTGEOM.FVS_ DISTRICT	184	PlotInit_plot.District = PLOTGEOM.FVS_ DISTRICT
Compartment	16	Pass null value	100	Pass null value	185	Pass null value
Location	17	StandInit_cond.Variant = PLOTGEOM.FVS_LOC_CD	101	StandInit_plot.Variant = PLOTGEOM.FVS_LOC_CD	186	StandInit_cond.Variant = PLOTGEOM.FVS_LOC_CD
Ecoregion	18	StandInit_cond.Ecoregion = PLOT.ECOSUBCD	102	StandInit_plot.Ecoregion = PLOT.ECOSUBCD	187	PlotInit_plot.Ecoregion = PLOT.ECOSUBCD

PV_Code	19	IF (PLOT.STATECD = 4 or PLOT.STATECD = 6 or PLOT.STATECD = 8 or PLOT.STATECD = 16 or PLOT.STATECD = 30 or PLOT.STATECD = 31 or PLOT.STATECD = 32 or PLOT.STATECD = 35 or PLOT.STATECD = 38 or PLOT.STATECD = 41 or PLOT.STATECD = 46 or PLOT.STATECD = 49 or PLOT.STATECD = 53 or PLOT.STATECD = 55 or PLOT.STATECD = 56 or PLOT.STATECD = 72 or PLOT.STATECD = 78) and (PLOT.FVS_REGION = Not Is Null and PLOT.FVS_VARIANT = Not Is Null and COND.HABTYPCD1 = Not Is Null and COND.HABTYPCD1_PUB_CD= Not Is Null)
		THEN StandInit_cond.PV_Code = FVS_SEQUENCE
		Where
		REF_FVS_VEGCODE.REGION = PLOT.FVS_REGION
		REF_FVS_VEGCODE.VARIANT = PLOT.FVS_VARIANT
		REF_FVS_VEGCODE. COND.HABTYPCD1 = COND.HABTYPCD1
		REF_FVS_VEGCODE. HABTYPCD1_PUB_CD = COND.HABTYPCD1_PUB_CD
		ELSE
		StandInit_cond.PV_Code is null
			103	IF (PLOT.STATECD = 4 or PLOT.STATECD = 6 or PLOT.STATECD = 8 or PLOT.STATECD = 16 or PLOT.STATECD = 30 or PLOT.STATECD = 31 or PLOT.STATECD = 32 or PLOT.STATECD = 35 or PLOT.STATECD = 38 or PLOT.STATECD = 41 or PLOT.STATECD = 46 or PLOT.STATECD = 49 or PLOT.STATECD = 53 or PLOT.STATECD = 55 or PLOT.STATECD = 56 or PLOT.STATECD = 72 or PLOT.STATECD = 78) and (PLOT.FVS_REGION = Not Is Null and PLOT.FVS_VARIANT = Not Is Null and COND.HABTYPCD1 = Not Is Null and COND.HABTYPCD1_PUB_CD= Not Is Null)
				THEN StandInit_plot.PV_Code = FVS_SEQUENCE
				Where
				REF_FVS_VEGCODE.REGION = PLOT.FVS_REGION
				REF_FVS_VEGCODE.VARIANT = PLOT.FVS_VARIANT
				REF_FVS_VEGCODE. HABTYPCD1 = COND.HABTYPCD1
				REF_FVS_VEGCODE. HABTYPCD1_PUB_CD = COND.HABTYPCD1_PUB_CD
				ELSE
				PV_Code is null
					188	IF (PLOT.STATECD = 4 or PLOT.STATECD = 6 or PLOT.STATECD = 8 or PLOT.STATECD = 16 or PLOT.STATECD = 30 or PLOT.STATECD = 31 or PLOT.STATECD = 32 or PLOT.STATECD = 35 or PLOT.STATECD = 38 or PLOT.STATECD = 41 or PLOT.STATECD = 46 or PLOT.STATECD = 49 or PLOT.STATECD = 53 or PLOT.STATECD = 55 or PLOT.STATECD = 56 or PLOT.STATECD = 72 or PLOT.STATECD = 78) and (PLOT.FVS_REGION = Not Is Null and PLOT.FVS_VARIANT = Not Is Null and COND.HABTYPCD1 = Not Is Null and COND.HABTYPCD1_PUB_CD= Not Is Null)
						THEN PlotInit_plot.PV_Code = FVS_SEQUENCE
						Where
						REF_FVS_VEGCODE.REGION = PLOT.FVS_REGION
						REF_FVS_VEGCODE.VARIANT = PLOT.FVS_VARIANT
						REF_FVS_VEGCODE. HABTYPCD1 = COND.HABTYPCD1
						REF_FVS_VEGCODE. HABTYPCD1_PUB_CD = COND.HABTYPCD1_PUB_CD
						ELSE
						PV_Code is null

PV_Ref_Code	20	StandInit_cond.PV_Ref_Code = COND.HABTYPCD1_PUB_CD	104	StandInit_plot.PV_Ref_Code = COND.HABTYPCD1_PUB_CD	189	PlotInit_plot.PV_Ref_Code = COND.HABTYPCD1_PUB_CD
PV_FIA_HabtypCD1	13 5	StandInit_cond.PV_FIA_HabtypCD1 = COND.HABTYPCD1	167	StandInit_plot.PV_FIA_HabtypCD1 = COND.HABTYPCD1 Where COND.CONDID is assigned using the "largest forested condition" logic	190	PlotInit_plot.PV_FIA_HabtypCD1 = COND.HABTYPCD1 Where COND.CONDID is assigned using the "largest forested condition" logic
Age	21	StandInit_cond.Age = COND.STDAGE	105	StandInit_plot.Age = COND.STDAGE	191	PlotInit_plot.Age = COND.STDAGE
Aspect	22	StandInit_cond.Aspect = COND.ASPECT	106	StandInit_plot.Aspect = COND.ASPECT	192	PlotInit_plot.Aspect = COND.ASPECT
Slope	23	StandInit_cond.Slope = COND.SLOPE	107	StandInit_plot.Slope = COND.SLOPE	193	PlotInit_plot.Slope = COND.SLOPE
Elevation	24	Pass null value.	108	Pass null value.	Xxx	Pass null value.
ElevFt	25	StandInit_cond.ElevFt = PLOT.ELEV	109	StandInit_plot.ElevFt = PLOT.ELEV	Xxx	PlotInit_plot.ElevFt = PLOT.ELEV
Basal_Area_Factor Inv_Plot_Size Brk_DBH Num_Plots	26 27 28 29	IF PLOT.DESIGNCD = 1 and COND.CONDPROP_UNADJ = 1 and PLOT.MACRO_BRKPT_DIA Is Null 'Include design info only for current mapped-plot design that have only one condition; exclude macroplot designs THEN StandInit_cond.Basal_Area_Factor = -24 StandInit_cond.Inv_Plot_Size = 300 StandInit_cond.Brk_DBH = 5 StandInit_cond.Num_Plots = 4 ELSE 'Multi-condition plot and all other designs; treed conditions only StandInit_cond.Basal_Area_Factor = 0 StandInit_cond.Inv_Plot_Size = 1 StandInit_cond.Brk_DBH = 999 StandInit_cond.NumPlots = 1	110 111 112 113	IF PLOT.DESIGNCD = 1 and PLOT.MACRO_BRKPT_DIA Is Null 'Include design info only for current mapped-plot design ; exclude macroplot designs THEN StandInit_plot.Basal_Area_Factor = -24 StandInit_plot.Inv_Plot_Size = 300 StandInit_plot.Brk_DBH = 5 StandInit_plot.Num_Plots = 4 ELSE 'Multi-condition plot and all other designs; treed conditions only StandInit_plot.Basal_Area_Factor = 0 StandInit_plot.Inv_Plot_Size = 1 StandInit_plot.Brk_DBH = 999 StandInit_plot.NumPlots = 1	196 197 198 199	IF PLOT.DESIGNCD = 1 and PLOT.MACRO_BRKPT_DIA Is Null 'Include design info only for current mapped-plot design; exclude macroplot designs THEN PlotInit_plot.Area_Factor = -24 PlotInit_plot.Inv_Plot_Size = 300 PlotInit_plot.Brk_DBH = 5 PlotInit_plot.Num_Plots = 1 ELSE 'Multi-condition plot and all other designs; treed conditions only PlotInit_plot.Basal_Area_Factor = 0 PlotInit_plot.Inv_Plot_Size = 1 PlotInit_plot.Brk_DBH = 999 PlotInit_plot.NumPlots = 1
NonStk_Plots	30	Pass null value.	114	Pass null value.	200	Pass null value.
Sam_Wt	31	StandInit_cond.Sam_Wt = PLOTSNAP.EXPALL*COND.CONDPROP_UNADJ	115	StandInit_plot.Sam_Wt = PLOTSNAP.EXPALL	201	PlotInit_plot.Sam_Wt = PLOTSNAP.EXPALL/NumPlots
Stk_Pcnt	32	Pass null value.	116	Pass null value.	202	Pass null value.
DG_Trans	33	If PLOT.REMPER>=1 Then StandInit_cond.DG_Trans=1 Else StandInit_cond.DG_Trans is null	117	If PLOT.REMPER>=1 Then StandInit_plot.DG_Trans=1 Else StandInit_plot.DG_Trans is null	203	If PLOT.REMPER>=1 Then PlotInit_plot.DG_Trans=1 Else DG_Trans is null
DG_Measure	34	If PLOT.REMPER>=1 Then StandInit_cond.DG_Measure = RoundInteger (PLOT.REMPER)	118	If PLOT.REMPER>=1 Then StandInit_plot.DG_Measure = RoundInteger (PLOT.REMPER)	204	If PLOT.REMPER>=1 Then PlotInit_plot.DG_Measure = RoundInteger (PLOT.REMPER)

