Compilation Subteam Notes
8/19/11
Attendees: Renate Bush, Harry Reed, Mike Shanta, Marcus McElhaney, Skeeter, Carol Apple, Marcus McElhaney, Rhonda Matheson, Carlos Ramirez

Notes on Compilation requirements document:
- Red is new text added to the previous outline
- Green is new text added by Renate that needs to be “approved” by compilation team. This is generally added prior to scheduled meetings to start discussion.
- Notes will build on documentation from the last time
- Blue are ticklers to Renate of things she needs to follow up on.
- Yellow highlights need to be discussed and decided by core team at next core team call
- Green highlights have to do with dataset needs and functionality

Background Information
This document was created by members of the DATIM core team and Compilation subteam including Renate Bush, Skeeter Czimskey, Jim Alegria, Harry Reed, Marcus McElhaney, JD Zeiler, Carlos Ramirez, Chip Scott, Don Vandendriesche, Susan Hua, Carol Apple, and Mike Shanta. It was created from bi-monthly conference calls, notes from Las Vegas (Use Scenario: Forest Plan Monitoring, see section 4), from “Virtual Portland Meeting” notes from 11/2/10, and Requirement 1.2 from Software Requirements Specification for the Analytical Tool for inventory and monitoring.

Compilation Functions:
- Plot Data
- Pop compiler
- Dataset compilation

Compilation Tool needs to accommodate:
- Creating a list of plots (done in ETL)
- Interfacing with Regional QA/QC Tools to ensure data is loaded as expected in FSVeg
- Selecting plots to move from FSVeg to the datamart
- Adds additional values
- Track which calculations are used by each Region
- Track which calculations have been applied to a plot (or list of plots)
- Allow for defining new attributes/calculations
- Keeping track of meta data associated with an attribute
- Reporting capabilities for the above functionality

Integration with SIT needs to be done for
- Ensuring that every spatial location has associated data in FSVeg that can be compiled
- Creating datasets
- Tying datasets to estimation units
Compilation Storyboard Flow

1. Authenticate user to ensure that they can compile data. Compilation can only be done by someone that has specific privileges within the Region.

2. Choose Region
   a. This screen chooses the Regional suite of compilation algorithms and tools used to process the data.
   b. This suite can be added to.

3. Audit data (Not part of Compilation but expected in order for ETL and compilation to work)
   a. Ensure that data is complete in FSVeg. QA/QC of the “raw data” in FSVeg happens prior to the ETL.
      i. Goal is to ensure that data in FSVeg is clean and complete so that other NFS applications can use it and will not produce results inconsistent with DATIM.
      ii. Work with FIA/FSVeg liaison to ensure that all data of interest has been loaded into FSVeg.
      iii. If data appears to be loaded in error or some data is missing, work with FIA/FSVeg liaison to fix errors.
      iv. If an error with the FSVeg data is found during the ETL process that is caused by an error in the loader from NIMS to FSVeg or is because data was missing in NIMS at time of migration then will need to come back to this step once data has been reloaded from NIMS.
      v. Would like to have a corporate solution to the QA/QC for NFS that works for all data stored in FSVeg
         1. Look into DARSI
         2. Independent auditing tool that can run on both NIMS and FSVeg
      vi. Best case scenario would be to provide a tool which works off of all data stored in FSVeg and not just FIA data.
   b. Open spatial location and geographic extent info from SIT – discuss further.
   c. Tie FSVeg CN to spatial location – discuss further.
      i. Each spatial location has associated data in FSVeg, for every fsveg cn, there is a unique feature within geodatabase.

4. Identify data to compile
   a. Select plots to compile based on a list.
      i. Want to be able to store and manage the list.
      ii. Need to store the list of cn’s and the list name.
   b. There will be functionality within the compiler to build the list which will be integrated with the GI and SIT.
      i. There will be functionality to build the list within the geodatabase.

5. Invoke ETL tool
   a. QA/QC data in DATIM. Three classes of errors that may be discovered in the ETL data output (this is directly copied from DATIM ETL discussion document).
      i. A bug in the extractor or loader, e.g. “is all the data present that I asked for”.

b. Open spatial location and geographic extent info from SIT – discuss further.


