Risk Based Wildland Fire Management Analysis Process

The U.S. Department of the Interior (DOI) Risk Based Wildland Fire Management (RBWFM) analysis calculates the expected acres of selected values burned by wildland fires. The intent of the analysis is to estimate the acres of National Life, Property and Resource (NLPR) values impacted by fires originating on DOI land. This provides an estimate for the exposure analysis portion of risk assessment (See Appendix B). The effects portion of the analysis will be left to the strategic business plans created by the DOI bureaus.

The RBWFM analysis project team was charged with developing a national geospatial analysis process. The project team performed significant research and analysis on concepts and technologies that optimize key risk based analysis components. Comments and input were sought and received from Tribes, agencies, bureaus and stakeholders. The Core Analysis Development Team included: William "Bill" Johnson, Craig Thompson and Brent Timothy.

Initial NLPRs were identified by an advisory team made up of representatives from the Bureau of Indian Affairs (BIA), the Bureau of Land Management (BLM), the U.S. Fish and Wildlife Service (FWS), and the National Park Service (NPS), led by the DOI's Office of Wildland Fire (OWF). The NPLRs were presented to the Interior Fire Executive Council (IFEC) by OWF leadership, along with the following guiding principles:

1. Values included in RBWFM analysis will be of national importance across DOI
2. Geospatial data should be nationally consistent from authoritative sources.

Two sets of values were used for the analysis:

- Eight NLPR values were selected for the continental US, Hawaii, and Puerto Rico.
- Seven NLPR values were selected for Alaska

A more detailed description of NLPR values can be found in Appendix A.

Table 1: NLPR Values Continental, Hawaii, and Puerto Rico

1	Developed Areas
2	Infrastructure
3	Recreation Sites
4	Riparian Areas
5	Forest Type Areas
6	Threatened and Endangered Species
7	Surface Drinking Water
8	Sagebrush Steppe Ecosystem

Table 2: NLPR Values Alaska

1	Developed Areas
2	Infrastructure
3	Recreation Sites
4	Native Allotments and Corporations
5	Reindeer Herding Areas
6	Threatened and Endangered Species
7	Surface Drinking Water

NLPR source GIS data layers that were initially vector feature types (point, line, or polygon) were converted to grid features (raster cells) with cell resolution of 270 meters (process steps can be found in Appendix A). NLPR raw GIS data layers that were raster layers were also resampled to 270 square meters. This resolution was selected to match the GIS simulation resolution used for the continental US, Hawaii, and Puerto Rico, which was from the large-fire simulation system, known as FSim (Finney, et al 2011). Alaska values were kept at 270 meters to match the rest of the analysis, although FSim fires in Alaska are at a 1-kilometer resolution.

Results of values burned are summarized to the location where the fire started that had an impact on at least one identified NLPR value. The approach is summarized in Figures 1 and 2. First, simulated perimeters for each fire are analyzed to count the number of cells with NLPRs that burned.

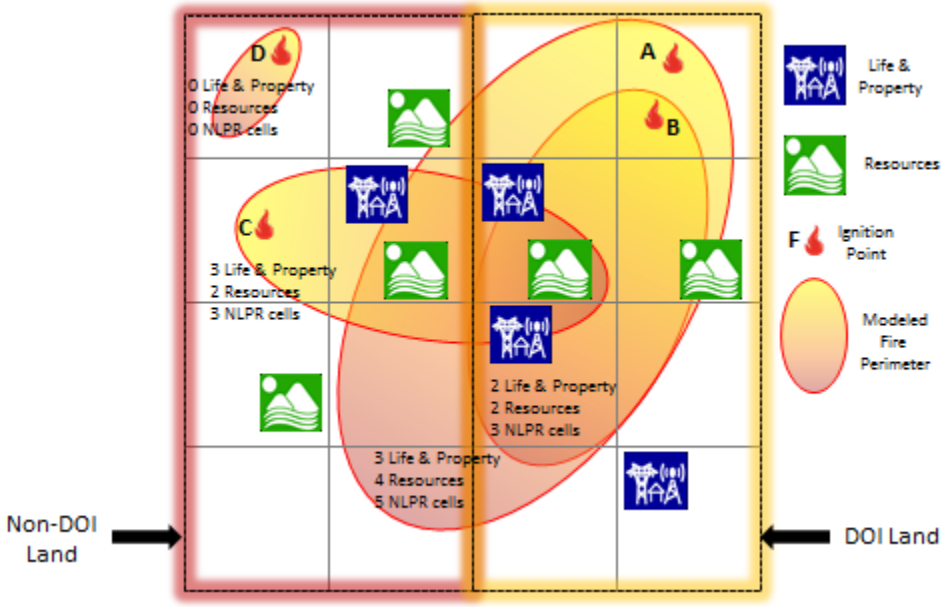


Figure 1: Diagram of counting value cells affected by wildfire

NLPRs are summed in the order shown in Table 1 above. The number of cells with developed areas is counted, and then the number of cells with either developed areas or infrastructure, or both, is counted, etc. If more than one NLPR occupies a cell, the cell is only counted once. For example, in Figure 1, the perimeter for fire A includes 3 life and property cells, and 4 resource cells. Note that it does not matter whether or not the NLPR is on DOI land or not. Two cells inside the perimeter for fire A contain more than one NLPR (both life and property, and resource), each of these cells are only counted once. Three cells contain only one NLPR each, the total cell count for NLPR cells affected by fire A would be five. These counts are then associated to the cell at the ignition point.

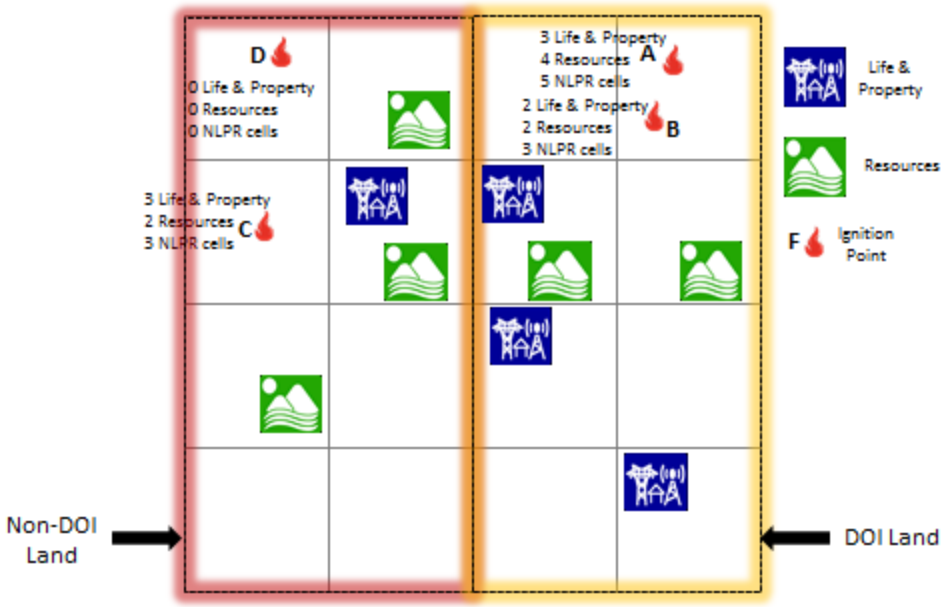


Figure 2: Diagram of attaching cell counts to fire ignition point

Figure 2 illustrates how values are accounted by summing the counts of the cells with NLPR values into the cell with the fire ignitions. The counts for each fire start in a cell are then summed, and divided by the number of modeled years in the FSim run in order to annualize the expected number of cells with NLPRs affected by each fire. In the example above, the counts for fires A and B would be combined to show fires in that cell affect 5 life and property cells, 6 resource cells, and 8 NLPR cells. Assuming 100 modeled years for FSim, this would mean an average of 0.05 life and property cells, 0.06 resource cells, and 0.08 NLPR cells to burn from fires starting in the cell with fires A and B, per year. The total number of cells is then multiplied by the acres per cell to give expected acres of NLPRs burned. Using 270 meter cells would result in 0.9 acres of life and property, 1.1 acres of resource, and 1.4 acres of NLPRs to burn from fires starting in that cell on average per year. Note that fire C would not be counted, the ignition point is not on DOI land, even though it burns NLPRs on DOI land.

Planning and preparation for analysis included the creation of 16 separate tools written in the object oriented scripting language Python (Lutz, M. 1996 and Zanderbergen, P. 2013; operating in an ArcGIS shell (version 10.2.1)). To expedite the data entry process, an Excel spreadsheet is created for each Python script tool. This allows tool commands for each Fire Planning Unit (FPU) to be entered in batch

mode for processing all FPUs within a specific Geographic Area (GA). Excel functions make creating the repetitive lists simple (see figures 3 and 4 for an example). A number of separate additional tools are created for OCONUS areas because DOI lands are in different GIS layers, requiring code to point to the different shapefiles.

<u>The 16 step process follows:</u>

Step 1. Clip NLPR GIS data layers to FPU boundaries with a 15km buffer. The buffer aligns the NLPR's data layer extent of the FSim runs, which also employs a 15 km buffer outside of FPU boundaries. Using buffered FPU boundaries facilitate a manageable processing time, working with smaller elements rather than the entire national extent. Figure 3 illustrates how Excel is used to create the nine inputs to Step 1 of the python script for each FPU.

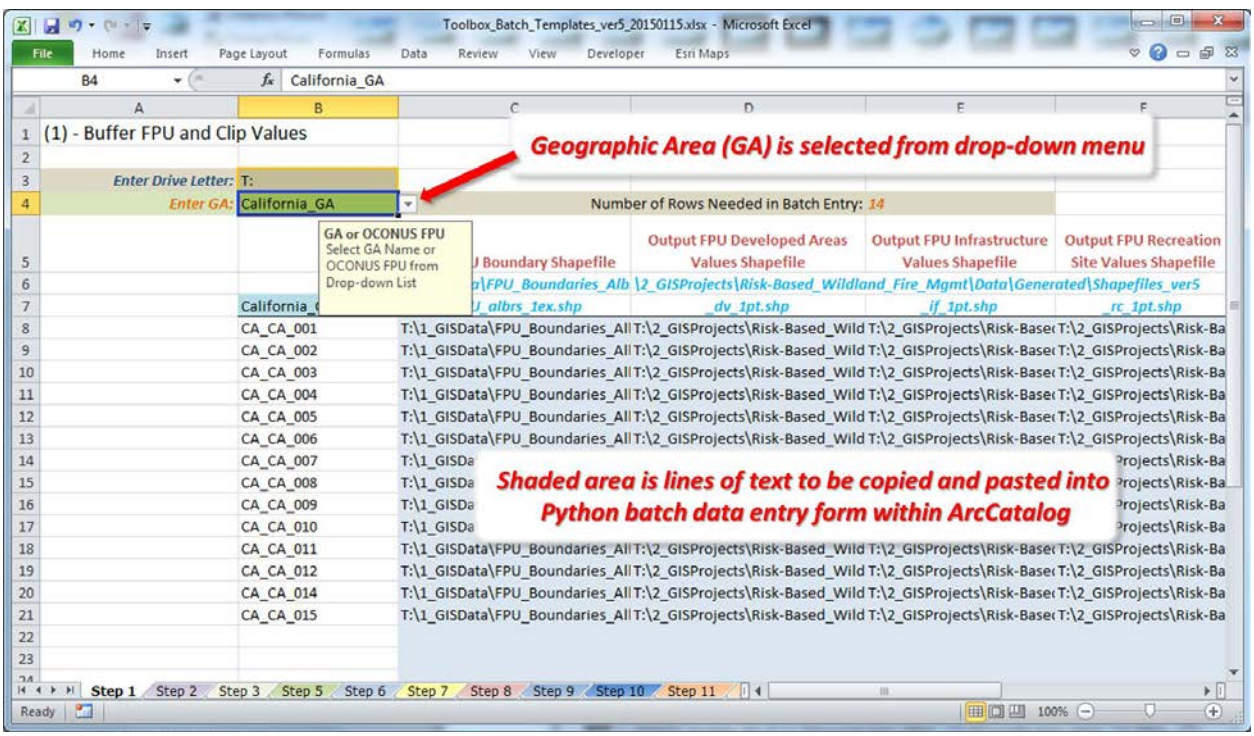
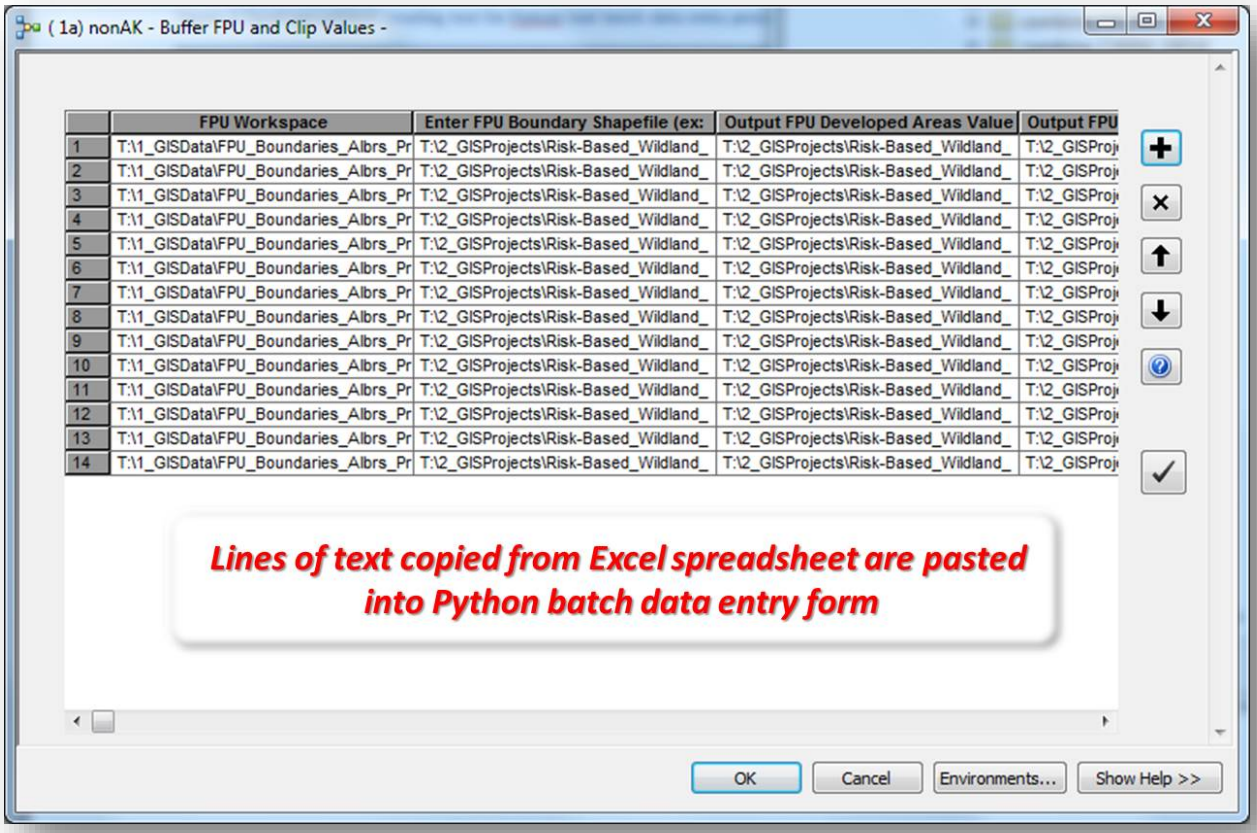
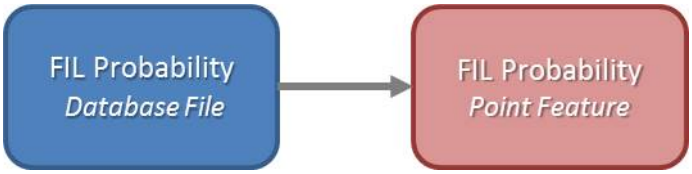