HTG_Trans	35	If PLOT.REMPER>=1 Then StandInit_cond.HTG_Trans=1 Else StandInit_cond.HTG_Trans is null	119	If PLOT.REMPER>=1 Then StandInit_plot.HTG_Trans=1 Else StandInit_plot.HTG_Trans is null	205	If PLOT.REMPER>=1 Then PlotInit_plot.HTG_Trans=1 Else PlotInit_plot.HTG_Trans is null
HTG_Measure	13 6	If PLOT.REMPER>=1 Then StandInit_cond.HTG_Measure = RoundInteger (PLOT.REMPER)	120	If PLOT.REMPER>=1 Then StandInit_plot.HTG_Measure = RoundInteger (PLOT.REMPER)	206	If PLOT.REMPER>=1 Then PlotInit_plot.HTG_Measure = RoundInteger (PLOT.REMPER)
Mort_Measure	36	If PLOT.REMPER>=1 Then StandInit_cond.Mort_Measure = RoundInteger (PLOT.REMPER)	121	If PLOT.REMPER>=1 Then StandInit_plot.Mort_Measure = RoundInteger (PLOT.REMPER)	207	If PLOT.REMPER>=1 Then PlotInit_plot.Mort_Measure = RoundInteger (PLOT.REMPER)
Max_BA	37	Pass null value.	123	Pass null value.	208	Pass null value.
Max_SDI	38	Pass null value.	124	Pass null value.	209	Pass null value.
Max_SDI_FIA	xxx	StandInit_cond.Max_SDI_FIA = COND.SDIMAX_RMRS	Xxx	StandInit_plot.Max_SDI_FIA = COND.SDIMAX_RMRS	Xxx	PlotInit_plot.Max_SDI_FIA = COND.SDIMAX_RMRS
Site_Species	39	StandInit_cond.Site_Species = COND.SISP StandInit_cond.Site_Index = COND.SICOND StandInit_cond.Site_Index_Base_Age = COND.SIBASE	124	StandInit_plot.Site_Species = COND.SISP StandInit_plot.Site_Index = COND.SICOND StandInit_plot.Site_Index_Base_Age = COND.SIBASE	210	PlotInit_plot.Site_Species = COND.SISP PlotInit_plot.Site_Index = COND.SICOND PlotInit_plot.Site_Index_Base_Age = COND.SIBASE
Site_Index	40		125		211	
Site_Index_Base_Ag e	41		126		212	
Model_Type	42	Pass null value.	127	Pass null value.	213	Pass null value.
Physio_Region	43	StandInit_cond.Physio_Region = COND.PHYSCLCD	128	StandInit_plot.Physio_Region = COND.PHYSCLCD	214	PlotInit_plot.Physio_Region = COND.PHYSCLCD
Forest_Type	44	Pass null value.	129	Pass null value.	215	Pass null value.
Forest_Type_FIA	xxx	IF COND.COND_STATUS_CD = 1 and COND.FLDTYPCD = IsNull THEN StandInit_cond.Forest_Type = COND.FORTYPCD ELSE StandInit_cond.Forest_Type = COND.FLDTYPCD	Xxx	IF COND.COND_STATUS_CD = 1 and COND.FLDTYPCD = IsNull THEN StandInit_plot.Forest_Type = COND.FORTYPCD ELSE StandInit_plot.Forest_Type = COND.FLDTYPCD	Xxx	IF COND.COND_STATUS_CD = 1 and COND.FLDTYPCD = IsNull THEN PlotInit_plot.Forest_Type = COND.FORTYPCD ELSE PlotInit_plot.Forest_Type = COND.FLDTYPCD
State	45	StandInit_cond.State = COND.STATECD	130	StandInit_plot.State = PLOT.STATECD	216	PlotInit_plot.State = PLOT.STATECD
County	46	StandInit_cond.County = COND.COUNTYCD	131	StandInit_plot.County = PLOT.COUNTYCD	217	PlotInit_plot.County = PLOT.COUNTYCD
Fuel_Model	47	StandInit_cond.Fuel_Model = DWM_FUELBED_TYPCD	132	StandInit_plot.Fuel_Model = COND.DWM_FUELBED_TYPCD	218	Fuel_Model = COND.DWM_FUELBED_TYPCD
Fuel_0_25_H	48	StandInit_cond.Fuel_0_25_H = COND_DWM_CALC.FWD_SM_DRYBIO_COND/2000	281	StandInit_plot.Fuel_0_25_H = COND_DWM_CALC.FWD_SM_DRYBIO_COND/2000	219	PlotInit_plot.Fuel_0_25_H = COND_DWM_CALC.FWD_SM_DRYBIO_COND/2000
Fuel_25_1_H	49	StandInit_cond.Fuel_25_1_H = COND_DWM_CALC.FWD_MD_DRYBIO_COND/2000	282	StandInit_plot.Fuel_25_1_H = COND_DWM_CALC.FWD_MD_DRYBIO_COND /2000	220	PlotInit_plot.Fuel_25_1_H = COND_DWM_CALC.FWD_MD_DRYBIO_COND /2000
Fuel_1_3_H	50	StandInit_cond.Fuel_1_3_H = COND_DWM_CALC.FWD_LG_DRYBIO_COND/2000	283	StandInit_plot.Fuel_1_3_H = COND_DWM_CALC.FWD_LG_DRYBIO_COND/2000	221	PlotInit_plot.Fuel_1_3_H = COND_DWM_CALC.FWD_LG_DRYBIO_COND/2000
Fuel_3_6_H	51	StandInit_cond.Fuel_3_6_H=((Sum Of DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_COND)/2000) Where (DWM_COARSE_WOODY_DEBRIS.DECAYCD=1 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=2 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=3) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=3 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<6)	284	StandInit_plot.Fuel_3_6_H=((Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_PLOT over StandInit_plot.Stand_ID)/2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=1 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=2 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=3) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=3 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<6)	222	PlotInit_plot.Fuel_3_6_H=((Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_PLOT over StandInit.StandPlot_ID)/2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=1 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=2 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=3) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=3 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<6)
Fuel_6_12_H	52	StandInit_cond.Fuel_6_12_H=((Sum Of DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_COND)/2000) Where (DWM_COARSE_WOODY_DEBRIS.DECAYCD=1 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=2 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=3) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=6 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<12)	285	StandInit_plot.Fuel_6_12_H=((Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_ PLOT over StandInit_plot.Stand_ID)/2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=1 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=2 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=3) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=6 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<12)	223	PlotInit_plot.Fuel_6_12_H=((Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_PLOT over StandInit.StandPlot_ID)/2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=1 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=2 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=3) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=6 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<12)