1. DATIM Developers will create code to ensure that all data from FSVeg is moved to DATIM so that the Regional DATIM Stewards will not take on additional responsibilities.
   a. This kind of bug can be detected in code via ad-hoc data integrity checks. Such tests might include comparing the number of records transferred against the expected number of records, summing condition values in both FSVeg and DATIM and comparing the results, etc. These tests could be automated but require programming and are open-ended in their scope.

2. If this error is found, Users need to work with DATIM developers to resolve and start ETL step over.
   ii. A transformation error in the way that computed fields are calculated or the way new attributes are created.
      1. Right now unsure if some calculations/transformations will occur within the ETL. – developer’s (Marcus) will let us know
         a. It may be that translations “non-math” items may best be translated here. If so, will need the ability to create a new attribute and tie it to a cross-walk somewhere.
      2. RB: Need an example of this type of bug since ETL should not be doing transformations or compilations that I know of
      3. This kind of bug is difficult to detect programmatically but can be fixed in the compilation system when identified and tested.
      4. If this error is found, Users need to work with DATIM developers to resolve and start ETL step over.

iii. Incorrect data, i.e. a typo or wrong data entered during data collection.
    1. Bugs of this nature are difficult to find programmatically and are usually detected visually by a user.
    2. If a DATIM user detects an error in the DATIM datamart, that person cannot change the data in the datamart, needs to contact the person in the Region with the role to compile and edit data.
    3. When the power user changes data in the datamart, the user must explain why the change was made. DATIM keeps a record of:
       a. The power user’s name or ID
       b. The power user’s explanation for the change
       c. The change date and time
       d. The CNs of the changed record(s)
       e. The original value and the changed value
       f. This will be stored and available to all users
4. Updates need to be made in FSVeg to keep data consistent. Need to notify FIA/FSVeg liaison to make sure data in FSVeg is properly updated.

b. If the problem is with the extractor/loader or a transformation, then data needs to be re-ETLled and process needs to start over.

6. Process data
   a. Import Data from DATIM database
   b. Ensure that all attributes needed are defined. If attributes are not defined, see Define a new attribute section
   c. Run Data Thru FVS (Calculate attributes using FVS)

   d. Calculate attributes using Regional algorithms
      i. This is for newly acquired data
      ii. An example might be to fill in values that have been subsampled on GST trees.
   e. Review current attributes that are available to your Region.
      i. List them
      ii. Indicate which are final, which are draft
      iii. Check back,
          1. Do you want to add more?
          2. Do you want to use something from a different region?
   f. Calculate (or spatially assign) attributes using Regional Algorithms
   g. Populate Database

7. Defining datasets

8. Data is now ready to use for analysis within ATIM
   a. 
      1. Define a New Attribute. Requirements for creating an Attribute. An attribute can NOT be used in ATIM unless it is defined.
a. Name of attribute *-full name of attribute
   i. Heuristic search to try to keep similar items with similar attribute names
   ii. Primary use is column name, row name, page name within a report
   iii. This column is called XXX in the meta data database table
   iv. Suggested attribute name provided by SIT but can be edited. Needs to be worked out by SIT/compiler group.

b. Attribute column name-* is the name of a the column in the data center table
   i. Want to use a suffix, _R1
   ii. Recommend using suffix of NFS Region (R1, R2), FIA Region (IW, PNW, etc), national FIA (FIA), NFS (NFS), FVS (FVS).
   iii. Attribute rules: What is the length? No wild card characters.
   iv. Heuristic search to try to keep similar items named the same
   v. This column is called YYY in the meta data database table
   vi. Cannot name a column with a name that has already been used
   vii. Does the datacenter have standards that we need to adhere to? Harry will follow up on this.
   viii. Suggested column name would be provided by SIT but can be edited. Needs to be worked out by SIT/compiler group.

c. Description *
   i. Paired down version of the definition
   ii. Short name = long label
   iii. May be useful for titles or graph axis titles, (for report headers) –
   iv. Suggested description would be provided by SIT but can be updated by the user. Needs to be worked out by SIT/compiler group.