Figure 3: Excel spreadsheet creating text for Python tool batch data entry process Step 1 - Clip NLPRs to buffered FPU extent

Figure 4 illustrates what the batch mode looks like with the values entered. Note: columns are automatically created, but rows must be added manually, and then all rows selected before pasting. (for Python script see Appendix C)



Step 2. Using SAS software (Version 9.3), the results from FSim are converted from text files to dBase (DBF) files for ingestion into ArcGIS. The FSim results includes the X and Y positions of each cell, the probability of the cell burning, and the Fire Intensity Level (FIL) probabilities for each of the six FIL categories. The results are converted into a Point Feature Shapefile through Python tools operating in an ArcGIS shell. (for SAS and Python scripts see Appendix D)



(Note: Included in the programming code, a new value is created for fires above four foot flame lengths by adding the probability for FIL3 through FIL 6 for other options not used in the final RBWFM analysis.)

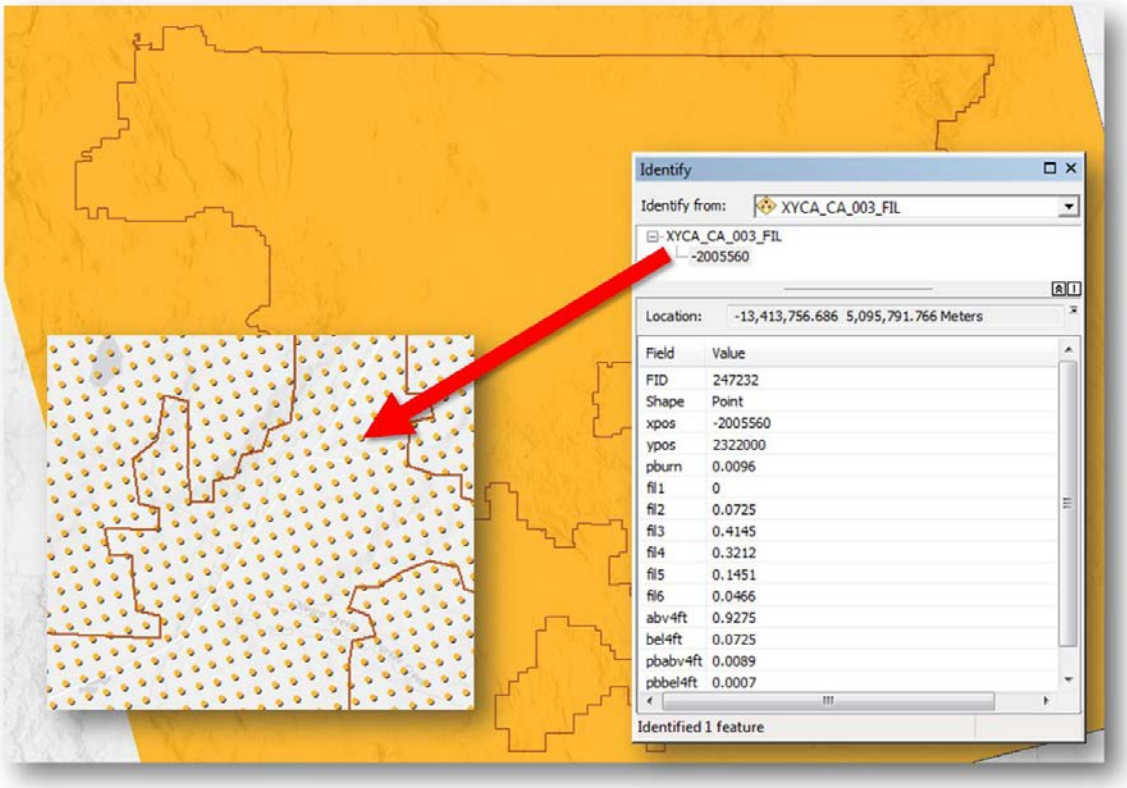


Figure 5: Output of Step 2-Point feature Shapefile (CA_CA_003 FPU red outline) FIL database attributes, with insert detail

Step 3. Perform a Spatial Index and a Spatial Join with the NLPRs and the FIL database attributes. The NLPR point feature attributes are joined with the FIL point feature locations and are written to a new Values/FIL point feature data layer. (for Python script see Appendix E)

(Note: A Spatial Index allows for location of attributes and their relationship with spatial features in a data layer. Spatial Joins connect attributes from one feature to another based on their spatial relationship.)

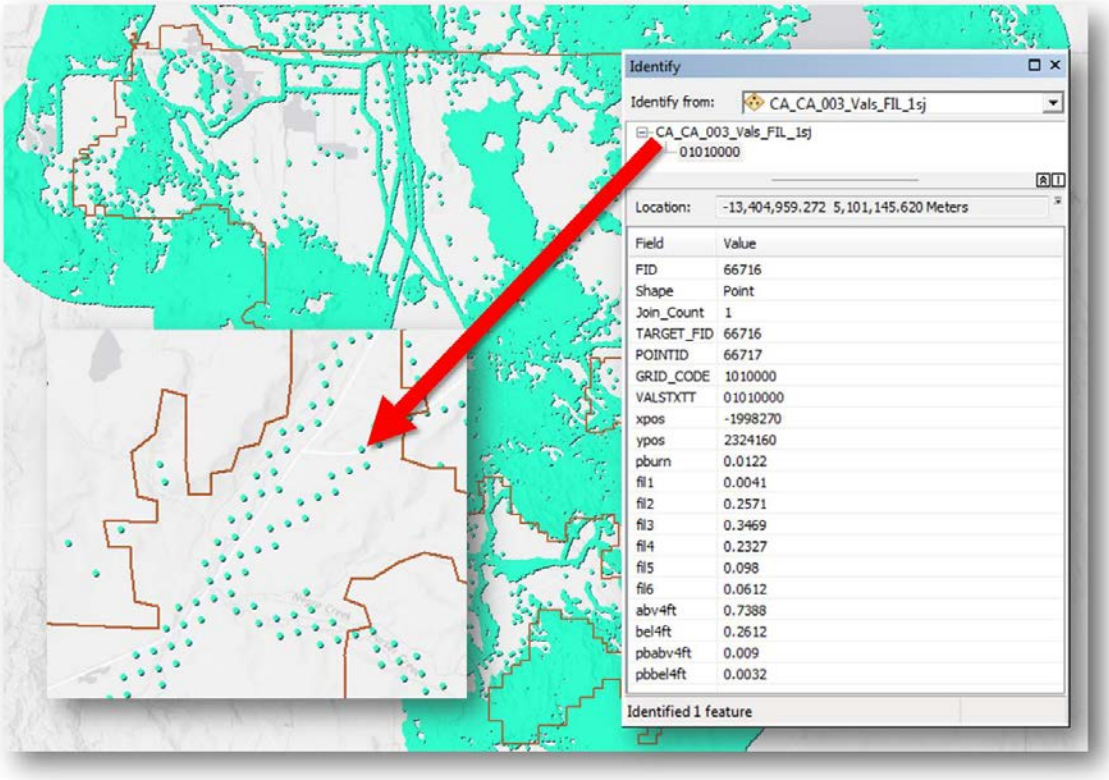
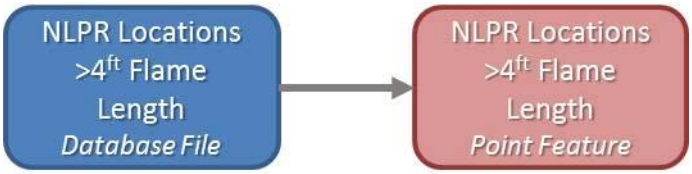


Figure 6: Output of Step 3- Point feature Shapefile (CA_CA_003 FPU with 15 km buffer) containing joined FIL database attributes and NLPR locations, with insert detail

Step 4. Using SAS, each value is put into cumulative order (see Table 1). Cells having an NLPR but did not burn in FSim were removed to reduce the file size. The SAS code also creates the Excel spreadsheet with file locations (path information). This allows the Python tool to ingest the resulting DBF file and create a point feature shapefile (for SAS and Python scripts see Appendix F)



Step 5. A Tabular Intersect is performed to make a relationship connection between each fire perimeter from the FSim analysis and with the NLPR point feature locations, as seen in figures 1 and 2 above. The Tabular Intersect function computes the intersection between two feature classes and cross-tabulates the count of the intersecting features. This allows for a count of the number of NLPRs that are affected by each modeled fire perimeter within the FSim results attributed to the Fire Start Location point. (for Python script see Appendix G)

(Note: This step may take up to 24 hours for one FPU to process)

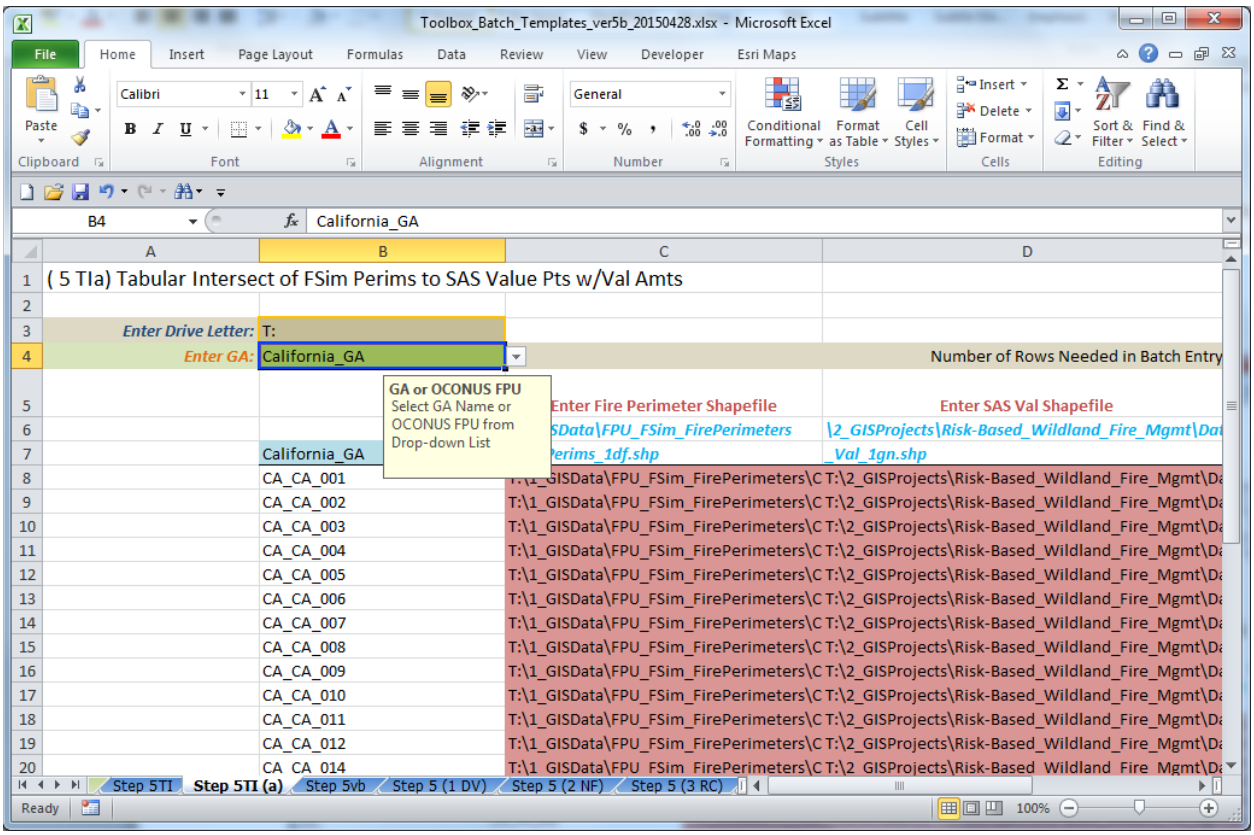


Figure 7: Excel spreadsheet creating text for Python data entry Step 5/6 - Tabular Intersect of FSim modeled Fire Perimeters to NLPR point features. (*Note...Because of the resources required for each calculation, a single FPU was run one at a time through the tool)