Fuel_12_20_H	53	StandInit_cond.Fuel_12_20_H=((Sum Of DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_COND)/2000) Where (DWM_COARSE_WOODY_DEBRIS.DECAYCD=1 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=2 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=3) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=12 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<20)	286	StandInit_plot.Fuel_12_20_H=((Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_ PLOT over StandInit_plot..Stand_ID)/2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=1 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=2 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=3) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=12 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<20)	224	PlotInit_plot.Fuel_12_20_H=((Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_PLOT over StandInit.StandPlot_ID)/2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=1 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=2 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=3) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=12 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<20)
Fuel_20_35_H	54	StandInit_cond.Fuel_20_35_H=((Sum Of DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_COND)/2000) Where (DWM_COARSE_WOODY_DEBRIS.DECAYCD=1 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=2 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=3) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=20 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<35)	287	StandInit_plot.Fuel_20_35_H=((Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_ PLOT over StandInit_plot.StandInit.Stand_ID)/2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=1 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=2 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=3) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=20 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<35)	225	PlotInit_plot.Fuel_20_35_H=((Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_PLOT over StandInit.StandPlot_ID)/2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=1 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=2 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=3) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=20 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<35)
Fuel_35_50_H	55	StandInit_cond.Fuel_35_50_H=((Sum Of DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_COND)/2000) Where (DWM_COARSE_WOODY_DEBRIS.DECAYCD=1 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=2 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=3) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=35 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<50)	288	StandInit_plot.Fuel_35_50_H=((Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_ PLOT over StandInit_plot.Stand_ID)/2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=1 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=2 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=3) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=35 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<50)	226	PlotInit_plot.Fuel_35_50_H=((Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_PLOT over StandInit.StandPlot_ID)/2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=1 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=2 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=3) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=35 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<50)
Fuel_gt_50_H	56	StandInit_cond.Fuel_gt_50_H=((Sum Of DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_COND)/2000) Where (DWM_COARSE_WOODY_DEBRIS.DECAYCD=1 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=2 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=3) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=50)	289	StandInit_plot.Fuel_gt_50_H=((Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_ PLOT over StandInit_plot.Stand_ID)/2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=1 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=2 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=3) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=50)	227	PlotInit_plot.Fuel_gt_50_H=((Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_PLOT over StandInit.StandPlot_ID)/2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=1 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=2 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=3) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=50)
Fuel_0_25_S	57	Pass null value.	290	Pass null value. FIA data do not include decay classes for FWD, so only hard class is used.	228	Pass null value. FIA data do not include decay classes for FWD, so only hard class is used.
Fuel_25_1_S	58	Pass null value.	291	Pass null value. FIA data do not include decay classes for FWD, so only hard class is used.	229	Pass null value. FIA data do not include decay classes for FWD, so only hard class is used.
Fuel_1_3_S	58	Pass null value.	292	Pass null value. FIA data do not include decay classes for FWD, so only hard class is used.	230	Pass null value. FIA data do not include decay classes for FWD, so only hard class is used.
Fuel_3_6_S	60	StandInit_cond.Fuel_3_6_S=((Sum Of DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_COND)/2000) Where (DWM_COARSE_WOODY_DEBRIS.DECAYCD=4 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=5) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=3 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<6)	293	StandInit_plot.Fuel_3_6_S=((Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_PLOT over StandInit_plot.Stand_ID)/2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=4 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=5) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=3 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<6)	231	PlotInit_plot.Fuel_3_6_S=((Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_PLOT over StandInit.StandPlot_ID)/2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=4 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=5) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=3 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<6)
Fuel_6_12_S	61	StandInit_cond.Fuel_6_12_S=(Sum Of DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_COND/2000) Where (DWM_COARSE_WOODY_DEBRIS.DECAYCD=4 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=5) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=6 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<12)	294	StandInit_plot.Fuel_6_12_S=(Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_PLOT over StandInit_plot.Stand_ID /2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=4 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=5) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=6 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<12)	232	PlotInit_plot.Fuel_6_12_S=(Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_PLOT over StandInit.StandPlot_ID /2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=4 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=5) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=6 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<12)
Fuel_12_20_S	62	StandInit_cond.Fuel_12_20_H=((Sum Of DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_COND)/2000) Where (DWM_COARSE_WOODY_DEBRIS.DECAYCD=4 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=5) AND	295	StandInit_plot.Fuel_12_20_H=((Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_PLOT over StandInit_plot.Stand_ID)/2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=4 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=5) AND	233	PlotInit_plot.Fuel_12_20_H=((Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_PLOT over StandInit.StandPlot_ID)/2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=4 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=5) AND