d. Format (e.g., number, text) *
   i. Format for attribute would be autofilled for attribute populated spatially.
   Could be edited. Needs to be worked out by SIT/compiler group.

e. Definition –*
   i. Could be hyperlink, GTR Reference, or text
   ii. could make this part of the intersection if we established metadata guidelines for the registered datasets so that the definition could be retrieved from the spatial dataset. SIT group will come up with suggestions that are discussed with the core team.

f. Units (drop-down) *
   i. Will have a look-up table for standard units
   ii. We feel that this list needs to be managed by a small group of people with the ability of users to request new units
   iii. Can add additional units through the form
   iv. Are there standard units we should be adhering to? Fill in without a pick list, then see how different we are, clean up and go forward.
   v. could make this part of the intersection with the ability to edit, if needed

g. Themes (SILV, Wildlife, insect/disease, fire/fuels, tree, carbon/biomass)
   i. drop-down
   ii. A blank theme indicates bring up under ALL in the Atim reports
   iii. Themes are a way for a user to see a pick list of attributes of interest for reporting.
iv. An attribute can have multiple themes
v. Chip will compare this list of themes to broad objectives currently in DTIM. Broad Objectives are: Forest Health (insect and disease), Forest Productivity (silviculture), Forest Fire Effects (fire and fuels), Ecosystem Restoration, Effects of Invasives, Biological Diversity, Wildlife Habitat (wildlife), carbon sequestration, watershed health
vi. We feel that this list needs to be managed by a small group of people with the ability of users to request new themes
vii. SIT tool would allow you to check themes at time of intersection, if appropriate

h. Citation –
   i. Citation of the publication
   ii. could make this part of the intersection if we established metadata guidelines for the registered datasets so that the definition could be retrieved from the spatial dataset. Needs to be worked out by SIT/compiler group
i. Information about where attribute is appropriate to use
   i. Not sure how to do this. Is it Regions, states, if it is just a text list, then cannot really automate
   ii. Need to provide guidance in how to fill this out.
   iii. could make this part of the intersection if we established metadata guidelines for the registered datasets so that the definition could be retrieved from the spatial dataset. Ensure that this is in the metadata.

j. Level of calculation/summarization *: Setting (plot)/plot-level (subplot)/condition
   i. Can be one, two, or all
   ii. Can be used for filtering plots
   iii. For attributes that used for plot selection for analysis, then the attribute should be stored.
   iv. Calculate and store derived variables using the field data. This is either done once or on the fly with a user-supplied routine (see REQ-1.4). Must be able to store the results so that the calculations can be validated, and reused for efficiency.
   v. Need to indicate what level we want information summarized to. Summarize field and derived data to Subplot/Condition and Plot levels. Caution should be used when summarizing metrics since plots are often not in a single condition. Metrics, such as basal area/ac, can be expanded to the subplot or condition level based on their portion of the plot or based on their contribution to the plot total.
   vi. Currently, FVS derived attributes cannot be summarized to condition level
   vii. SIT tool could make this part of the intersection

k. Pre-conditions
   i. If the attribute is dependent upon other attributes being populated
   ii. This would help to create a trigger indicating if a pre-condition attribute has changed that potentially this attribute needs to be recalculated
iii. Although how preconditions will be supported over time has not been determined, would like a system that at least identifies existing attributes that are used to calculate the attribute.

iv. SIT tool could make this part of the intersection. Metadata, if well-defined should have this in it.

v. Need overarching pre-conditions and pre-conditions associated with FIA, intensified grid, and stand exam data.

vi. For attributes created in FVS, need to make sure that all scenarios are accounted for.