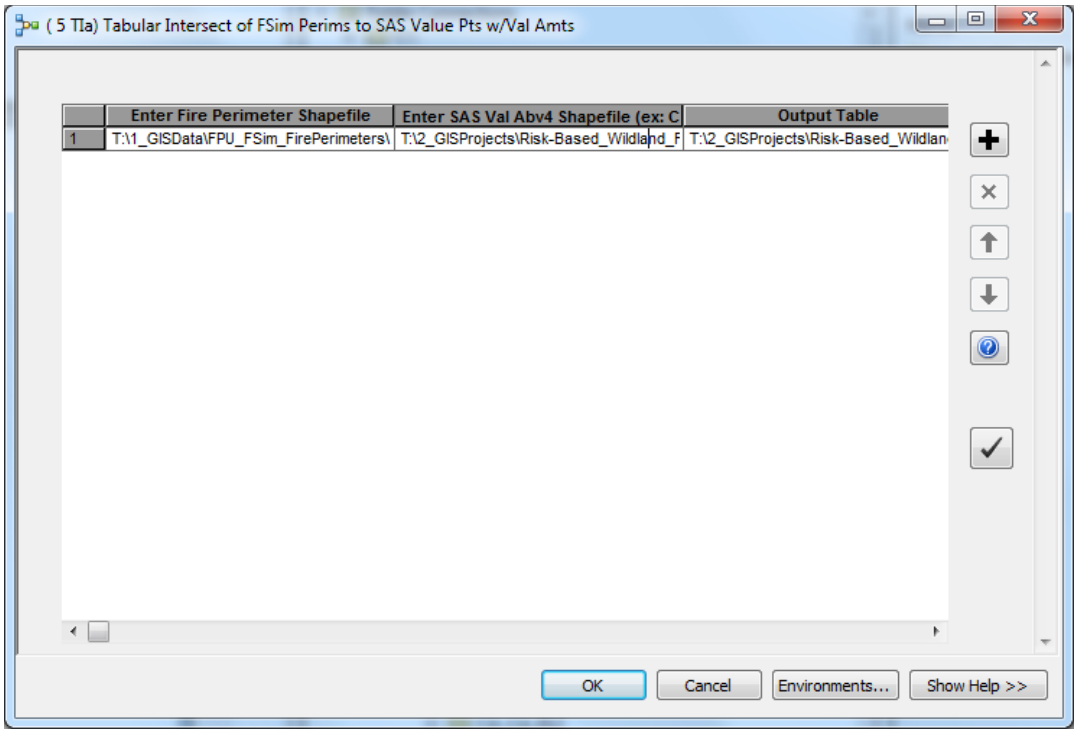


Figure 8: Python tool entry form Step 5/6 - Tabular Intersect of FSim modeled Fire Perimeters and associated Fire Start Location point to NLPR point features (single entry)

Step 6. The number of NLPRs in each modeled fire perimeter is merged into a new point feature shapefile containing the ignition point locations of the modeled fires. (for Python script see Appendix G).

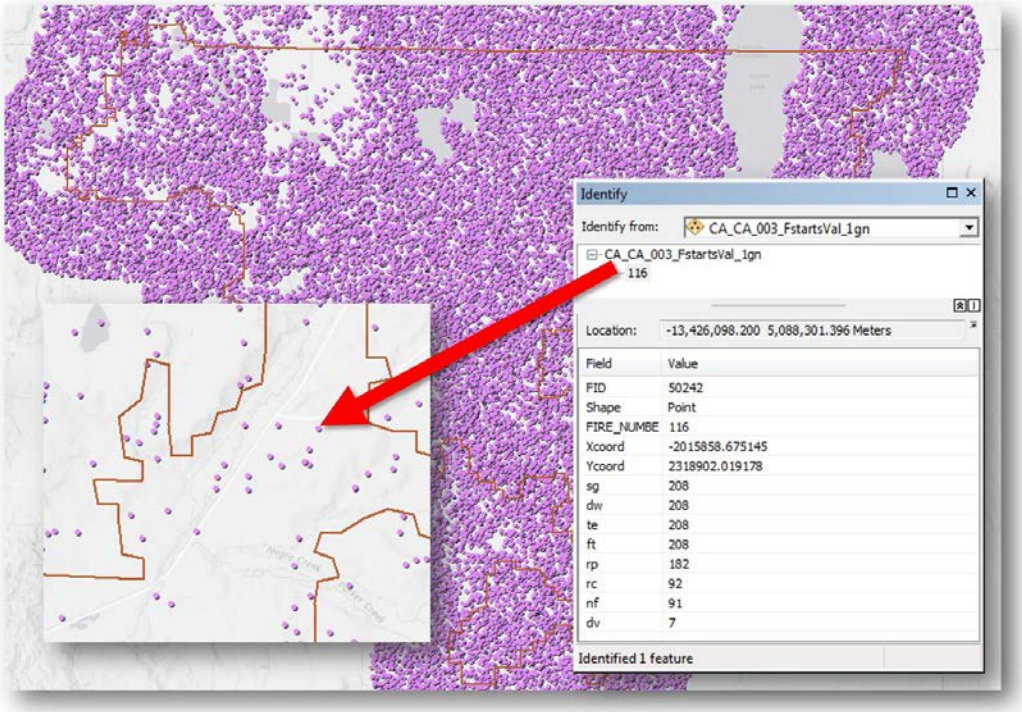
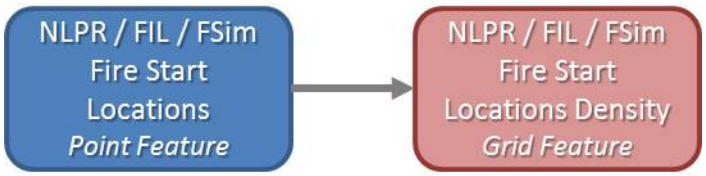
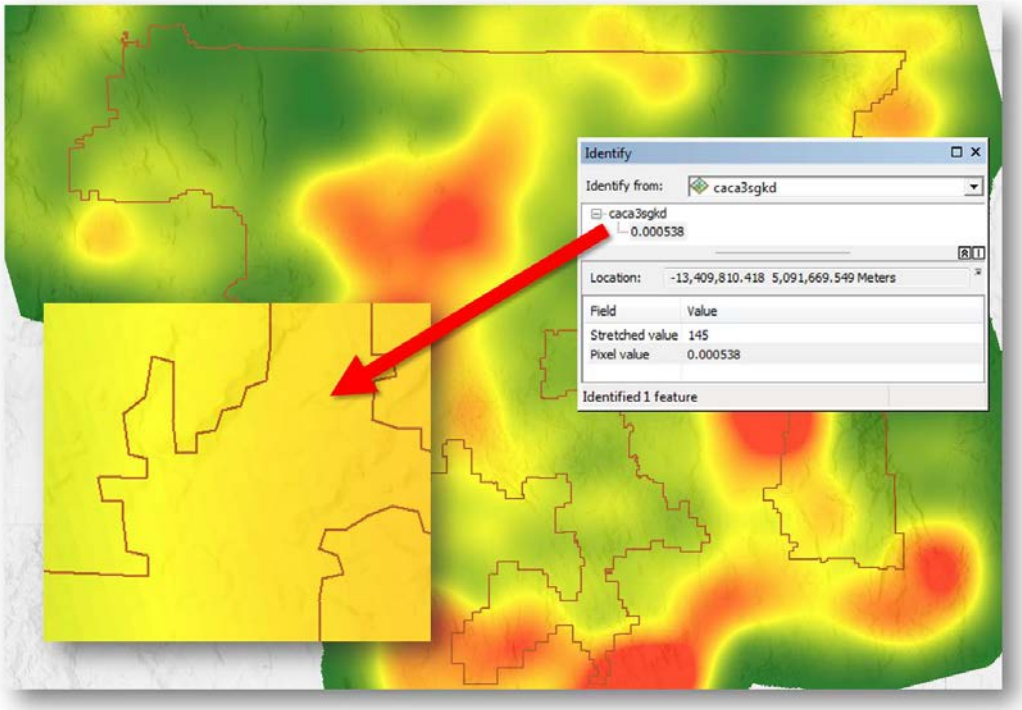


Figure 9: Output for Step 5/6 - Point feature Shapefile (CA_CA_003 FPU with 15 km buffer) displaying modeled Fire Start Location points containing joined FIL database attributes and NLPR occurrences with insert detail

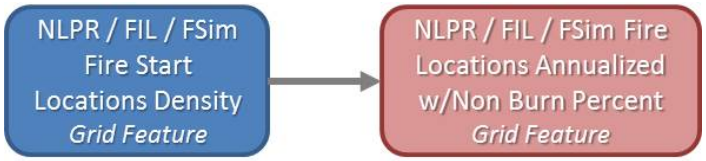
Step 7. Density grid feature data layers are created for each NLPR category with Python script using the Kernel Density command in the ArcGIS shell. (for Python script see Appendix I)



(Note: Kernel Density creates a raster grid data layer by calculating the density of point features around each output raster cell. Conceptually, a smoothly curved surface is fitted over each point, the extent of which is governed by a pre-determined search radius (10 kilometers).)



Step 8. The Density Grid values are annualized based on the number of simulation years used to calculate burn probability within the FSim model. This is accomplished by using the Raster Calculator in the ArcGIS shell through Python script to perform Map Algebra functions.



The map algebra equation used is :

$(KD_{VAL} * Cell_m^2 * Cell_{AC})/ Sim_{YR} * Burn_{PCT} / Burn_{PCT}^{Mn}$

Where:

KD_{VAL} = NLPR Kernel Density Grid Value

$Cell_m^2$ = Number of Meters per Grid Cell (270 meters squared)

$Cell_{AC}$ = Number of Acres in a Grid Cell

Sim_{YR} = Number of Years in the FSim Burn Probability Model run

$Burn_{PCT}$ = Grid Cell's Percent of being Burnable

$Burn_{PCT}^{Mn}$ = Mean of Grid Cell's Percent of being Burnable

(for Python script see Appendix J).

This changes the results from number of NLPR cells affected by fire to the estimate of NLPR acres burned.

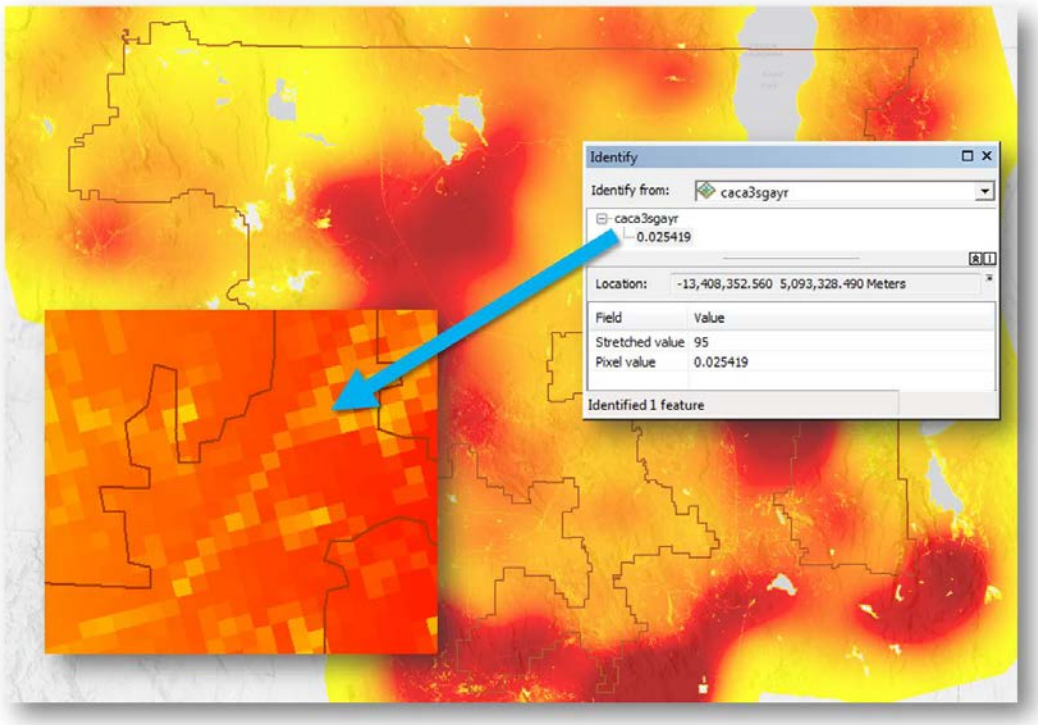


Figure 11: Output of Step 8-raster Grid feature (CA_CA_003 FPU with 15 km buffer) displaying Density Grid of NLPR FSim fire start locations with acres affected per simulation year with regards to Burnable Percent. Sagebrush Steppe NLPR is shown.