		(DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=12 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<20)		(DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=12 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<20)		(DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=12 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<20)
Fuel_20_35_S	63	StandInit_cond.Fuel_20_35_H=((Sum Of DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_COND)/2000) Where (DWM_COARSE_WOODY_DEBRIS.DECAYCD=4 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=5) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=20 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<35)	296	StandInit_plot.Fuel_20_35_H=((Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_PLOT over StandInit_plot.Stand_ID)/2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=4 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=5) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=20 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<35)	234	PlotInit_plot.Fuel_20_35_H=((Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_PLOT over StandInit.StandPlot_ID)/2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=4 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=5) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=20 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<35)
Fuel_35_50_S	64	StandInit_cond.Fuel_35_50_H=((Sum Of DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_COND)/2000) Where (DWM_COARSE_WOODY_DEBRIS.DECAYCD=5 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=5) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=35 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<50)	297	StandInit_plot.Fuel_35_50_H=((Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_PLOT over StandInit_plot.Stand_ID)/2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=5 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=5) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=35 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<50)	235	PlotInit_plot.Fuel_35_50_H=((Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_PLOT over StandInit.StandPlot_ID)/2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=5 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=5) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=35 AND DWM_COARSE_WOODY_DEBRIS.TRANSDIA<50)
Fuel_gt_50_S	65	StandInit_cond.Fuel_gt_50_S=((Sum Of DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_COND)/2000) Where (DWM_COARSE_WOODY_DEBRIS.DECAYCD=4 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=5) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=50)	298	StandInit_plot.Fuel_gt_50_S=((Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_PLOT over StandInit_plot.Stand_ID)/2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=4 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=5) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=50)	236	PlotInit_plot.Fuel_gt_50_S=((Sum DWM_COARSE_WOODY_DEBRIS.DRYBIO_AC_PLOT over StandInit.StandPlot_ID)/2000) Where DWM_COARSE_WOODY_DEBRIS.DECAYCD.CWD_SAMPLE_METHOD>0 AND (DWM_COARSE_WOODY_DEBRIS.DECAYCD=4 OR DWM_COARSE_WOODY_DEBRIS.DECAYCD=5) AND (DWM_COARSE_WOODY_DEBRIS.TRANSDIA>=50)
Fuel_Litter	66	StandInit_cond.Fuel_Litter = COND_DWM_CALC.LITTER_BIOMASS / 2000	299	StandInit_plot.Fuel_Litter = COND_DWM_CALC.LITTER_BIOMASS / 2000 Where, for the plot being processed, condition-level variables are based on the largest forested condition on the plot. The condition used to populate this variable is the same as the condition used in StandInit_plot.Groups.	237	PlotInit_plot.Fuel_Litter = COND_DWM_CALC.LITTER_BIOMASS / 2000 Where, for the plot being processed, condition-level variables are based on the largest forested condition on the plot. The condition used to populate this variable is the same as the condition used in StandInit.Groups.
Fuel_Duff	67	StandInit_cond.Fuel_Duff = COND_DWM_CALC.DUFF_BIOMASS / 2000	300	StandInit_plot.Fuel_Duff = COND_DWM_CALC.DUFF_BIOMASS / 2000 Where, for the plot being processed, condition-level variables are based on the largest forested condition on the plot. The condition used to populate this variable is the same as the condition used in StandInit.Groups.	238	PlotInit_plot.Fuel_Duff = COND_DWM_CALC.DUFF_BIOMASS / 2000 Where, for the plot being processed, condition-level variables are based on the largest forested condition on the plot. The condition used to populate this variable is the same as the condition used in StandInit.Groups.
Photo_Ref	68	Pass null value. FIA not using photo reference method.	133	Pass null value. FIA not using photo reference method.	239	Pass null value. FIA not using photo reference method.
Photo_Code	69	Pass null value. FIA not using photo reference method.	134	Pass null value. FIA not using photo reference method.	240	Pass null value. FIA not using photo reference method.

Appendix B.
Column list and translation logic for tree records in FVS TreeInit tables

Column Name	GH #	TreeInit_cond Description and translation logic	GH #	TreeInit_plot Description and translation logic
N/A	139	Tree records allowed for translation must meet the following criteria: (TPA_UNADJ >0 or TPAGROW_UNADJ >0) 'Must have either TPA_UNADJ or TPAGROW_UNADJ populated And ((TREE.STATUSCD = 1 and TREE.DIA is not null and (TREE.DIA > 0 or TREE.DIACALC >0)) 'Live trees on sampled forest and sampled nonforest Or (TREE.STATUSCD = 2 and TREE.STANDING_DEAD_CD = 1 and TREE.DIA is not null and (TREE.DIA > 0 or TREE.DIACALC >0)) 'All standing dead trees on sampled forest and sampled nonforest Or (TREE.STATUSCD = 2 and TREE.AGENTCD >0) 'All mortality trees Or (TREE.STATUSCD = 3)) 'All removal trees	241	Tree records allowed for translation must meet the following criteria: (TPA_UNADJ >0 or TPAGROW_UNADJ >0) 'Must have either TPA_UNADJ or TPAGROW_UNADJ populated And ((TREE.STATUSCD = 1 and TREE.DIA is not null and (TREE.DIA > 0 or TREE.DIACALC >0)) 'Live trees on sampled forest and sampled nonforest Or (TREE.STATUSCD = 2 and TREE.STANDING_DEAD_CD = 1 and TREE.DIA is not null and (TREE.DIA > 0 or TREE.DIACALC >0)) 'All standing dead trees on sampled forest and sampled nonforest Or (TREE.STATUSCD = 2 and TREE.AGENTCD >0) 'All mortality trees Or (TREE.STATUSCD = 3)) 'All removal trees
Stand_CN	81	TreeInit_cond.Stand_CN = COND.CN	242	TreeInit_plot.Stand_CN = PLOT.CN
Stand_ID	82	TreeInit_cond.Stand_ID = concatenation of: STATECD(4) + INVYR(4) + CYCLE(2) + SUBCYCLE(2) + UNITCD(2) + COUNTYCD(3) + PLOT(5) + CONDID (1).	243	TreeInit_plot.Stand_ID = concatenation of: STATECD(4) + INVYR(4) + CYCLE(2) + SUBCYCLE(2) + UNITCD(2) + COUNTYCD(3) + PLOT(5).
Plot_CN	83	TreeInit_cond.Plot_CN = PLOT.CN	244	TreeInit_plot.Plot_CN = PLOT.CN
StandPlot_CN	84	TreeInit_cond.StandPlot_CN=concatenation of TreeInit_cond.Stand_CN+"_"+TREE.SUBPLOT	247	TreeInit_plot.StandPlot_CN=concatenation of TreeInit_plot.Stand_CN+"_"+TREE.SUBPLOT
StandPlot_ID	140	TreeInit_cond.StandPlot_ID=concatenation of TreeInit_cond.Stand_ID+"_"+TREE.SUBPLOT	248	TreeInit_plot.StandPlot_ID=concatenation of TreeInit_plot.Stand_ID+"_"+TREE.SUBPLOT
Plot_ID	71	IF PLOT.DESIGNCD = 1 and COND.CONDPROP_UNADJ = 1 and PLOT.MACRO_BRKPT_DIA Is Null 'Include design info only for current mapped-plot design that have only one condition THEN For records in TREE TreeInit_cond.Plot_ID = TREE.SUBP For records in SEEDLING TreeInit_cond.PlotID_ID = SEEDLING.SUBP ELSE 'Multi-condition plot and all other designs; treed conditions only For records in TREE TreeInit_cond.Plot_ID = 1 For records in SEEDLING TreeInit_cond.PlotID_ID = 1	249	IF PLOT.DESIGNCD = 1 and PLOT.MACRO_BRKPT_DIA Is Null 'Include design info only for current mapped-plot design that have only one condition THEN For records in TREE TreeInit_plot.Plot_ID = TREE.SUBP For records in SEEDLING TreeInit_plot.PlotID_ID = SEEDLING.SUBP ELSE 'Multi-condition plot and all other designs; treed conditions only For records in TREE TreeInit_plot.Plot_ID = 1 For records in SEEDLING TreeInit_plot.PlotID_ID = 1
Tree_CN	72	For records in TREE TreeInit_cond.Tree_ID = TREE.CN For records in SEEDLING TreeInit_cond.Tree_ID = "S" + SEEDLING.CN	251	For records in TREE TreeInit_plot.Tree_ID = TREE.CN For records in SEEDLING TreeInit_plot.Tree_ID = "S" + SEEDLING.CN
Tag_ID	xxx	Pass null value.	xxx	Pass null value.
Tree_ID	73	For records in TREE TreeInit_cond.Tree_ID = TREE.TREE For records in SEEDLING TreeInit_cond.Tree_ID = 1000000 + (SEEDLING.SPCD*1000) + SEEDLING.SUBPLOT	250	For records in TREE TreeInit_plot.Tree_ID = TREE.TREE For records in SEEDLING TreeInit_plot.Tree_ID = 1000000 + (SEEDLING.SPCD*1000) + SEEDLING.SUBPLOT
Azimuth	141	For records in TREE TreeInit_cond.Azimuth = TREE.AZIMUTH For records in SEEDLING TreeInit_cond.Azimuth = IsNull	252	For records in TREE TreeInit_plot.Azimuth = TREE.AZIMUTH For records in SEEDLING TreeInit_plot.Azimuth = IsNull