1. How will plots that don’t have trees be attributed? FVS will drop them from the runstream, what value should be loaded into DATIM?
2. Will plots with no data collected, for example non-forest be treated differently than plots that have no trees sampled on the plot but are forested?

l. An archive flag if no longer used
   i. a pointer to the new attribute if it has been redone?
   ii. If we made updates to a layer that had already been intersected, then archive the old one and use the newer version. For spatial intersection information, if spatial data is already registered and the registered data layer has been updated then automatically archive.

m. Archive date

n. If the attribute is Draft or final
   i. (whoever creates can move to final after QA/QC process is finished)
   ii. A check mark when it moves to final indicates that all users can use the attribute. A null value indicates that only users with certain privileges can “see” this attribute.
   iii. No protection for overwriting values if draft. Final cannot overwrite.
   iv. For spatial intersection information, if spatial data is already registered and the registered data layer has been updated then automatically draft

o. Date created
   i. Date that attribute was defined
   ii. This is auto-filled based on when the attribute is defined (which means the required fields (see * items) are filled in)
   iii. Populated for spatially derived attributes

p. Final date
   i. Date that attribute is moved from draft to final
   ii. This is automated when the attribute goes from draft to final

q. Logon user name who created the attribute.
   i. This is automatically populated for tabular and spatial intersection data

r. Source
   i. Drop down list
   ii. Redundant with suffix in the attribute column name
   iii. Recommend using suffix of NFS Region (R1, R2), FIA Region (IW, PNW, etc), national FIA (FIA), NFS (NFS), FVS (FVS).
iv. Appropriate for spatial intersecton data as well. SIT/Compilation could establish metadata standards that can be changed by the user.

v. If source is FVS then need identify the name of the attribute from the output FVS run.
   1. In a pop up window
      a. choose from output table
      b. Name of attribute in FVS output

s. Source control (how you get the source code (if applicable) to talk to system
   i. ATIM user can submit algorithm and if “board” agrees then can be programmed for them
   ii. Devoloper’s will develop webservices and have a framework in place
      1. Need communication rules
      2. Standardized procedures to talk to the algorithms so this is language and system agnostic will be determined by the developers
   iii. Not needed for spatial intersection data

iv. For FVS data,
   1. Need to indicate
      a. kcp file that has this attribute defined
      b. where attribute is stored in output generated by FVS. For example, is it in the output access database? Is it a text file?
      c. name of attribute in FVS output
   2. any additional modifications that are made to FVS output before loading.
      a. How will values that are returned as “-1” be handled? These are settings that do not have the necessary criteria to have the attribute of interest calculated
   3. Need to look at the output from FVS and add additional information/requirements, if needed

   t. Certain fields will be required before an attribute can be “registered” = * above

2. Within package, need to keep track of which compilation packages (ie Regions) are using the algorithm.

3. If the algorithm for Attribute X changes, then:
   a. Need to define new attribute Y (never overwrite existing attribute X)
   b. Could copy metadata info from X to Y (where appropriate)
   c. Be able to automatically populate Attribute Y for any plots that have Attribute X populated.
   d. Check any dependencies that other attributes have with Attribute X and Define a New Attribute based on Attribute X being replaced by Attribute Y. Probably won’t be done now but could be done in phase 2.
   e. Notify all the other Regions that are using Attribute X that a new attribute Y has been defined and why they may want to populate Y
   f. Archive attribute X (if appropriate)

4. Calculating attributes for a set of plots:
   a. Pre-compilation report: this report shows the current attributes that are available within the Regional template to calculate for the plots that have been identified.
      i. Separates draft attributes from enabled attributes.
ii. Breaks attributes out by “code source”, whether it is calculated in FVS or by other means.

iii. Option to show archived attributes, that will not be run

iv. For the list of plots, identifies any of the attributes that have already been calculated for any of the plots.

v. Possible format could be as follows. A check would be in the box for each attribute that had already been calculated for an attribute.

<table>
<thead>
<tr>
<th>Regional attributes</th>
<th>National attrib (such as FIA classifications)</th>
<th>FVS Attrib</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting</td>
<td>Att A</td>
<td>Att B</td>
</tr>
<tr>
<td>cn</td>
<td>Att C</td>
<td>Att A</td>
</tr>
<tr>
<td></td>
<td>Att B</td>
<td>Variant</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Attrib A</td>
</tr>
</tbody>
</table>

vi. Need option to allow user to specify which attributes (or multiple attributes) for a plot (or set of plots) are added or loaded or skipped