Step 9. The annualized density grids are clipped to FPU boundaries so that the 15 Km buffer areas will not be counted multiple times. (for Python script see Appendix K)

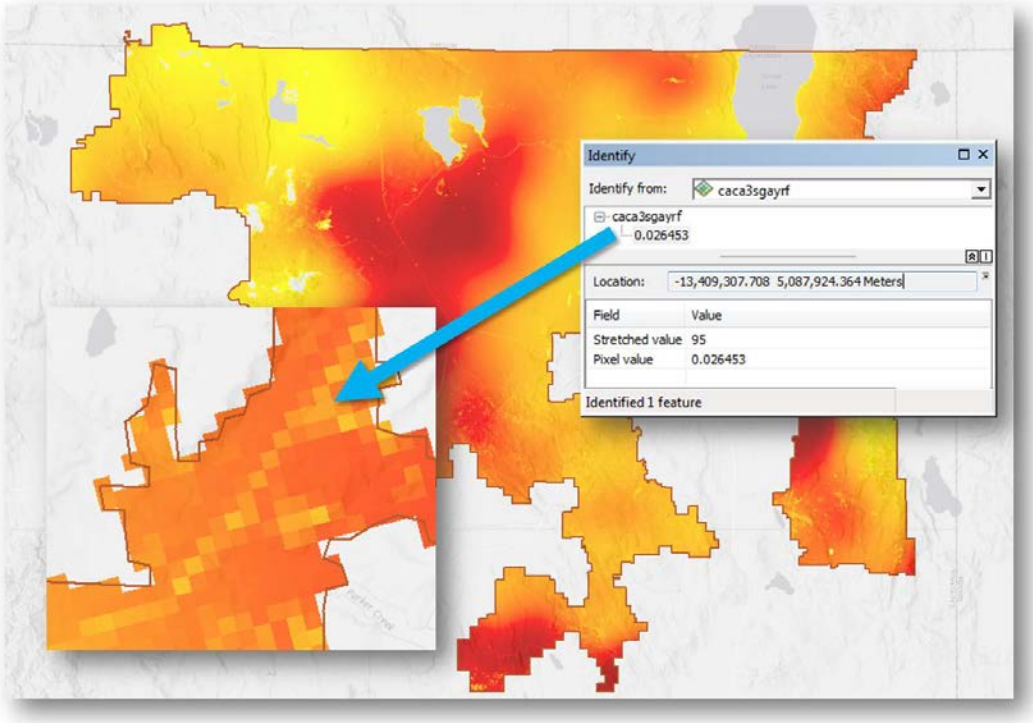
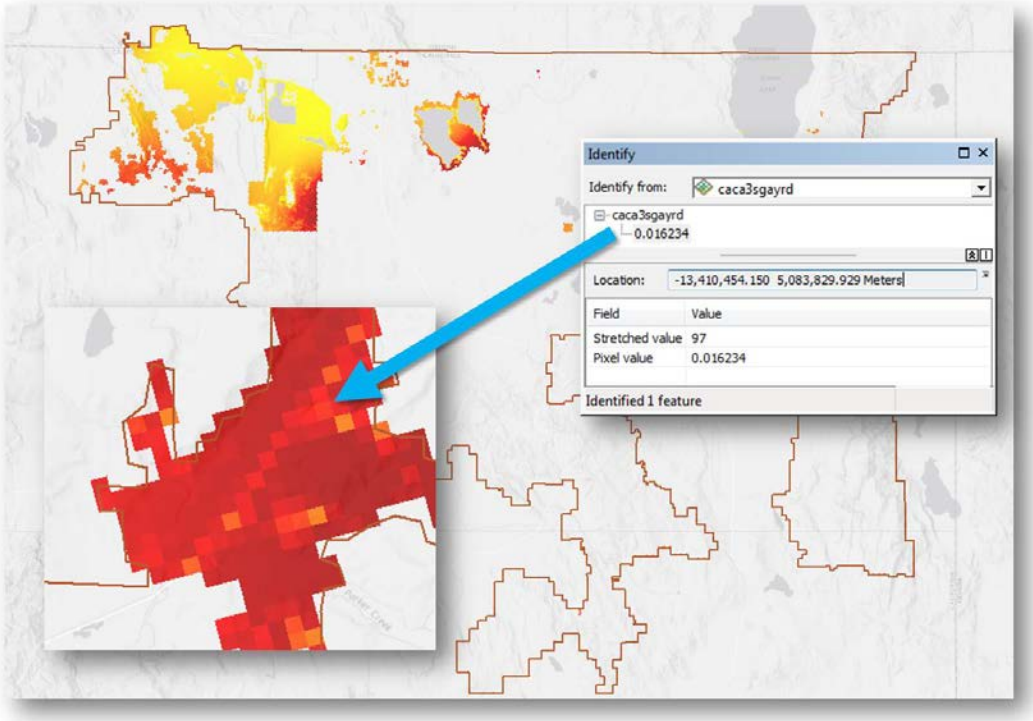
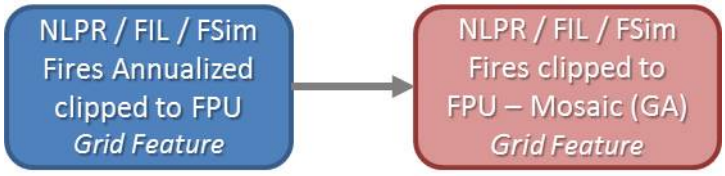


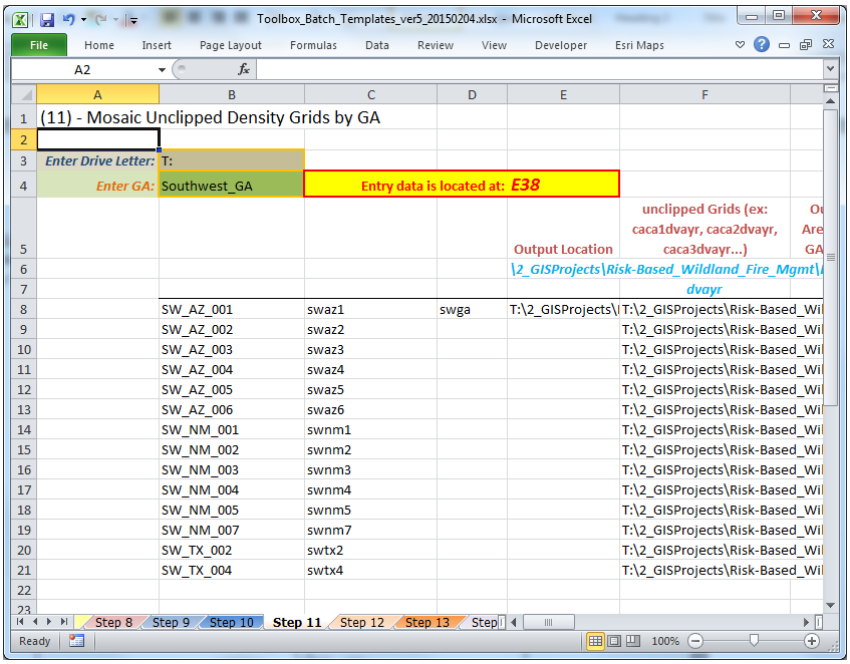
Figure 12: Output of Step 9-raster Grid feature (CA_CA_003 FPU) displaying Density Grid of NLPR FSim fire start locations with acres affected per simulation year in regard to Burnable Percent clipped to FPU Boundary. Developed Areas NLPR is shown.

Step 10. Clip annualized raster data layer Density grids to DOI Bureau boundaries. (for Python script Appendix L)



Step 11. Create a mosaic of output from Step 8 with Geographic Area (GA) extents. The Python script activates the 'Mosaic to New Raster' tool via ArcGIS shell. (for Python script see Appendix M)



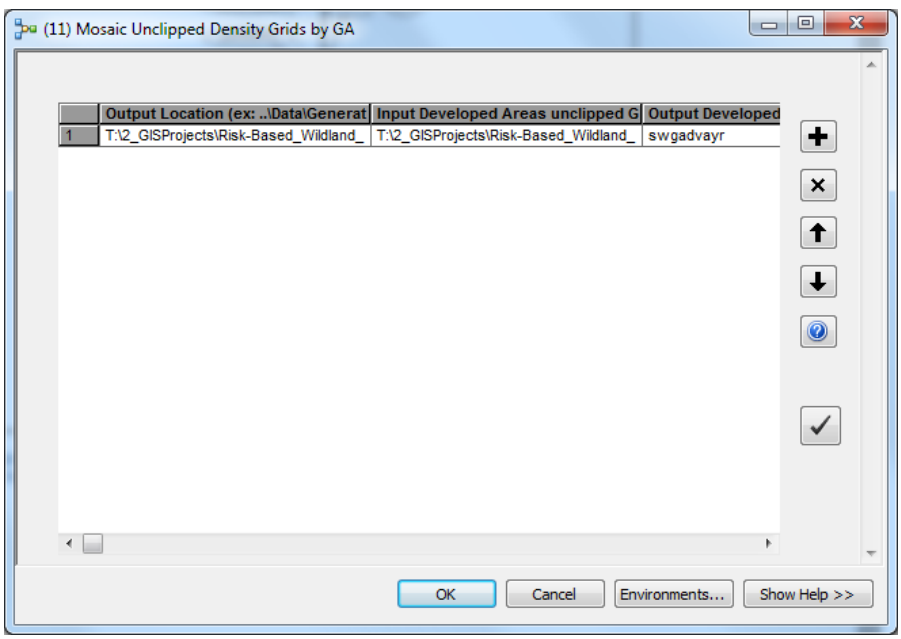


Toolbox_Batch_Templates_ver5_20150204.xlsx - Microsoft Excel

	A	B	C	D	E	F
1	(11) - Mosaic Unclipped Density Grids by GA					
2						
3	Enter Drive Letter:	T:				
4	Enter GA:	Southwest_GA	Entry data is located at: E38			
5					Output Location	unclipped Grids (ex: caca1dvayr, caca2dvayr, caca3dvayr...)
6					\2_GISProjects\Risk-Based_Wildland_Fire_Mgmt\	
7						dvayr
8		SW_AZ_001	swaz1	swga	T:\2_GISProjects\	T:\2_GISProjects\Risk-Based_Wil
9		SW_AZ_002	swaz2			T:\2_GISProjects\Risk-Based_Wil
10		SW_AZ_003	swaz3			T:\2_GISProjects\Risk-Based_Wil
11		SW_AZ_004	swaz4			T:\2_GISProjects\Risk-Based_Wil
12		SW_AZ_005	swaz5			T:\2_GISProjects\Risk-Based_Wil
13		SW_AZ_006	swaz6			T:\2_GISProjects\Risk-Based_Wil
14		SW_NM_001	swnm1			T:\2_GISProjects\Risk-Based_Wil
15		SW_NM_002	swnm2			T:\2_GISProjects\Risk-Based_Wil
16		SW_NM_003	swnm3			T:\2_GISProjects\Risk-Based_Wil
17		SW_NM_004	swnm4			T:\2_GISProjects\Risk-Based_Wil
18		SW_NM_005	swnm5			T:\2_GISProjects\Risk-Based_Wil
19		SW_NM_007	swnm7			T:\2_GISProjects\Risk-Based_Wil
20		SW_TX_002	swtx2			T:\2_GISProjects\Risk-Based_Wil
21		SW_TX_004	swtx4			T:\2_GISProjects\Risk-Based_Wil
22						

Step 8 | Step 9 | Step 10 | Step 11 | Step 12 | Step 13 | Step

Figure 14: Excel spreadsheet with text for Python tool Step 11 - Mosaic FPU clipped Density Grids to form new raster grid feature extending to Geographic Area (GA) borders. Note that cells to copy start at E38.



(11) Mosaic Unclipped Density Grids by GA

	Output Location (ex: ..\Data\Generat	Input Developed Areas unclipped G	Output Developed
1	T:\2_GISProjects\Risk-Based_Wildland_	T:\2_GISProjects\Risk-Based_Wildland_	swgadvayr

OK | Cancel | Environments... | Show Help >>

Figure 15: Python tool with populated entry Step 11 - Mosaic FPU clipped Density Grids to form new raster grid feature extending to Geographic Area (GA) borders

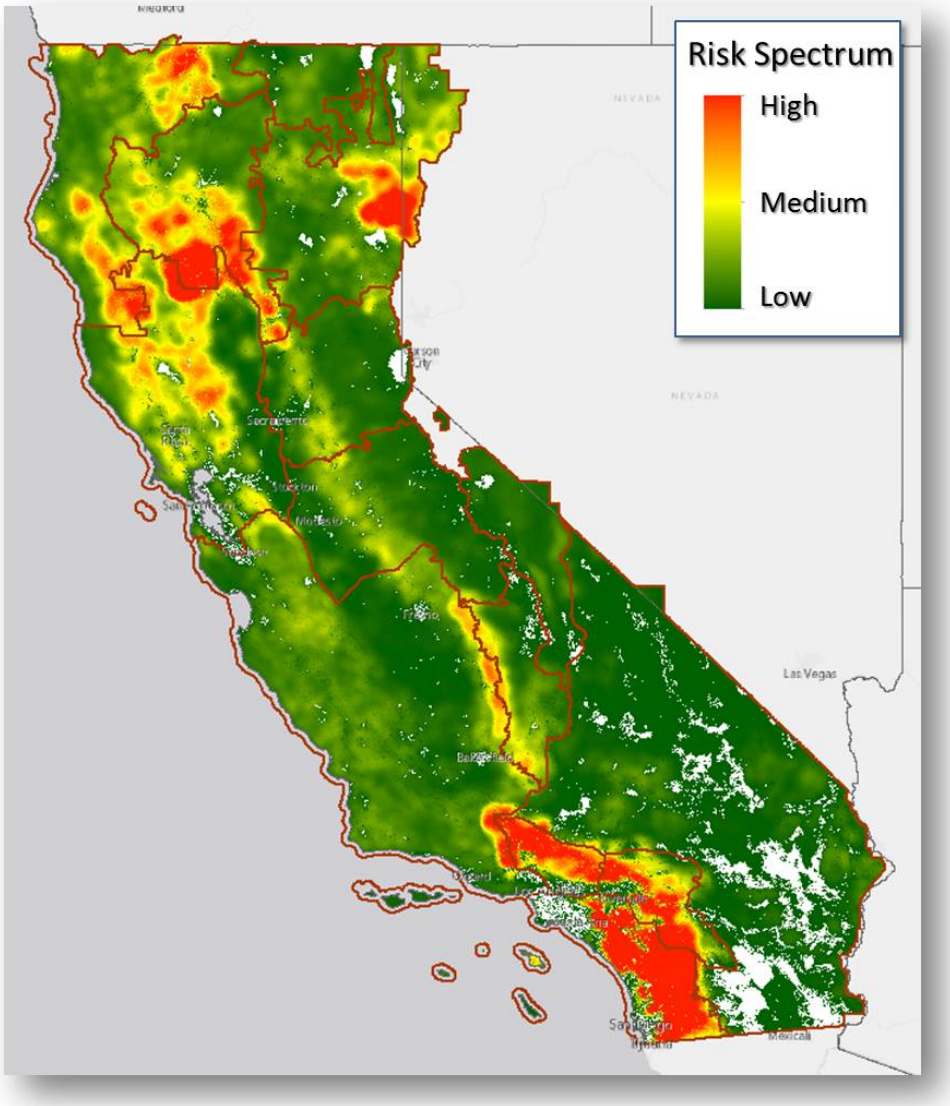
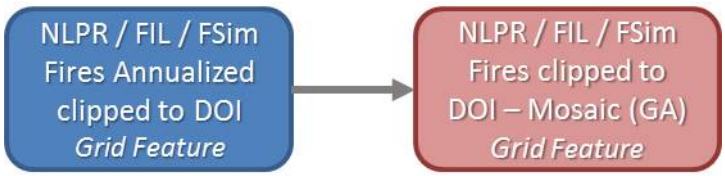


Figure 16: Example of raster Grid feature (California GA) displaying Density Grid of NLPR FSim fire start locations with acres affected per simulation year mosaicked into Geographic Area (GA) extents. The Cumulative analysis results for all NLPRs are shown.