Distance	142	For records in TREE TreeInit_cond.Distance = TREE.DISTANCE For records in SEEDLING TreeInit_cond.Distance = IsNull	253	For records in TREE TreeInit_plot.Distance = TREE.DISTANCE For records in SEEDLING TreeInit_plot.Distance = IsNull
Tree_Count	75	For records in TREE If PLOT.DESIGNCD = 1 and COND.CONDPROP_UNADJ = 1 and PLOT.MACRO_BRKPT_DIA Is Null Then TreeInit_cond.Tree_Count = 1 ‘Only use Tree_Count = 1 for DESIGNCD = 1 Else If TREE.TPA_UNADJ>0 ‘Fore rest, use expanded tree counts Then TreeInit_cond.Tree_Count = TREE.TPA_UNADJ/COND.CONDPROP_UNADJ Else If TREE.TPAGROW_UNADJ > 0 Then TreeInit_cond.Tree_Count = TREE.TPAGROW_UNADJ/COND.CONDPROP_UNADJ Else If TreeInit_cond.Tree_Count = TREE.TPAMORT_UNADJ*PLOT.REMPER /COND.CONDPROP_UNADJ Else TreeInit_cond.Tree_Count = null For records in SEEDLING If PLOT.DESIGNCD = 1 and COND.CONDPROP_UNADJ = 1 Then TreeInit_cond.Tree_Count = SEEDLING.TREECOUNT_CALC Else TreeInit_cond.Tree_Count = SEEDLING.TPA_UNADJ/COND.CONDPROP_UNADJ	254	For records in TREE If PLOT.DESIGNCD = 1 and PLOT.MACRO_BRKPT_DIA Is Null Then TreeInit_plot.Tree_Count = 1 ‘Only use Tree_Count = 1 for DESIGNCD = 1 Else If TREE.TPA_UNADJ>0 ‘For rest, use expanded tree counts Then TreeInit_plot.Tree_Count = TREE.TPA_UNADJ Else If TREE.TPAGROW_UNADJ > 0 Then TreeInit_plot.Tree_Count = TREE.TPAGROW_UNADJ Else TreeInit_plot.Tree_Count = TREE.TPAMORT_UNADJ*PLOT.REMPER Else TreeInit_plot.Tree_Count = null For records in SEEDLING If PLOT.DESIGNCD = 1 and PLOT.MACRO_BRKPT_DIA Is Null Then TreeInit_plot.Tree_Count = SEEDLING.TREECOUNT_CALC Else TreeInit_plot.Tree_Count = SEEDLING.TPA_UNADJ
Mort_Agent	143	If TREE.STATUSCD = 2 and TREE.AGENTCD >0 Then TreeInit_cond.Mort_Agent = TREE.AGENTCD Else TreeInit_cond.Mort_Agent Is Null	255	If TREE.STATUSCD = 2 and TREE.AGENTCD >0 Then TreeInit_plot.Mort_Agent = TREE.AGENTCD Else TreeInit_plot.Mort_Agent Is Null
History	144	For records in TREE IF TREE.STATUSCD = 0 FIA tree record should be skipped for FVS translation ELSE IF TREE.STATUSCD = 1 THEN TreeInit_cond.History = 1 ELSE IF TREE.STATUSCD = 2 AND TREE.AGENTCD > 0 AND (TREE.STANDING_DEAD_CD = 0 OR TREE.STANDING_DEAD_CD = 1 OR TREE.STANDING_DEAD_CD = IsNull) THEN TreeInit_cond.History = 6 ELSE IF TREE.STATUSCD = 3 or (TREE.STATUSCD = 2 AND TREE.AGENTCD = IsNull AND (TREE.STANDING_DEAD_CD = 1 OR TREE.STANDING_DEAD_CD = IsNull)) THEN TreeInit_cond.History = 8 ELSE TreeInit_cond.History = Is Null For records in SEEDLING TreeInit_cond.History = 1	256	For records in TREE IF TREE.STATUSCD = 0 FIA tree record should be skipped for FVS translation ELSE IF TREE.STATUSCD = 1 THEN TreeInit_plot.History = 1 ELSE IF TREE.STATUSCD = 2 AND TREE.AGENTCD > 0 AND (TREE.STANDING_DEAD_CD = 0 OR TREE.STANDING_DEAD_CD = 1 OR TREE.STANDING_DEAD_CD = IsNull) THEN TreeInit_plot.History = 6 ELSE IF TREE.STATUSCD = 3 or (TREE.STATUSCD = 2 AND TREE.AGENTCD = IsNull AND (TREE.STANDING_DEAD_CD = 1 OR TREE.STANDING_DEAD_CD = IsNull)) THEN TreeInit_plot.History = 8 ELSE TreeInit_plot.History = Is Null For records in SEEDLING TreeInit_plot.History = 1
Species	74	For records in TREE TreeInit_cond.Species = TREE.SPCD For records in SEEDLING TreeInit_cond.Species = SEEDLING.SPCD	257	For records in TREE TreeInit_plot.Species = TREE.SPCD For records in SEEDLING TreeInit_plot.Species = SEEDLING.SPCD
Diameter	76	For records in TREE (all trees 1.0 inches and larger) IF TREE.DIA = not null THEN TreeInit_cond.Diameter = TREE.DIA ELSEIF TREE.DIACALC = not null THEN TreeInit_cond.Diameter = TREE.DIACALC ELSEIF TREE.PREVDIA = not null THEN TreeInit_cond.Diameter = TREE.PREVDIA ELSE TreeInit_cond.Diameter = Is Null For records in SEEDLING TreeInit_cond.Diameter = 0.1 ‘(seedlings don’t have recorded diameters)	258	For records in TREE (all trees 1.0 inches and larger) IF TREE.DIA = not null THEN TreeInit_plot.Diameter = TREE.DIA ELSEIF TREE.DIACALC = not null THEN TreeInit_plot.Diameter = TREE.DIACALC ELSEIF TREE.PREVDIA = not null THEN TreeInit_plot.Diameter = TREE.PREVDIA ELSE TreeInit_plot.Diameter = Is Null For records in SEEDLING TreeInit_plot.Diameter = 0.1 ‘(seedlings don’t have recorded diameters)
Diameter_HtCD	145	For records in TREE TreeInit_cond.Diameter_Ht = TREE.DIAHTCD	259	For records in TREE TreeInit_plot.Diameter_Ht = TREE.DIAHTCD