5. using FVS:
   a. Ultimately would like to create a system that interfaces with Suppose, when needed to build additional keywords. Would like to have functionality which can take keywords defined and link to attributes in DATIM
   b. Currently plan to modify translator built off of FIADB to create FVS ready data. If need be, can use translator built off of FSVeg and just have the list of cn’s create the FVS ready data during the ETL process.
   c. Need to keep track of version of variant that is being run. If updates are made to FVS versions which effect Attributes stored in DATIM database, then need to figure out how to handle.
      i. This is the role of the Regional Coordinator
   d. Prep
      i. Data in correct form (FVS_data.mdb, FIA2FVS.mdb)
         1. FVS_Standinit
            a. Stocking percent and non-stockable plots. How should this be handled with non-forested/non-sampled. Non-forested/sampled, forested/not-sampled (hazardous)
            b. See word document FIA to FVS_r1_scenarios.docx
         2. FVS_Plotinit
         3. FVS_Treeinit
         4. FVS_Groupaddfilesandkeywords
      ii. Assign Variant
         1. Want flexibility in how variant is assigned
            a. Currently data in FSVeg translated to FVS ready format has variant automatically assigned.
               i. This is based on Region and Forest
            b. Need to be able to have variant assigned consistently with how it is assigned for the same plot in FSVeg
2. Want the ability to intersect plot locations with FVS Variant map (or a further refined Regional/forest/local version of an FVS variant) within SIT
   a. Right now the FVS variant map only returns one variant.
   b. Want to create a feature to use with DATIM Compilation which assigns only 1 variant to each plot.
   c. Need to start with FVS variant map in the GI.
   d. Carlos will schedule a webinar which shows users how to access coverage in GI and submit needed updates for their Region.
   e. Until this variant assignment map is done, will use the first variant assigned in the list coming from FSVegetation.
   f. The variant will be put into the Stand_init and Plot_init table in FVS ready database and will be stored in a table in DATIM.

iii. Kcp files
   1. Compute variables (.kcp file)
      a. Need to be able to save within Regional compilation area
      b. Regional stewards can work with Regional FVS coordinator to create
      c. Keep rules of creating an FVS variable in mind:
         i. Maximum of 8 characters
         ii. FVS does not give any warning if you exceed the 8-character limit but, it only recognizes the first 8 characters. If 2 attributes have the same first 8 characters in a name, then FVS will think the variables are the same and use the last calculated attribute. In the output, only the first 8 characters will be shown.
         iii. Cannot start with a number, but there can be numbers in the string
         iv. Should not contain spaces
         v. Cannot use variables that have been predefined or keywords such as Event Monitor variables or function name
         vi. Can't be a process controller {IF, THEN, ENDF, END}
         vii. Can't contain a mathematical operator, parentheses, or ampersand {+, -, *, /, =, (,), &}
         viii. Can't be a logical operator {AND, OR, NOT, LT, LE, GT, GE, EQ}
         ix. Can't be a math function {SIN, COS, TAN, ARCSIN, ARCCOS, ARCTAN, ABS, ALOG, ALLOG10, EXP, FRAC, INT, MAX, MIN, MOD, SQRT}
2. Any other keywords that are needed to get “good” run (make adjustments)

3. Keywords for base run
   a. Numcycle 1
   b. Treelist output file
   c. etc

iv. Include national “core” FVS attributes in a single kcp file
   1. Tree-level attributes:
      a. Natcruise volumes = NatCr_Volume
      b. Biomass = FVS_Biomass
   2. Canopy cover = FVS_Can_Cov
   3. qmd calculated for trees 1”+ in DBH = FVS_QMD_1+
   4. FFE outputs from the potential fire report

  e. Run FVS
     i. Create outputs for plot(Setting) AND subplot (plot)
     ii. Only 1 variant at a time can run although may have multiple variants needed for the list of plots
         1. Marcus has application that splits database into a separate database for each variant found in the data
     iii. Ensure that all of your needed output has been “created”
     iv. Run any associated post-processors
         1. Such as R9 dominance typing
         2. Execute dominance routines that are maintained in FVS

  f. Audit FVS Output Data
     i. User is sent email with “run report” when run is complete.
     ii. Run report includes:
         1. .err file: this contains any problems found during FVS runtime. This file contains any errors found in the FVS Output file, designated by *** for all the settings in the batch file. This is a good place to look for potential problems.
         2. A comparison of the number of plots in the list and the number of plots actually run through FVS. This is needed because FVS aborts the run if the program crashes. (It does not skip the plot and run the rest of the data).
         3. Ability to go to website and retrieve any of the other FVS generated files:
            a. output files \n            b. base input files
            c. Post-load processor generated outputs
            d. SVS files
            e. etc

  g. Check for errors
     i. Ability to run a subset of plots and look at all outputs to ensure as expected
ii. Make sure that plots that are missing in FVS output are accounted for. This is because FVS drops plots that don’t have trees. Need to specify if missing values have a null value, 0, or something else.