Step 12. Create a mosaic of output from Step 9 with Geographic Area (GA) extents. Python script activates 'Mosaic to New Raster' tool via ArcGIS shell. (for Python script see Appendix N)



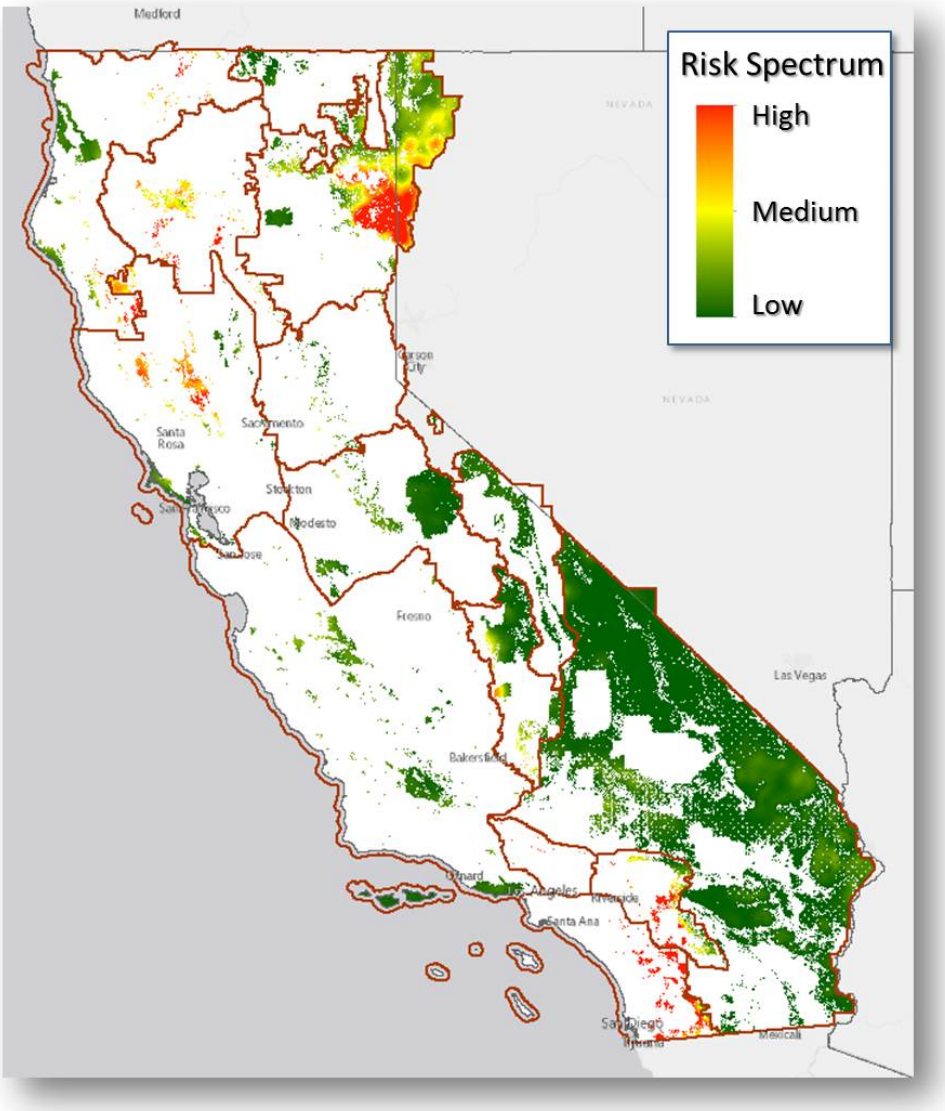
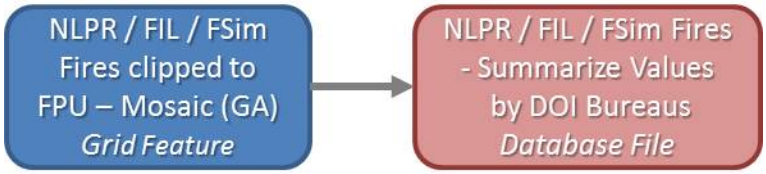


Figure 17: Output of Step 12-raster Grid feature (California GA) displaying Density Grid of NLPR FSim fire start locations with acres affected per simulation year (clipped to DOI wildfire-fighting Bureau boundaries) mosaicked into Geographic Area (GA) extents. The Cumulative analysis results for all NLPRs are shown.

Step 13. A Python script performs the 'Zonal Statistics as Table' function in an ArcGIS shell to summarize the values for each Geographic Area. This process compares the values of each NLPR raster grid within the zones of the DOI grid. Each zone is comprised of cells in the DOI raster grid feature that have the same DOI Bureau value (BIA, BLM, FWS or NPS). The sum is calculated for each zone as defined by grid cells containing each Bureau's lands. These are based on NLPR values associated to the same cell

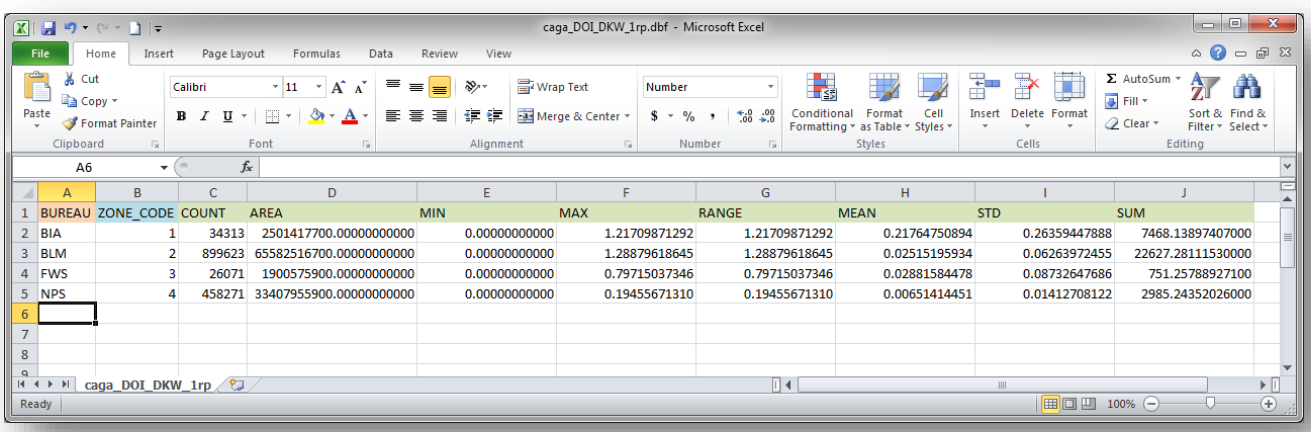
location from the NLPR datasets (see Figure 19). The process returns the results as a database table. (See Figure 20). (for Python script see Appendix O)



=

Rowid	VALUE	COUNT	AREA	MIN	MAX	RANGE	MEAN	STD	SUM	VARIETY	MAJORITY	MINORITY	MEDIAN
1	0	5	5	0	2	2	0.6	0.8	3	3	0	1	0
2	1	5	5	0	3	3	1	1.095	5	3	0	3	1
3	2	3	3	1	2	1	1.667	0.471	5	2	2	1	2
4	4	1	1	3	3	0	3	0	3	1	3	3	3

Figure 18: Illustration of Zonal Statistics being generated where the DOI Bureaus grid is the 'Zonal Raster' and the NLPR grid is the 'Values Raster' with the corresponding summarized statistics in a database table.



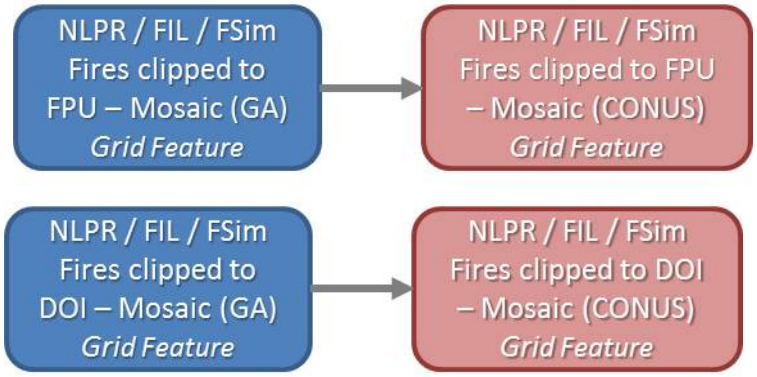
BUREAU	ZONE_CODE	COUNT	AREA	MIN	MAX	RANGE	MEAN	STD	SUM
BIA	1	34313	2501417700.00000000000	0.00000000000	1.21709871292	1.21709871292	0.21764750894	0.26359447888	7468.13897407000
BLM	2	899623	65582516700.00000000000	0.00000000000	1.28879618645	1.28879618645	0.02515195934	0.06263972455	22627.28111530000
FWS	3	26071	1900575900.00000000000	0.00000000000	0.79715037346	0.79715037346	0.02881584478	0.08732647686	751.25788927100
NPS	4	458271	33407955900.00000000000	0.00000000000	0.19455671310	0.19455671310	0.00651414451	0.01412708122	2985.24352026000

Figure 19: Example of database table (California GA) displaying the summarized values of each NLPR raster grid cells within the zones of the DOI Bureau grid cells. Drinking Water-Index of Importance is the NLPR in this example.

Step 14. Create a mosaic of output from Step 12, combining GAs into groups of two. Python script activates 'Mosaic to New Raster' tool via ArcGIS shell. (for Python script see Appendix P)

Step 15. Create a mosaic of output from Step 14, combining GAs into groups of four. Python script activates 'Mosaic to New Raster' tool via ArcGIS shell. (for Python script see Appendix P)

Step 16. Create a mosaic of output from Step 15, combining GAs into whole composite of a CONUS extent raster grid, one form is a mosaic of FPUs and the other a mosaic of DOI Lands, as illustrated in Figure 21. (for Python script see Appendix P)



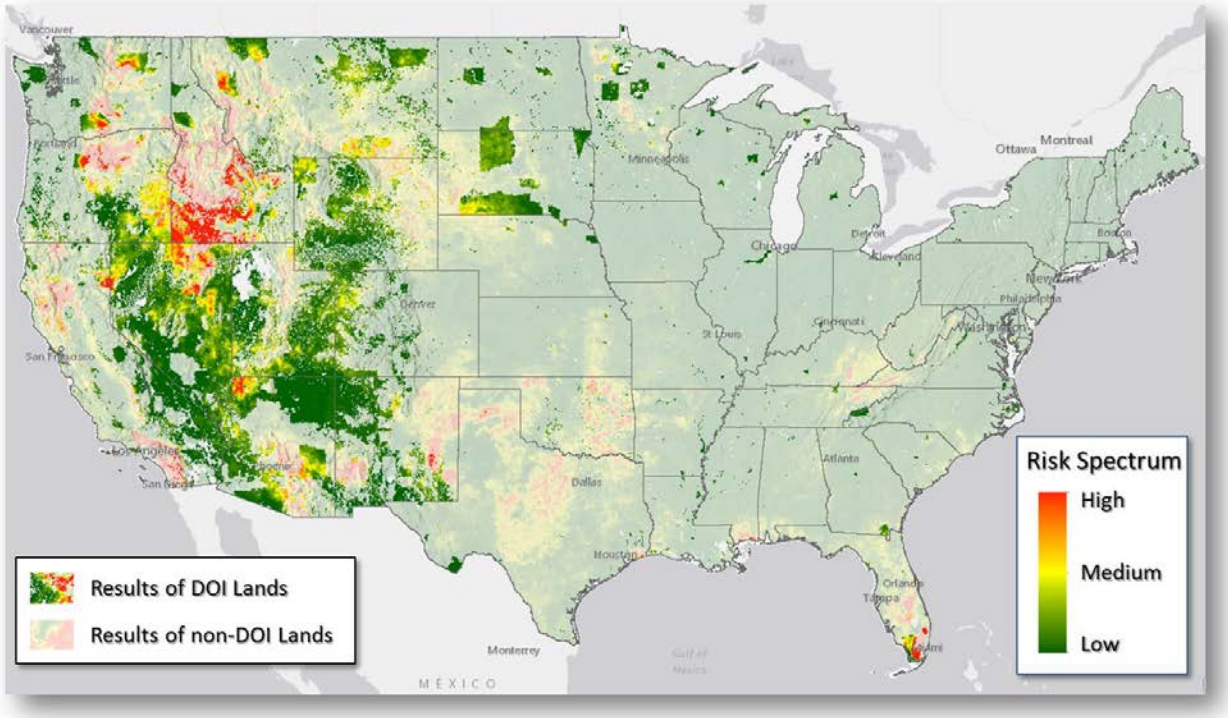


Figure 20. Output of Step 16 – raster Grid feature displaying merged grid features at the GA coverage mosaicked to CONUS level extent. The cumulative analysis results for all NLPRs are shown.

After the 16 steps are complete, the sums from the tables produced in step 13 are put into an Excel spreadsheet. The GA results are summed into a national result of expected NLPR acres burned.

References

Bachmann, A. and Allgöwer, B. (1999). The need for a consistent wildfire risk terminology. In: G.E. Gollberg (Editor), The Joint Fire Science Conference and Workshop, University of Idaho and International Association of Wildland Fire. Boise, Idaho

Calkin, David E.; Ager, Alan A.; Gilbertson-Day, Julie, eds. 2010. Wildfire risk and hazard: procedures for the first approximation. Gen. Tech. Rep. RMRS-GTR-235. Fort Collins, CO: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station

Finney, MA, McHugh, CW., Grenfell, IC, Riley, KL, Short, KC (2011) A simulation of probabilistic wildfire risk components for the continental United States, Stoch Environ Res Risk Assess DOI 10.1007/s00477-011-0462-z

Lutz, M. (1996) Programming Python. O'Reilly

Scott, Joe H.; Thompson, Matthew P.; Calkin, David E. 2013. A wildfire risk assessment framework for land and resource management. Gen. Tech. Rep. RMRS-GTR-315. U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station.