		For records in SEEDLING TreeInit_cond.Diameter_Ht = IsNull		For records in SEEDLING TreeInit_plot.Diameter_Ht = IsNull
DG	146	For records in TREE IF PLOT.REMPER>=1 AND TREE.STATUSCD = 1 AND TREE.PREVDIA > 0 AND TREE.DIA > TREE.PREVDIA THEN TreeInit_cond.DG= TREE.PREVDIA ‘we’re only including trees with positive growth ELSE TreeInit_cond.DG = IsNull For records in SEEDLING TreeInit_cond.DG = IsNull	260	For records in TREE IF PLOT.REMPER>=1 AND TREE.STATUSCD = 1 AND TREE.PREVDIA > 0 AND TREE.DIA > TREE.PREVDIA THEN TreeInit DG= TREE.PREVDIA ‘we’re only including trees with positive growth ELSE TreeInit_plot DG = IsNull For records in SEEDLING TreeInit_plot DG = IsNull
Ht	147	For records in TREE TreeInit_cond.Ht = TREE.HT For records in SEEDLING TreeInit_cond.Ht = IsNull	261	For records in TREE TreeInit_plot.Ht = TREE.HT For records in SEEDLING TreeInit.Ht_plot = IsNull
HtG	77	For records in TREE IF TREE.PREV_TRE_CN Not Null AND PLOT.REMPER>0 AND TREE.STATUSCD = 1 ‘do we have all parts? THEN ‘yes, then compare T1 and T2 hts Using current TREE.CN as T2 record and TREE.PREV_TRE_CN as TREE.CN for T1 record Get T1 ACTUAL_HT and T2 ACTUAL_HT HeightGrowth = T2 ACTUAL_HT – T1 ACTUAL_HT ‘need to test for +/- IF HeightGrowth > 0 ‘only populate positive values THEN TreeInit_cond.HtG = T1 ACTUAL_HT ELSE TreeInit_cond.HtG = IsNull ELSE TreeInit_cond.HtG = IsNull ‘let any other case fall though as null For records in SEEDLING TreeInit_cond.HtG = IsNull	262	For records in TREE IF TREE.PREV_TRE_CN Not Null AND PLOT.REMPER>0 AND TREE.STATUSCD = 1 ‘do we have all parts? THEN ‘yes, then compare T1 and T2 hts Using current TREE.CN as T2 record and TREE.PREV_TRE_CN as TREE.CN for T1 record Get T1 ACTUAL_HT and T2 ACTUAL_HT HeightGrowth = T2 ACTUAL_HT – T1 ACTUAL_HT ‘need to test for +/- IF HeightGrowth > 0 ‘only populate positive values THEN TreeInit_plot.HtG = T1 ACTUAL_HT ELSE TreeInit_plot.HtG = IsNull ELSE TreeInit_plot.HtG = IsNull ‘let any other case fall though as null For records in SEEDLING TreeInit_plot.HtG = IsNull
HtTopK	148	For records in TREE IF (TREE.HT - TREE.ACTUALHT) > 0 THEN TreeInit_cond.HtTopK = TREE.ACTUALHT ELSE TreeInit_cond.HtTopK = IsNull For records in SEEDLING TreeInit_cond.HtTopK = IsNull	263	For records in TREE IF (TREE.HT - TREE.ACTUALHT) > 0 THEN TreeInit_plot.HtTopK = TREE.ACTUALHT ELSE TreeInit_plot.HtTopK = IsNull For records in SEEDLING TreeInit_plot.HtTopK = IsNull
HtToCrownBase	149	If TREE.ACTUALHT >0 and TREE.UNCRCD>0 Then TreeInit_cond.HtToCrownBase = (TREE.ACTUALHT-(TREE.UNCRCD/100*TREE.ACTUALHT)) Else TreeInit_cond.HtToCrownBase is null	264	If TREE.ACTUALHT >0 and TREE.UNCRCD>0 Then TreeInit_plot.HtToCrownBase = (TREE.ACTUALHT-(TREE.UNCRCD/100*TREE.ACTUALHT)) Else TreeInit_plot.HtToCrownBase is null
CrRatio	150	If TREE.CR >0 Then TreeInit_cond.CrRatio = TREE.CR Else TreeInit_cond.CrRatio is null	265	If TREE.CR >0 Then TreeInit_plot.CrRatio = TREE.CR Else TreeInit_plot.CrRatio is null
UnCrRatio	151	If TREE.UNCRCD >0 Then TreeInit_cond.CrRatio = TREE.UNCRCD Else TreeInit_cond.CrRatio is null	266	If TREE.UNCRCD >0 Then TreeInit_plot.CrRatio = TREE.UNCRCD Else TreeInit_plot.CrRatio is null
CrClass	152	For records in TREE TreeInit_cond.TreeInit.CrClass = TREE.CCLCD For records in TREE TreeInit_cond.TreeInit.CrClass = IsNull	xxx	For records in TREE TreeInit_plot.CrClass = TREE.CCLCD For records in TREE TreeInit_plot.CrClass = IsNull
Damage1 Severity1	153	For records in TREE IF TREE.DAMAGE_AGENT_CD1 = 90001 ‘broken/missing top THEN TreeInit_cond.Damage1 = 96 ELSEIF TREE.DAMAGE_AGENT_CD1 = 90002 ‘dead top THEN TreeInit_cond.Damage1 = 97	267	For records in TREE IF TREE.DAMAGE_AGENT_CD1 = 90001 ‘broken/missing top THEN TreeInit_plot.Damage1 = 96 ELSEIF TREE.DAMAGE_AGENT_CD1 = 90002 ‘dead top THEN TreeInit_plot.Damage1 = 97