h. Load data into DATIM database
   i. Load for plots and subplots
   ii. Field map to cross the FVS attributes to the Datamart fields
   iii. Account for plots that were water, not sampled and therefore not run through FVS. These cases need to be addressed when defining an attribute.

**Bin items to discuss:**

1. A single estimation unit may have a one to many relationship with data sets. For example, an estimation unit may belong to both a regional data set and a national data set. Somehow, we need to keep track of all of the attributes available for a data point. It is important to recognize that defining the extent and content (in terms of inventories) of the data sets will dictate the attributes available for the user. Only attributes that are in common across all of the inventories that are the building blocks of the data sets will be seen as options in the drop-down lists in ATIM. So need to decide how to set these up.

   • Link the estimation units to the datasets. (SIT/Compilation)

   Need a linked, hierarchical system tree structure.

   Won’t be implemented in phase 1.

   Needs to be added to database.

   Be able to have a many to many relationship between estimation units to datasets.

   Need to be able to define the vector of attributes associated with each data point to create each dataset associated with each (or many) estimate. (This is actually said down below)

   • Be able to select a collection of attributes that are assigned to datasets (Compilation)

   Be able to have multiple sets of attributes assigned to datasets.

   • Be able to associate the collection of attributes to the subplot center

All one meta data table or separate tables depending upon which goes together?

   • Make sure we connect an attribute with a table.

Ongoing discussion of how Compiler integrates with

   • SIT
   • ATIM
   • What else?

When associating an Attribute with a plot:

   • Date it was run, by who

Comments on the existing storyboard:

   • Bottom 2 choices Wing it & ? need to be moved to post-process section

Need to have existing compilations added to the database (spreadsheet?) that Marcus has created
• Gary Brand needs to add the information from the proxy datamart.

Ability to manage users, roles and permissions within the DATIM schema based on your role. This is similar to what was developed for MIDAS. The MIDAS schema currently has 5 MIDAS user IDs at the datacenter which start with the prefix of MIDAS: MIDAS, MIDASADMIN, MIDASCREWLEADER, et. al. Each has a different set of permissions in the database. At login, the software maps the user's login name to one of these database user IDs, and the software interacts with the database using the mapped database userID, not the user's login name. This way, login user names are managed just like other software data and do not require special setup vis a vis a help desk or the data center, etc. (Marcus, Skeeter)

Next Meeting:
Friday, May 13, 1-2:30 mountain
888-930-9770
Participant: 3145547
Appendix A:

ATIM storyboards (to have an idea how the attributes created during compilation may be used in reports)
Appendix B:

Possible attributes to compile. Table until someone stewards them through Bin for now until there is a need or equations are easily available:

Soils
- Universal Soil Loss Equation
- Etc.

Lichens
- Species richness
- Etc.

Vegetation/Invasive Plants
- Species Richness
- Etc.

Ground cover
- Litter layer carbon
- Etc.
Notes from meeting with Mike VanDyck, Forest Vegetation Simulator Program Manager, Forest Management Service Center.