Zanderbergen, P. (2013) Python Scripting for ArcGIS. ESRI Press

Appendix A

A. NLPR Elements used in analysis for CONUS, Hawaii and Puerto Rico

1. Developed Areas
    a. Landscan
        i. CONUS_day
        ii. CONUS_night
        iii. HI_day
        iv. HI_night
        v. PR_night

        Landscan is a high resolution global population data set from Oakridge National Laboratory.
        Data Source – HSIP Gold 2013 Landscan_USA
        HSIP Gold 2013 is a unified homeland infrastructure geospatial data inventory for common use by the Federal Homeland Security and Homeland Defense Community
    b. Communities at Risk (CARs)
        i. Com_risk_tiger_113_dd83.shp

        Data Source – GeoMAC.gov
        National Association of State Foresters- Communities considered "At Risk" from wildland fires.

    c. NPS Facilities
        i. NPS_FacilityDatasets.gdb

        Data Source – NPS

2. Critical Infrastructure
    a. Communications
        i. AM Antennas – All data used
        ii. Antenna Structure Registrate – All data used
        iii. Cellular Towers– All data used
        iv. FM Antennas– All data used
        v. Microwave Service Towers– All data used
        vi. Paging Service Transmitters– All data used
        vii. TV Digital Station Transmitters– All data used
        viii. TV NTSC Station Transmitters– All data used

        Data Source – HSIP GOLD 2013 →Infrastructure→Communications.gdb

    b. Energy
        i. Electric Generating Units – Only Operating
        ii. Electric Power Generation Plants – Only Operating
        iii. Flowgates – Points – All data used
        iv. Natural Gas Storage Facilities– Operation or Operational with Planned Additions
        v. Oil and Natural Gas Interconnects – All data used
        vi. Nuclear Power Plants – Operating and Retired
        vii. Nuclear Research Facilities – All data used
        viii. POL Terminals / Storage Facilities / Tank Farms – In Service
        ix. Propane Locations – All data used
        x. Oil Refineries – All except Proposed

xi. Substations – In Service
xii. Oil and Natural Gas Wells - Only Operating
xiii. Flowgates – Line – All data used
xiv. Oil and Natural Gas Pipelines - Only Operating
xv. Transmission Lines - Only In Service

Data Source – HSIP GOLD 2013 →Infrastructure→Energy.gdb

c. Transportation Air
   i. Aero-Navigational Aids – All data used
   ii. Aero Obstructions – All data used

   Data Source – HSIP GOLD 2013 →Infrastructure→TransAir.gdb

d. Transportation Ground
   i. Not used in this effort

   Data Source – HSIP GOLD 2013 →Infrastructure→TransGround.gdb

3. Recreation
   a. Campgrounds
      Data Source – HSIP GOLD 2013 →Infrastructure→PubVenues.gdb
      Data Source – http:www.uscampgrounds.info for additional campgrounds in CONUS and HI
      Data Source – http://www2.pr.gov/agencias/gis/descargaGeodatos/Dotaciones/Pages/Recreaci%C3%B3n.aspx for additional campgrounds in PR

   b. Ski Areas
      Data Source – HSIP GOLD 2013 →Infrastructure→PubVenues.gdb

4. Riparian
   Data Source – National Hydrography Dataset (NHD) flowline, medium resolution, streams and rivers, perennial; PR- NHD flowline, high resolution, streams and rivers, perennial. http://nhd.usgs.gov/index.html

5. Forest Areas
   Data Source – Canopy cover threshold >10 % from LANDFIRE 1.2.0 Canopy Cover layer. Forest Type data for Hawaii and Puerto Rico are from LANDFIRE 1.2.0 EVT layer with a canopy cover threshold of >10% using LANDFIRE 1.2.0 Canopy Cover layer LANDFIRE http://www.landfire.gov/

6. Threatened and Endangered Species
   Data Source – Critical Habitat for Threatened & Endangered Species - Environmental Conservation Online System http://ecos.fws.gov/ecos/home.action

7. Surface Drinking Water
   Data Source – Land areas most important to surface drinking water, USDA Forest to Faucets http://www.fs.fed.us/ecosystemservices/FS_Efforts/GetF2FData/index.php

8. Sagebrush Steppe

   Data Source – Preliminary Priority Habitat (PPH), BLM GRSG_GUSG_Habitat.shp

   http://www.blm.gov/wo/st/en/prog/more/sagegrouse/documents_and_resources.html

9. FSim

   Data Source – Fire Program Analysis 2014 Standard FSim results

   Available upon request, OWF, Boise, ID

B. NLPR Elements used in analysis for Alaska

There are seven categories of Alaska NLPR values.

Once the NLPRs were identified, the data was obtained under these guiding principals

1. Geospatial data should be nationally consistent from Authoritative Sources
2. All NLPR values will be represented without Buffering
3. All Developed Areas, Infrastructure, Native Allotments & Corporation Lands and Reindeer Herding Areas are only within the Fire Management Options designated as "Critical", "Full", or "Modified" suppression areas.

   Alaska descriptions of the four wildland fire management options:

   A. Critical Management Option - Intent

   These are the highest priority areas/sites for suppression actions and assignment of available firefighting resources. Lands in wildland urban interface and other populated areas where there is an immediate threat to human life, primary residences, inhabited property, community-dependent infrastructure, and structural resources designated as National Historic Landmarks qualify to be considered for this designation. This classification is applicable an entire village or town as well as a single inhabited structure.

   B. Full Management Option - Intent

   This option provides for protection of cultural and paleontological sites, developed recreational facilities, physical developments, administrative sites and cabins, uninhabited structures, high-value natural resources, and other high-value areas that do not involve the protection of human life and inhabited property. Structures on or eligible for inclusion on the National Register of Historic Places and non-structural sites on the National Register are placed within this category. Either broad areas or specific sites qualify to be designated as Full.

   C. Limited Management Option - Intent

   Limited is designed for broad, landscape-scale areas where the low density and wide distribution of values to be protected best allows for fire to function in its ecological role. Sites that warrant higher levels of protection may occur within the boundaries of Limited areas and actions to protect these sites will be taken when warranted without compromising the intent of this management option.

   Wildland fire is used as a management tool to maintain, enhance and improve ecological conditions. In these areas, fire is routinely able to function in its natural roles as an essential ecological process. Limited is also assigned to areas where the cost of suppression may exceed the value of the resources to be protected or the

environmental impacts of fire suppression activities may have more negative impacts on the resources than the effects of the fire.

D. Modified Management Option - Intent

Modified is the most adaptable option available to jurisdictional agencies since the response to wildland fire is based upon fire and environmental conditions. The essential element of this option is the conversion dates described below. Sites that warrant higher levels of protection may occur within Modified areas. The Modified option provides a management level between Full and Limited. Unlike Full management areas, the intent is not to minimize burned acres, but to balance acres burned with suppression costs and, similar to Limited, to accomplish land and resource management objectives when conditions are favorable.

1. Under suitable fire and environmental conditions, accomplish fire-related landuse and resource objectives in a cost-effective manner while providing appropriate levels of protection to identified sites.

2. Maintain the flexibility to respond to fire conditions and tailor the initial action to those conditions.

4. Life and Property & Resource Values

1. Life & Property

-Landscan Day and Night*
-Communities at Risk**
-National Park Service Facilities
-Alaska Known Developed sites***

*LandScan is a high resolution global population data set – Oakridge National Laboratory

** National Association of State Foresters - Communities considered "At Risk" from wildland fire

***Known Developed Sites in Critical, Full or Modified Suppression Areas – Alaska Fire Service (AFS)

2. Infrastructure

-Communications sites*
-Energy corridors*
-Air Transportation sites*
-Alaska Known Infrastructure sites**

*HSIP Gold 2013 is a unified homeland infrastructure geospatial data inventory for common use by the Federal Homeland Security and Homeland Defense Community (HSIP Gold 2013)

**Known Infrastructure Sites in Critical, Full or Modified Suppression Areas – Alaska Fire Service (AFS)

3. Recreation Sites

-Ski areas*
-Campgrounds*

*(HSIP Gold 2013)

4. Native Allotments and Native Corporation Lands

-ANCSA Allotments and Corporation Lands*

*(Native Corporation Lands in Critical, Full or Modified Suppression Areas – Alaska Fire Service (AFS)

5. Reindeer Herding Areas

-Bering Land Bridge National Preserve (NPS) / Anchorage Field Office & Central Yukon Field Office (BLM)*

*Reindeer herding areas in Critical, Full or Modified Suppression Areas – Alaska Fire Service (AFS)

6. Threatened and Endangered Species

-Critical Habitat and Recovery Plan sites for Threatened & Endangered Species*

*USFWS Environmental Conservation Online System

7. Surface Drinking Water

-Land areas most important to surface drinking water*

*The USDA Forest Service Forests to Faucets project uses GIS to model and map the continental United States land areas most important to surface drinking water

Processing Steps:

1. Prior to creating one combined layer, all layers need to have the same extent and the null values filled in.
2. Use the ArcGIS Append tool to create one layer for a NLPR type.
3. Reproject the Feature Class to NAD83 Albers. The coordinate system for the grid data layers that were used in this modeling analysis contained the following parameters:

Projection: Albers	Geographic Coordinate System: GCS North American 1983
False Easting: 0.0°	Angular Unit: Degree (0.0174532925199433)
False Northing: 0.0°	Prime Meridian: Greenwich (0.0°)
Central Meridian: -96.0°	Datum: D North American 1983 (NAD83)
Standard Parallel 1: 29.5°	Spheroid: GRS 1980
Standard Parallel 2: 45.5°	Semi-major Axis: 6378137.0m
Latitude of Origin: 23.0°	Semi-minor Axis: 6356752.314140356m
Linear Unit: Meter (1.0)	Inverse Flattening: 298.257222101m
ESRI Name: NAD_1983_Albers	

4. Create Raster layer by using Feature to Raster. For Processing Extent, use the FSim results. The X and Y coordinate should now be 270 meters for CONUS and 1000 meters for Alaska.
5. In order to combine all values later, a coding was selected. As there were eight values selected, there are eight Boolean values.
   i. Sagebrush Steppe was 10000000, or a 1 in the first position.
   ii. Surface Drinking water was 01000000
   iii. Threatened and Endangered Species was 00100000
   iv. Forest Areas was 00010000
   v. Riparian was 00001000

vi. Recreation Sites was 00000100

vii. Critical Infrastructure was 00000010

viii. Developed Areas was 00000001

ix. When combined, the values are added, so for example a location with Sagebrush Steppe and Developed Areas would be 10000001.

6. Combine with the other values layers. Using the coding just created, combine using the Raster Calculator into one field called grdcode.

(Note 1: Several NLPR vector base data layers were rasterized into an intermediate step that formed cell resolution of 322m^2. These were only in the instance for the FPU designated "Caribbean" (SA_PR_001) which encompasses Puerto Rico and the US Virgin Islands. The NLPR values that were converted this way were: Forest Types, Riparian Areas, and Recreation Sites. These were subsequently resampled and transformed into cell resolutions of 270m^2 which is the model's standard raster cell dimensions for areas in the OCONUS extent.**)**

(Note 2: A few of the grid data layers that went through the intermediate process of converting from point, line, and polygon vector data were found to have different cell alignments then the model's desired grid cell orientation (this being the same extent and cell alignment as the FSim Burn Probability grid data layer). The ArcTool 'Snap Raster' utility was used to adjust the extent of these misaligned grid data layers to match the cell alignment of the FSim grid data layer's cell arrangement. This ensured that all of the model's grid data were harmonized during subsequent geospatial processing procedures.**)**

Appendix B

The process for the RBWFM Analysis was informed by what's found in the publications The Need for a Consistent Wildfire Risk Terminology (Bachmann, Allgöwer 1999), Wildfire Risk and Hazard: Procedures for the First Approximation (RMRS-GTR-235), and A Wildfire Risk Assessment Framework for Land and Resource Management (RMRS-GTR-315).

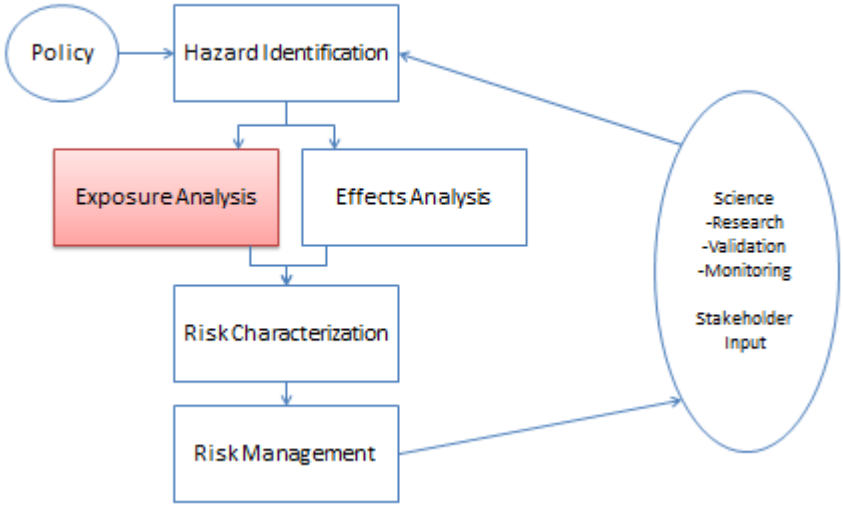


Figure A- 1 Risk Analysis Framework

This portion of the RBWFM focuses on the Expected Acres burned, which is the Exposure Analysis portion of Risk Analysis. Effects analysis are more complicated at a national scale, and so are left to the bureaus to incorporate their specific missions, local knowledge, and planning.

It should also be noted for future analyses that a workload coefficient was also investigated, based on fuel characteristics (tonnage, and bed depth), as well as production rates, but further investigation is needed.