		ELSEIF TREE.DAMAGE_AGENT_CD1 = 20015 OR TREE.DAMAGE_AGENT_CD1 = 20017 OR TREE.DAMAGE_AGENT_CD1 = 20023 ‘dwarf mistletoe, but not DF-specific THEN IF (SURVEY.RSCD = 22 OR SURVEY.RSCD = 26 OR SURVEY.RSCD = 27) AND TREE.MIST_CLS_CD>0 ‘use actual DMR in TREE.MIST_CLS_CD in RMRS and ‘PNW THEN IF TREE.SPCD=108 ‘ use specific code for lodgepole pine THEN TreeInit_cond.Damage1 = 31 TreeInit_cond.Severity1 = TREE.MIST_CL_CD ELSEIF TREE.SPCD = 73 ‘ use specific code for western larch THEN TreeInit_cond.Damage1 = 32 TreeInit_cond.Severity1 = TREE.MIST_CL_CD ELSEIF TREE.SPCD = 202 ‘ use specific code for Douglas-fir by species+agent THEN TreeInit_cond.Damage1 = 33 TreeInit_cond.Severity1 = TREE.MIST_CL_CD ELSEIF TREE.SPCD = 122 ‘ use specific code for ponderosa pine THEN TreeInit_cond.Damage1 = 34 TreeInit_cond.Severity1 = TREE.MIST_CL_CD ELSE TreeInit_cond.Damage1 = 30 ‘use generic dwarf mistletoe damage for other species TreeInit_cond.Severity1 = TREE.MIST_CL_CD ELSE TreeInit_cond.Damage1 = 30 ‘if no MIST_CL_CD use damage threshold DMR of 3 TreeInit_cond.Severity1 = 3 ELSEIF TREE.DAMAGE_AGENT_CD1 = 20011 ‘ use specific code for Douglas-fir by agent THEN TreeInit_cond.Damage1 = 33 IF TREE.MIST_CL_CD >0 THEN TreeInit_cond.Severity1 = TREE.MIST_CL_CD ELSE TreeInit_cond.Severity1 = 3 ELSEIF TREE.DAMAGE_AGENT_CD1 = 11000 ‘general bark beetle THEN TreeInit_cond.Damage1 = 01 TreeInit_cond.Severity1 = 03 ELSEIF TREE.DAMAGE_AGENT_CD1 = 11003 OR TREE.DAMAGE_AGENT_CD1 = 11009 OR TREE.DAMAGE_AGENT_CD1 = 11011 OR TREE.DAMAGE_AGENT_CD1 = 11012 ‘other bark beetles not included below, coded as ‘general in FVS THEN TreeInit_cond.Damage1 = 01 TreeInit_cond.Severity1 = 03 ELSEIF TREE.DAMAGE_AGENT_CD1 = 11006 ‘mountain pine beetle THEN TreeInit_cond.Damage1 = 02 TreeInit_cond.Severity1 = 03 ELSEIF TREE.DAMAGE_AGENT_CD1 = 11030 ‘Ips engraver beetle THEN TreeInit_cond.Damage1 = 06 TreeInit_cond.Severity1 = 03 ELSEIF TREE.DAMAGE_AGENT_CD1 = 12040 ‘western spruce budworm THEN TreeInit_cond.Damage1 = 011 TreeInit_cond.Severity1 = 01 ELSEIF TREE.DAMAGE_AGENT_CD1 = 26000 ‘general stem rusts THEN TreeInit_cond.Damage1 = 36 TreeInit_cond.Severity1 = IsNull ELSEIF TREE.DAMAGE_AGENT_CD1 = 26001 OR TREE.DAMAGE_AGENT_CD1 = 26002 OR TREE.DAMAGE_AGENT_CD1 = 26004 OR TREE.DAMAGE_AGENT_CD1 = 26006 OR TREE.DAMAGE_AGENT_CD1 = 26009 ‘specific stem rusts, coded as general for FVS THEN TreeInit_cond.Damage1 = 36 TreeInit_cond.Severity1 = IsNull ELSEIF TREE.DAMAGE_AGENT_CD1 = 21000 ‘general root/butt disease THEN TreeInit_cond.Damage1 = 60 TreeInit_cond.Severity1 = 02 ELSEIF TREE.DAMAGE_AGENT_CD1 = 21008 OR TREE.DAMAGE_AGENT_CD1 = 21014 OR TREE.DAMAGE_AGENT_CD1 = 21015 OR TREE.DAMAGE_AGENT_CD1 = 21016 OR TREE.DAMAGE_AGENT_CD1 = 21018 OR TREE.DAMAGE_AGENT_CD1 = 21019 OR TREE.DAMAGE_AGENT_CD1 = 21020 ‘specific root/butt disease not listed below THEN TreeInit_cond.Damage1 = 60 TreeInit_cond.Severity1 = 02 ELSEIF TREE.DAMAGE_AGENT_CD1 = 21001 ‘Armillaria root disease THEN TreeInit_cond.Damage1 = 61		ELSEIF TREE.DAMAGE_AGENT_CD1 = 20015 OR TREE.DAMAGE_AGENT_CD1 = 20017 OR TREE.DAMAGE_AGENT_CD1 = 20023 OR ‘dwarf mistletoe, but not DF-specific THEN IF (SURVEY.RSCD = 22 OR SURVEY.RSCD = 26 OR SURVEY.RSCD = 27) AND TREE.MIST_CLS_CD>0 ‘use actual DMR in TREE.MIST_CLS_CD in RMRS and ‘PNW THEN IF TREE.SPCD=108 ‘ use specific code for lodgepole pine THEN TreeInit_plot.Damage1 = 31 TreeInit_plot.Severity1 = TREE.MIST_CL_CD ELSEIF TREE.SPCD = 73 ‘ use specific code for western larch THEN TreeInit_plot.Damage1 = 32 TreeInit_plot.Severity1 = TREE.MIST_CL_CD ELSEIF TREE.SPCD = 202 ‘ use specific code for Douglas-fir by species+agent THEN TreeInit_plot.Damage1 = 33 TreeInit_plot.Severity1 = TREE.MIST_CL_CD ELSEIF TREE.SPCD = 122 ‘ use specific code for ponderosa pine THEN TreeInit_plot.Damage1 = 34 TreeInit_plot.Severity1 = TREE.MIST_CL_CD ELSE TreeInit_plot.Damage1 = 30 ‘use generic dwarf mistletoe damage for other species TreeInit_plot.Severity1 = TREE.MIST_CL_CD ELSE TreeInit_plot.Damage1 = 30 ‘if no MIST_CL_CD use damage threshold DMR of 3 TreeInit_plot.Severity1 = 3 ELSEIF TREE.DAMAGE_AGENT_CD1 = 20011 ‘ use specific code for Douglas-fir by agent THEN TreeInit_plot.Damage1 = 33 IF TREE.MIST_CL_CD >0 THEN TreeInit_plot.Severity1 = TREE.MIST_CL_CD ELSE TreeInit_plot.Severity1 = 3 ELSEIF TREE.DAMAGE_AGENT_CD1 = 11000 ‘general bark beetle THEN TreeInit_plot.Damage1 = 01 TreeInit_plot.Severity1 = 03 ELSEIF TREE.DAMAGE_AGENT_CD1 = 11003 OR TREE.DAMAGE_AGENT_CD1 = 11009 OR TREE.DAMAGE_AGENT_CD1 = 11011 OR TREE.DAMAGE_AGENT_CD1 = 11012 ‘other bark beetles not included below, coded as ‘general in FVS THEN TreeInit_plot.Damage1 = 01 TreeInit_plot.Severity1 = 03 ELSEIF TREE.DAMAGE_AGENT_CD1 = 11006 ‘mountain pine beetle THEN TreeInit_plot.Damage1 = 02 TreeInit_plot.Severity1 = 03 ELSEIF TREE.DAMAGE_AGENT_CD1 = 11030 ‘Ips engraver beetle THEN TreeInit_plot.Damage1 = 06 TreeInit_plot.Severity1 = 03 ELSEIF TREE.DAMAGE_AGENT_CD1 = 12040 ‘western spruce budworm THEN TreeInit_plot.Damage1 = 011 TreeInit_plot.Severity1 = 01 ELSEIF TREE.DAMAGE_AGENT_CD1 = 26000 ‘general stem rusts THEN TreeInit_plot.Damage1 = 36 TreeInit_plot.Severity1 = IsNull ELSEIF TREE.DAMAGE_AGENT_CD1 = 26001 OR TREE.DAMAGE_AGENT_CD1 = 26002 OR TREE.DAMAGE_AGENT_CD1 = 26004 OR TREE.DAMAGE_AGENT_CD1 = 26006 OR TREE.DAMAGE_AGENT_CD1 = 26009 ‘specific stem rusts, coded as general for FVS THEN TreeInit_plot.Damage1 = 36 TreeInit_plot.Severity1 = IsNull ELSEIF TREE.DAMAGE_AGENT_CD1 = 21000 ‘general root/butt disease THEN TreeInit_plot.Damage1 = 60 TreeInit_plot.Severity1 = 02 ELSEIF TREE.DAMAGE_AGENT_CD1 = 21008 OR TREE.DAMAGE_AGENT_CD1 = 21014 OR TREE.DAMAGE_AGENT_CD1 = 21015 OR TREE.DAMAGE_AGENT_CD1 = 21016 OR TREE.DAMAGE_AGENT_CD1 = 21018 OR TREE.DAMAGE_AGENT_CD1 = 21019 OR TREE.DAMAGE_AGENT_CD1 = 21020 ‘specific root/butt disease not listed below THEN TreeInit_plot.Damage1 = 60 TreeInit_plot.Severity1 = 02 ELSEIF TREE.DAMAGE_AGENT_CD1 = 21001 ‘Armillaria root disease THEN TreeInit_plot.Damage1 = 61