8/5/2011

Notes from Mike VanDyck, FVS Discussion Topics:

- Discuss current data translation software for FVS and support/maintenance plan for the various translators you maintain (or work with FSVeg to maintain)
  - FIA2FVS was developed by Don Vandendriesche. This is still being worked on querying capabilities to select which plots to use. Currently processes for an entire state and is slow. Ultimately this will be handed off to FIA for maintenance but FIA and FVS contact are not decided upon. This was funded by FIA.
  - FSVeg Utility called FSVeg DBlInk. This is maintained by the FSVeg Staff. Roy Mita is the contact. Previously, Don worked with Roy. Chad Keyser will be the contact in the future.
  - Updating existing software when modifications are needed.
    - For the FSVeg utility, FVS staff compiles a list of needs and works with FSVeg to get additions into next release (this takes up to 8 months lead time)

- Current FVS computing capabilities (I am thinking the stand and the plot)
  - FVS has the ability to process a stand and individual plots within a stand using the Plot_init table. For FIA, this works for both Plot and Subplot.
  - Are there plans to enhance translation software to divide data by condition
    - There are currently no plans to create input data by condition. If that is a need, then it needs to be communicated to the FVS staff and it will be put on their priority list. Updates for input data would need to be made to both the FIA and the FSVeg translators in order to use this functionality when and if it is done.
    - Marcus asked couldn’t you just create input data by condition and run them as stands?
      - Yes, you could divide plots by condition. See below, a “stand” could be created from plots 1 and 2 and a different “stand” from plots 3 and 4.

- What if a subplot has more than 1 condition?
- Functionality of how subplots with more than 1 condition would be handled needs to be determined
- Example of subplots with more than one condition is below. Because TPA expansions have already been done, FVS does not need the expansions. However, if you only use a portion of the trees on a supplot, then FVS will assume that there is less competition on the subplot.

- Just because it could be done, need to think through all of ramifications
  - Decided that FVS attributes will not be available at the Condition Level at this time.
  - Questions about current FVS functionality:
    - What are the specifications for naming a computed attribute (are there format limitations? Special characters?)
      - FVS was built around the following conventions:
        - Maximum of 8 characters
          - FVS does not give any warning if you exceed the 8-character limit but, it only recognizes the first 8 characters. If 2 attributes have the same first 8 characters in a name, then FVS will think the variables are the same and use the last calculated attribute. In the output, only the first 8 characters will be shown.
        - Cannot start with a number, but there can be numbers in the string
        - Should not contain spaces
        - Cannot use variables that have been predefined or keywords such as Event Monitor variables or function name
        - Can’t be a process controller {IF, THEN, ENDF, END}
        - Can’t contain a mathematical operator, parentheses, or ampersand {+, -, *, /, =, (, ), &}
        - Can’t be a logical operator {AND, OR, NOT, LT, LE, GT, GE EQ}
        - Can’t be a math function {SIN, COS, TAN, ARCSIN, ARCCOS, ARCTAN, ABS, ALOG, ALOG10, EXP, FRAC, INT, MAX, MIN, MOD, SQRT}
      - Mike recommends using just number, letters, underscore
Mike noted that there is not extensive documentation discussing this currently but he is going to add it to the Essential FVS Guide.

- How are the FVS variants versioned? Do all variants have the same version number? Are previous executables stored?
  - FVS recently changed how variants are versioned. It corresponds to the version number assigned in the code repository.
  - Previous executables are not stored but FVS can build past versions.
    - Will the DATIM group be archiving previous versions?
      - Yes, we did decide to save the versions of FVS but this will need to be tackled by the Administrative group
  - When FVS releases new software (about every 2-3 months), it sends out a bulletin. Bulletins document all of the changes. These bulletins are also stored on the FVS website. Anyone can get an FVS release email. Contact FVS staff. Right now the FVS contact for each NFS region gets the bulletins.
    - When changes are made that affect a Region’s attributes then the administrative group will figure out how a coordinator is notified.
    - The Regional data steward has final say about whether data is updated.
    - Keeping up with changes in FVS and how they affect Regional attributes in DATIM is up to the Regional data steward.

- How does FVS handle plots within a stand with no trees?
  - In the absence of trees, FVS will include the subplots in the averages.

- How does FVS handle stands within a runstream with no trees?
  - The text output file will have an error message about “too few trees for the projection”. This will also show up in the .err file. However, in the outputs such as database or spreadsheets, these records will be dropped.
  - FVS would run with only dead trees in the input data. The answer is yes. The trees are all reported as mortality trees.

- Would the FSVegetation Spatial Data Analyzer potentially be of use to us?
  - No, only works on stands delineated in FSVegetation Spatial.