Appendix C

All Python code was created in the PythonWin 2.7.5 application GUI and IDE editor
(default, May 15 2013, 22:43:36) [MSC v.1500 32 bit (Intel)] on win32 using PyWin32 Build 214.

PythonWin - Python IDE and GUI Framework for Windows.
Copyright 1994-2008 Mark Hammond

Python is Copyright (c) 2001-2015 Python Software Foundation.
All Rights Reserved. <www.python.org>

Copyright (c) 1991-1995 Stichting Mathematisch Centrum, Amsterdam.
All Rights Reserved.

Python Script for Step 1:

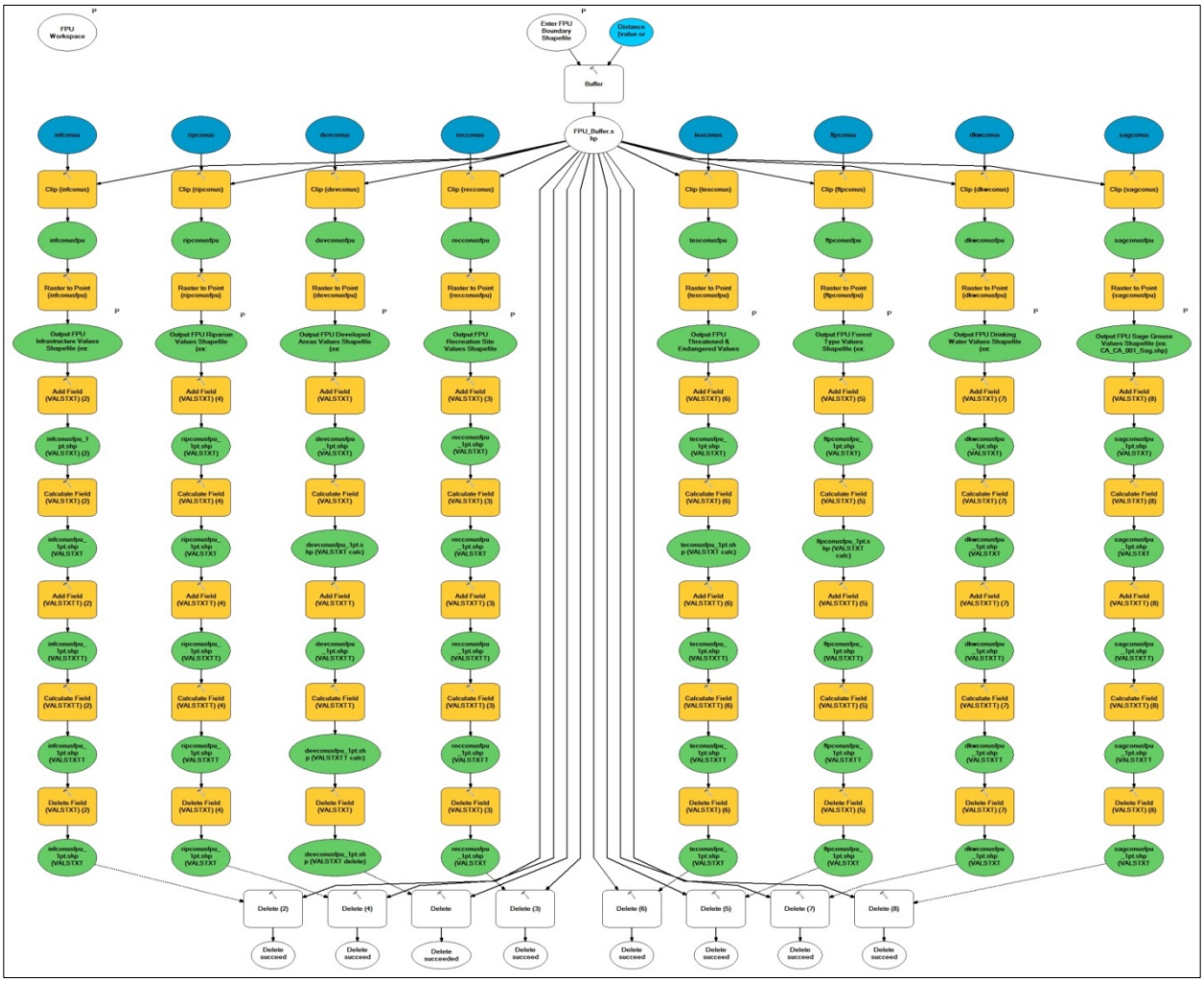


Figure A- 2: Step 1 process flow diagram - Clip NLPRs to buffered FPU extent

```
# ---------------------------------------------------------------------------
# Buffer_FPU_Clip_NLPR.py
# Created on: 2015-02-06 11:27:08.00000
# Usage: Buffer_FPU_Clip_NLPR <FPU_Workspace>
<Enter_FPU_Boundary_Shapefile__ex__CA_CA_001_FPU_albrs_1ex_shp_>
<Output_FPU_Developed_Areas_Values_Shapefile__ex__CA_CA_001_Dev_shp_>
<Output_FPU_Infrastructure_Values_Shapefile__ex__CA_CA_001_Inf_shp_>
<Output_FPU_Recreation_Site_Values_Shapefile__ex__CA_CA_001_Rec_shp_>
<Output_FPU_Riparian_Values_Shapefile__ex__CA_CA_001_Rip_shp_>
<Output_FPU_Forest_Type_Values_Shapefile__ex__CA_CA_001_Ftp_shp_>
<Output_FPU_Threatened___Endangered_Values_Shapefile__ex__CA_CA_001_Tes_shp_>
<Output_FPU_Drinking_Water_Values_Shapefile__ex__CA_CA_001_Dkw_shp_>
<Output_FPU_Sage_Grouse_Values_Shapefile__ex__CA_CA_001_Sag_shp_>
# Description: Clip NLPRs to buffered FPU extent
# ---------------------------------------------------------------------------

# Import arcpy module
import arcpy

# Script arguments
FPU_Workspace = arcpy.GetParameterAsText(0)

Enter_FPU_Boundary_Shapefile__ex__CA_CA_001_FPU_albrs_1ex_shp_ =
arcpy.GetParameterAsText(1)

Output_FPU_Developed_Areas_Values_Shapefile__ex__CA_CA_001_Dev_shp_ =
arcpy.GetParameterAsText(2)
if Output_FPU_Developed_Areas_Values_Shapefile__ex__CA_CA_001_Dev_shp_ == '#' or not
Output_FPU_Developed_Areas_Values_Shapefile__ex__CA_CA_001_Dev_shp_:
   Output_FPU_Developed_Areas_Values_Shapefile__ex__CA_CA_001_Dev_shp_ = "%FPU
Workspace%\\devconusfpu_1pt.shp" # provide a default value if unspecified

Output_FPU_Infrastructure_Values_Shapefile__ex__CA_CA_001_Inf_shp_ =
arcpy.GetParameterAsText(3)
if Output_FPU_Infrastructure_Values_Shapefile__ex__CA_CA_001_Inf_shp_ == '#' or not
Output_FPU_Infrastructure_Values_Shapefile__ex__CA_CA_001_Inf_shp_:
   Output_FPU_Infrastructure_Values_Shapefile__ex__CA_CA_001_Inf_shp_ = "%FPU
Workspace%\\infconusfpu_1pt.shp" # provide a default value if unspecified

Output_FPU_Recreation_Site_Values_Shapefile__ex__CA_CA_001_Rec_shp_ =
arcpy.GetParameterAsText(4)
if Output_FPU_Recreation_Site_Values_Shapefile__ex__CA_CA_001_Rec_shp_ == '#' or not
Output_FPU_Recreation_Site_Values_Shapefile__ex__CA_CA_001_Rec_shp_:
   Output_FPU_Recreation_Site_Values_Shapefile__ex__CA_CA_001_Rec_shp_ = "%FPU
Workspace%\\recconusfpu_1pt.shp" # provide a default value if unspecified

Output_FPU_Riparian_Values_Shapefile__ex__CA_CA_001_Rip_shp_ = arcpy.GetParameterAsText(5)
```

```
if Output_FPU_Riparian_Values_Shapefile__ex__CA_CA_001_Rip_shp_ == '#' or not
Output_FPU_Riparian_Values_Shapefile__ex__CA_CA_001_Rip_shp_:
   Output_FPU_Riparian_Values_Shapefile__ex__CA_CA_001_Rip_shp_ = "%FPU
Workspace%\\ripconusfpu_1pt.shp" # provide a default value if unspecified

Output_FPU_Forest_Type_Values_Shapefile__ex__CA_CA_001_Ftp_shp_ =
arcpy.GetParameterAsText(6)
if Output_FPU_Forest_Type_Values_Shapefile__ex__CA_CA_001_Ftp_shp_ == '#' or not
Output_FPU_Forest_Type_Values_Shapefile__ex__CA_CA_001_Ftp_shp_:
   Output_FPU_Forest_Type_Values_Shapefile__ex__CA_CA_001_Ftp_shp_ = "%FPU
Workspace%\\ftpconusfpu_1pt.shp" # provide a default value if unspecified

Output_FPU_Threatened___Endangered_Values_Shapefile__ex__CA_CA_001_Tes_shp_ =
arcpy.GetParameterAsText(7)
if Output_FPU_Threatened___Endangered_Values_Shapefile__ex__CA_CA_001_Tes_shp_ == '#' or not
Output_FPU_Threatened___Endangered_Values_Shapefile__ex__CA_CA_001_Tes_shp_:
   Output_FPU_Threatened___Endangered_Values_Shapefile__ex__CA_CA_001_Tes_shp_ = "%FPU
Workspace%\\teconusfpu_1pt.shp" # provide a default value if unspecified

Output_FPU_Drinking_Water_Values_Shapefile__ex__CA_CA_001_Dkw_shp_ =
arcpy.GetParameterAsText(8)
if Output_FPU_Drinking_Water_Values_Shapefile__ex__CA_CA_001_Dkw_shp_ == '#' or not
Output_FPU_Drinking_Water_Values_Shapefile__ex__CA_CA_001_Dkw_shp_:
   Output_FPU_Drinking_Water_Values_Shapefile__ex__CA_CA_001_Dkw_shp_ = "%FPU
Workspace%\\dkwconusfpu_1pt.shp" # provide a default value if unspecified

Output_FPU_Sage_Grouse_Values_Shapefile__ex__CA_CA_001_Sag_shp_ =
arcpy.GetParameterAsText(9)
if Output_FPU_Sage_Grouse_Values_Shapefile__ex__CA_CA_001_Sag_shp_ == '#' or not
Output_FPU_Sage_Grouse_Values_Shapefile__ex__CA_CA_001_Sag_shp_:
   Output_FPU_Sage_Grouse_Values_Shapefile__ex__CA_CA_001_Sag_shp_ = "%FPU
Workspace%\\sagconusfpu_1pt.shp" # provide a default value if unspecified

# Local variables:
FPU_Buffer_shp = Enter_FPU_Boundary_Shapefile__ex__CA_CA_001_FPU_albrs_1ex_shp_
devconusfpu = FPU_Buffer_shp
Delete_succeeded = FPU_Buffer_shp
infconusfpu = FPU_Buffer_shp
Delete_succeeded__2_ = FPU_Buffer_shp
recconusfpu = FPU_Buffer_shp
Delete_succeeded__3_ = FPU_Buffer_shp
ripconusfpu = FPU_Buffer_shp
Delete_succeeded__4_ = FPU_Buffer_shp
ftpconusfpu = FPU_Buffer_shp
Delete_succeeded__5_ = FPU_Buffer_shp
tesconusfpu = FPU_Buffer_shp
Delete_succeeded__6_ = FPU_Buffer_shp
dkwconusfpu = FPU_Buffer_shp
```