		TreeInit_cond.Severity1 =02 ELSEIF TREE.DAMAGE_AGENT_CD1 = 21017 ‘Phellinus/laminated root rot THEN TreeInit_cond.Damage1 = 62 TreeInit_cond.Severity1 =02 ELSEIF TREE.DAMAGE_AGENT_CD1 = 21010 ‘Annosus/heterobasidion root disease THEN TreeInit_cond.Damage1 = 64 TreeInit_cond.Severity1 =02 ELSE TreeInit_cond.Damage1 = IsNull ‘records with no dam1 values TreeInit_cond.Severity1 = IsNull ENDIF For records in SEEDLING TreeInit_cond.Damage1 = IsNull TreeInit_cond.Severity1 = IsNull		TreeInit_plot.Severity1 =02 ELSEIF TREE.DAMAGE_AGENT_CD1 = 21017 ‘Phellinus/laminated root rot THEN TreeInit_plot.Damage1 = 62 TreeInit_plot.Severity1 =02 ELSEIF TREE.DAMAGE_AGENT_CD1 = 21010 ‘Annosus/heterobasidion root disease THEN TreeInit_plot.Damage1 = 64 TreeInit_plot.Severity1 =02 ELSE TreeInit_plot.Damage1 = IsNull ‘records with no dam1 values TreeInit_plot.Severity1 = IsNull ENDIF For records in SEEDLING TreeInit_plot.Damage1 = IsNull TreeInit_plot.Severity1 = IsNull
Damage2 Severity2	154	See Damage1 and Severity1	268	See Damage1 and Severity1
Damage3 Severity3	155	See Damage1 and Severity1	269	See Damage1 and Severity1
Defect_Cubic	156	TreeInit_plot.DefectCubic = TREE.CULLCF ‘value is percentage of gross cubic-ft volume, as used in damage coding	270	TreeInit_plot.DefectCubic = TREE.CULLCF ‘value is percentage of gross cubic-ft volume, as used in damage coding
Defect_Board	157	TreeInit_plot.DefectBoard = TREE.CULLBF ‘value is percentage of gross board-ft volume, as used in damage coding	271	TreeInit_plot.DefectBoard = TREE.CULLBF ‘value is percentage of gross board-ft volume, as used in damage coding
TreeValue	158	Pass null value.	272	Pass null value.
Prescription	159	Pass null value. User supplied.	273	Pass null value. User supplied.
Age	160	For records in TREE IF TREE.BHAGE > 0 THEN TreeInit_plot.Age = TREE.BHAGE ELSE IF TREE.TOTAGE >0 THEN TreeInit_plot.Age = TREE.TOTAGE ELSE TreeInit_plot.Age = IsNull For records in SEEDLING TreeInit_plot.Age = IsNull	274	For records in TREE IF TREE.BHAGE > 0 THEN TreeInit_plot.Age = TREE.BHAGE ELSE IF TREE.TOTAGE >0 THEN TreeInit_plot.Age = TREE.TOTAGE ELSE TreeInit_plot.Age = IsNull For records in SEEDLING TreeInit_plot.Age = IsNull
Slope	161	TreeInit_plot.Slope = SUBPLOT.SLOPE	275	TreeInit_plot.Slope = SUBPLOT.SLOPE
Aspect	162	TreeInit_plot.Aspect = SUBPLOT.ASPECT	276	TreeInit_plot.Aspect = SUBPLOT.ASPECT
PV_Code	163	Pass null value	277	Pass null value
PV_Ref_Code	164	Pass null value	278	Pass null value
TopoCode	165	Pass null value	279	Pass null value
SitePrep	166	Pass null value.	280	Pass null value.

Appendix C. FIA Inventories by State, Year, and Type

STATECD	State name	Periodic Inventory Years	Start of Annual Inventory
1	Alabama	1972, 1982, 1990	2000
2	Alaska	1998, 20032004	2004
4	Arizona	1985, 1999	2001
5	Arkansas	1978, 1988, 1995	2000
6	California	1994	2001
8	Colorado	1984	2002
9	Connecticut	1985, 1998	2003
10	Delaware	1986, 1999	2004
12	Florida	1970, 1980, 1987, 1995	2002
13	Georgia	1972, 1982, 1989	1997
15	Hawaii	-[1]	2010
16	Idaho	1991	2004
17	Illinois	1985, 1998	2001
18	Indiana	1986, 1998	1999
19	Iowa	1990	1999
20	Kansas	1981, 1994	2001
21	Kentucky	1988	2000
22	Louisiana	1974, 1984, 1991	2001
23	Maine	1995	1999
24	Maryland	1986, 1999	2004
25	Massachusetts	1985, 1998	2003
26	Michigan	1980, 1993	2000
27	Minnesota	1977, 1990	1999
28	Mississippi	1977, 1987, 1994	2006
29	Missouri	1989	1999
30	Montana	1989	2003
31	Nebraska	1983, 1994	2001
32	Nevada	1989	2004[2]
33	New Hampshire	1983, 1997	2002
34	New Jersey	1987, 1999	2004
35	New Mexico	1987, 1999	2005[3]
36	New York	1993	2002
37	North Carolina	1984, 1990	2002
38	North Dakota	1980, 1995	2001
39	Ohio	1991	2001
40	Oklahoma	1989 (central/west), 1976, 1986, 1993 (east)	2008 (east), 2009 (west)
41	Oregon	1999	2001
42	Pennsylvania	1989	2000
44	Rhode Island	1985, 1998	2003
45	South Carolina	1968, 1978, 1986, 1993	1999
46	South Dakota	1980, 1995	2001
47	Tennessee	1980, 1989	1999
48	Texas	1975, 1986, 1992	2001 (east), 2004 (west)
49	Utah	1993	2000
50	Vermont	1983, 1997	2003
51	Virginia	1977, 1985, 1992	1998
53	Washington	1991, 2001	2002
54	West Virginia	1989, 2000	2004
55	Wisconsin	1983, 1996	2000
56	Wyoming	1984, 2000	2011
60	American Samoa	-[1]	2001
66	Guam	-[1]	2002
70	Palau	-[1]	2003
72	Puerto Rico	-[1]	2001
78	US Virgin Islands	-[1]	2004

[1]Periodic inventories were not conducted.

[2]Due to insufficient funding, annual inventory ceased after 2005. Sampling resumed in 2010 including plots that would have been measured in inventory years (INVYR) 2006-2009. Therefore, measurement year (MEASYR) is frequently different from INVYR.

[3]Annual inventory sampling began in 2008. Due to the State of New Mexico receiving The American Recovery and Reinvestment Act of 2009 (ARRA) money, sampling was accelerated beginning in 2010 and broadened to include plots that would have been surveyed had the inventory started in 2005. Therefore, measurement year (MEASYR) is frequently different from inventory year (INVYR).