Delete_succeeded__7_ = FPU_Buffer_shp
sagconusfpu = FPU_Buffer_shp
Delete_succeeded__8_ = FPU_Buffer_shp
Distance__value_or_field_ = "15 Kilometers"
devconus = "T:\\2_GISProjects\\Risk-Based_Wildland_Fire_Mgmt\\Data\\Grids\\Rasters_ver4\\Developed_Areas\\devconus"
devconusfpu_1pt_shp__VALSTXT_ = Output_FPU_Developed_Areas_Values_Shapefile__ex__CA_CA_001_Dev_shp_
devconusfpu_1pt_shp__VALSTXT_calc_ = devconusfpu_1pt_shp__VALSTXT_
devconusfpu_1pt_shp__VALSTXTT_ = devconusfpu_1pt_shp__VALSTXT_calc_
devconusfpu_1pt_shp__VALSTXTT_calc_ = devconusfpu_1pt_shp__VALSTXTT_
devconusfpu_1pt_shp__VALSTXT_delete_ = devconusfpu_1pt_shp__VALSTXTT_calc_
infconus = "T:\\2_GISProjects\\Risk-Based_Wildland_Fire_Mgmt\\Data\\Grids\\Rasters_ver4\\Infrastructure\\infconus"
infconusfpu_1pt_shp__VALSTXT___2_ = Output_FPU_Infrastructure_Values_Shapefile__ex__CA_CA_001_Inf_shp_
infconusfpu_1pt_shp__VALSTXT_calc_ = infconusfpu_1pt_shp__VALSTXT___2_
infconusfpu_1pt_shp__VALSTXTT_ = infconusfpu_1pt_shp__VALSTXT_calc_
infconusfpu_1pt_shp__VALSTXTT_calc_ = infconusfpu_1pt_shp__VALSTXTT_
infconusfpu_1pt_shp__VALSTXT_delete_ = infconusfpu_1pt_shp__VALSTXTT_calc_
recconus = "T:\\2_GISProjects\\Risk-Based_Wildland_Fire_Mgmt\\Data\\Grids\\Rasters_ver4\\Recreation_Sites\\recconus"
recconusfpu_1pt_shp__VALSTXT___3_ = Output_FPU_Recreation_Site_Values_Shapefile__ex__CA_CA_001_Rec_shp_
recconusfpu_1pt_shp__VALSTXT_calc_ = recconusfpu_1pt_shp__VALSTXT___3_
recconusfpu_1pt_shp__VALSTXTT_ = recconusfpu_1pt_shp__VALSTXT_calc_
recconusfpu_1pt_shp__VALSTXTT_calc_ = recconusfpu_1pt_shp__VALSTXTT_
recconusfpu_1pt_shp__VALSTXT_delete_ = recconusfpu_1pt_shp__VALSTXTT_calc_
ripconus = "T:\\2_GISProjects\\Risk-Based_Wildland_Fire_Mgmt\\Data\\Grids\\Rasters_ver4\\Riparian_Areas\\ripconus"
ripconusfpu_1pt_shp__VALSTXT_ = Output_FPU_Riparian_Values_Shapefile__ex__CA_CA_001_Rip_shp_
ripconusfpu_1pt_shp__VALSTXT_calc_ = ripconusfpu_1pt_shp__VALSTXT_
ripconusfpu_1pt_shp__VALSTXTT_ = ripconusfpu_1pt_shp__VALSTXT_calc_
ripconusfpu_1pt_shp__VALSTXTT_calc_ = ripconusfpu_1pt_shp__VALSTXTT_
ripconusfpu_1pt_shp__VALSTXT_delete_ = ripconusfpu_1pt_shp__VALSTXTT_calc_
ftpconus = "T:\\2_GISProjects\\Risk-Based_Wildland_Fire_Mgmt\\Data\\Grids\\Rasters_ver4\\Forest_Areas\\ftpconus"
ftpconusfpu_1pt_shp__VALSTXT_ = Output_FPU_Forest_Type_Values_Shapefile__ex__CA_CA_001_Ftp_shp_
ftpconusfpu_1pt_shp__VALSTXT_calc_ = ftpconusfpu_1pt_shp__VALSTXT_
ftpconusfpu_1pt_shp__VALSTXTT_ = ftpconusfpu_1pt_shp__VALSTXT_calc_
ftpconusfpu_1pt_shp__VALSTXTT_calc_ = ftpconusfpu_1pt_shp__VALSTXTT_
ftpconusfpu_1pt_shp__VALSTXT_delete_ = ftpconusfpu_1pt_shp__VALSTXTT_calc_
tesconus = "T:\\2_GISProjects\\Risk-Based_Wildland_Fire_Mgmt\\Data\\Grids\\Rasters_ver4\\Threatened_Endangered\\tesconus"
teconusfpu_1pt_shp__VALSTXT_ = Output_FPU_Threatened___Endangered_Values_Shapefile__ex__CA_CA_001_Tes_shp_

```
teconusfpu_1pt_shp__VALSTXT_calc_ = teconusfpu_1pt_shp__VALSTXT_
teconusfpu_1pt_shp__VALSTXTT_ = teconusfpu_1pt_shp__VALSTXT_calc_
teconusfpu_1pt_shp__VALSTXTT_calc_ = teconusfpu_1pt_shp__VALSTXTT_
teconusfpu_1pt_shp__VALSTXT_delete_ = teconusfpu_1pt_shp__VALSTXTT_calc_
dkwconus = "T:\\2_GISProjects\\Risk-
Based_Wildland_Fire_Mgmt\\Data\\Grids\\Rasters_ver4\\Drinking_Water\\dkwconus"
dkwconusfpu_1pt_shp__VALSTXT_ =
Output_FPU_Drinking_Water_Values_Shapefile__ex__CA_CA_001_Dkw_shp_
dkwconusfpu_1pt_shp__VALSTXT_calc_ = dkwconusfpu_1pt_shp__VALSTXT_
dkwconusfpu_1pt_shp__VALSTXTT_ = dkwconusfpu_1pt_shp__VALSTXT_calc_
dkwconusfpu_1pt_shp__VALSTXTT_calc_ = dkwconusfpu_1pt_shp__VALSTXTT_
dkwconusfpu_1pt_shp__VALSTXT_delete_ = dkwconusfpu_1pt_shp__VALSTXTT_calc_
sagconus = "T:\\2_GISProjects\\Risk-
Based_Wildland_Fire_Mgmt\\Data\\Grids\\Rasters_ver4\\Sage_Grouse\\sagconus"
sagconusfpu_1pt_shp__VALSTXT_ =
Output_FPU_Sage_Grouse_Values_Shapefile__ex__CA_CA_001_Sag_shp_
sagconusfpu_1pt_shp__VALSTXT_calc_ = sagconusfpu_1pt_shp__VALSTXT_
sagconusfpu_1pt_shp__VALSTXTT_ = sagconusfpu_1pt_shp__VALSTXT_calc_
sagconusfpu_1pt_shp__VALSTXTT_calc_ = sagconusfpu_1pt_shp__VALSTXTT_
sagconusfpu_1pt_shp__VALSTXT_delete_ = sagconusfpu_1pt_shp__VALSTXTT_calc_

# Process: Buffer
arcpy.Buffer_analysis(Enter_FPU_Boundary_Shapefile__ex__CA_CA_001_FPU_albrs_1ex_shp_,
FPU_Buffer_shp, Distance__value_or_field_, "FULL", "ROUND", "NONE", "")

# Process: Clip (devconus)
arcpy.Clip_management(devconus, "-2354719.4099919 312634.153254099 2179660.5900081
3161944.1532541", devconusfpu, FPU_Buffer_shp, "0", "ClippingGeometry", "NO_MAINTAIN_EXTENT")

# Process: Raster to Point (devconusfpu)
arcpy.RasterToPoint_conversion(devconusfpu,
Output_FPU_Developed_Areas_Values_Shapefile__ex__CA_CA_001_Dev_shp_, "Value")

# Process: Add Field (VALSTXT)
arcpy.AddField_management(Output_FPU_Developed_Areas_Values_Shapefile__ex__CA_CA_001_Dev
_shp_, "VALSTXT", "TEXT", "", "", "10", "", "NULLABLE", "NON_REQUIRED", "")

# Process: Calculate Field (VALSTXT)
arcpy.CalculateField_management(devconusfpu_1pt_shp__VALSTXT_, "VALSTXT", "[GRID_CODE]",
"VB", "")

# Process: Add Field (VALSTXTT)
arcpy.AddField_management(devconusfpu_1pt_shp__VALSTXT_calc_, "VALSTXTT", "TEXT", "", "", "10",
"", "NULLABLE", "NON_REQUIRED", "")

# Process: Calculate Field (VALSTXTT)
arcpy.CalculateField_management(devconusfpu_1pt_shp__VALSTXTT_, "VALSTXTT", "Right(
\"00000000\" &  CStr([VALSTXT]),8 )", "VB", "")
```

```
# Process: Delete Field (VALSTXT)
arcpy.DeleteField_management(devconusfpu_1pt_shp__VALSTXTT_calc_, "VALSTXT")

# Process: Delete
arcpy.Delete_management(FPU_Buffer_shp, "")

# Process: Clip (infconus)
arcpy.Clip_management(infconus, "-2354719.4099919 312634.153254099 2179660.5900081
3161944.1532541", infconusfpu, FPU_Buffer_shp, "15", "ClippingGeometry", "NO_MAINTAIN_EXTENT")

# Process: Raster to Point (infconusfpu)
arcpy.RasterToPoint_conversion(infconusfpu,
Output_FPU_Infrastructure_Values_Shapefile__ex__CA_CA_001_Inf_shp_, "Value")

# Process: Add Field (VALSTXT) (2)
arcpy.AddField_management(Output_FPU_Infrastructure_Values_Shapefile__ex__CA_CA_001_Inf_shp_
, "VALSTXT", "TEXT", "", "", "10", "", "NULLABLE", "NON_REQUIRED", "")

# Process: Calculate Field (VALSTXT) (2)
arcpy.CalculateField_management(infconusfpu_1pt_shp__VALSTXT___2_, "VALSTXT", "[GRID_CODE]",
"VB", "")

# Process: Add Field (VALSTXTT) (2)
arcpy.AddField_management(infconusfpu_1pt_shp__VALSTXT_calc_, "VALSTXTT", "TEXT", "", "", "10",
"", "NULLABLE", "NON_REQUIRED", "")

# Process: Calculate Field (VALSTXTT) (2)
arcpy.CalculateField_management(infconusfpu_1pt_shp__VALSTXTT_, "VALSTXTT", "Right(
\"00000000\" &  CStr([VALSTXT]),8 )", "VB", "")

# Process: Delete Field (VALSTXT) (2)
arcpy.DeleteField_management(infconusfpu_1pt_shp__VALSTXTT_calc_, "VALSTXT")

# Process: Delete (2)
arcpy.Delete_management(FPU_Buffer_shp, "")

# Process: Clip (recconus)
arcpy.Clip_management(recconus, "-2354719.4099919 312634.153254099 2179660.5900081
3161944.1532541", recconusfpu, FPU_Buffer_shp, "0", "ClippingGeometry", "NO_MAINTAIN_EXTENT")

# Process: Raster to Point (recconusfpu)
arcpy.RasterToPoint_conversion(recconusfpu,
Output_FPU_Recreation_Site_Values_Shapefile__ex__CA_CA_001_Rec_shp_, "Value")

# Process: Add Field (VALSTXT) (3)
arcpy.AddField_management(Output_FPU_Recreation_Site_Values_Shapefile__ex__CA_CA_001_Rec_s
hp_, "VALSTXT", "TEXT", "", "", "10", "", "NULLABLE", "NON_REQUIRED", "")
```

```
# Process: Calculate Field (VALSTXT) (3)
arcpy.CalculateField_management(recconusfpu_1pt_shp__VALSTXT___3_, "VALSTXT", "[GRID_CODE]",
"VB", "")

# Process: Add Field (VALSTXTT) (3)
arcpy.AddField_management(recconusfpu_1pt_shp__VALSTXT_calc_, "VALSTXTT", "TEXT", "", "", "10",
"", "NULLABLE", "NON_REQUIRED", "")

# Process: Calculate Field (VALSTXTT) (3)
arcpy.CalculateField_management(recconusfpu_1pt_shp__VALSTXTT_, "VALSTXTT", "Right(
\"00000000\" &  CStr([VALSTXT]),8 )", "VB", "")

# Process: Delete Field (VALSTXT) (3)
arcpy.DeleteField_management(recconusfpu_1pt_shp__VALSTXTT_calc_, "VALSTXT")

# Process: Delete (3)
arcpy.Delete_management(FPU_Buffer_shp, "")

# Process: Clip (ripconus)
arcpy.Clip_management(ripconus, "-2354719.4099919 312634.153254099 2179660.5900081
3161944.1532541", ripconusfpu, FPU_Buffer_shp, "-32768", "ClippingGeometry",
"NO_MAINTAIN_EXTENT")

# Process: Raster to Point (ripconusfpu)
arcpy.RasterToPoint_conversion(ripconusfpu,
Output_FPU_Riparian_Values_Shapefile__ex__CA_CA_001_Rip_shp_, "Value")

# Process: Add Field (VALSTXT) (4)
arcpy.AddField_management(Output_FPU_Riparian_Values_Shapefile__ex__CA_CA_001_Rip_shp_,
"VALSTXT", "TEXT", "", "", "10", "", "NULLABLE", "NON_REQUIRED", "")

# Process: Calculate Field (VALSTXT) (4)
arcpy.CalculateField_management(ripconusfpu_1pt_shp__VALSTXT_, "VALSTXT", "[GRID_CODE]", "VB",
"")

# Process: Add Field (VALSTXTT) (4)
arcpy.AddField_management(ripconusfpu_1pt_shp__VALSTXT_calc_, "VALSTXTT", "TEXT", "", "", "10",
"", "NULLABLE", "NON_REQUIRED", "")

# Process: Calculate Field (VALSTXTT) (4)
arcpy.CalculateField_management(ripconusfpu_1pt_shp__VALSTXTT_, "VALSTXTT", "Right(
\"00000000\" &  CStr([VALSTXT]),8 )", "VB", "")

# Process: Delete Field (VALSTXT) (4)
arcpy.DeleteField_management(ripconusfpu_1pt_shp__VALSTXTT_calc_, "VALSTXT")

# Process: Delete (4)
```

```
arcpy.Delete_management(FPU_Buffer_shp, "")

# Process: Clip (ftpconus)
arcpy.Clip_management(ftpconus, "-2354719.4099919 312634.153254099 2179660.5900081
3161944.1532541", ftpconusfpu, FPU_Buffer_shp, "-32768", "ClippingGeometry",
"NO_MAINTAIN_EXTENT")

# Process: Raster to Point (ftpconusfpu)
arcpy.RasterToPoint_conversion(ftpconusfpu,
Output_FPU_Forest_Type_Values_Shapefile__ex__CA_CA_001_Ftp_shp_, "Value")

# Process: Add Field (VALSTXT) (5)
arcpy.AddField_management(Output_FPU_Forest_Type_Values_Shapefile__ex__CA_CA_001_Ftp_shp_,
"VALSTXT", "TEXT", "", "", "10", "", "NULLABLE", "NON_REQUIRED", "")

# Process: Calculate Field (VALSTXT) (5)
arcpy.CalculateField_management(ftpconusfpu_1pt_shp__VALSTXT_, "VALSTXT", "[GRID_CODE]", "VB",
"")

# Process: Add Field (VALSTXTT) (5)
arcpy.AddField_management(ftpconusfpu_1pt_shp__VALSTXT_calc_, "VALSTXTT", "TEXT", "", "", "10",
"", "NULLABLE", "NON_REQUIRED", "")

# Process: Calculate Field (VALSTXTT) (5)
arcpy.CalculateField_management(ftpconusfpu_1pt_shp__VALSTXTT_, "VALSTXTT", "Right(
\"00000000\" &  CStr([VALSTXT]),8 )", "VB", "")

# Process: Delete Field (VALSTXT) (5)
arcpy.DeleteField_management(ftpconusfpu_1pt_shp__VALSTXTT_calc_, "VALSTXT")

# Process: Delete (5)
arcpy.Delete_management(FPU_Buffer_shp, "")

# Process: Clip (tesconus)
arcpy.Clip_management(tesconus, "-2354719.4099919 312634.153254099 2179660.5900081
3161944.1532541", tesconusfpu, FPU_Buffer_shp, "-2147483647", "ClippingGeometry",
"NO_MAINTAIN_EXTENT")

# Process: Raster to Point (tesconusfpu)
arcpy.RasterToPoint_conversion(tesconusfpu,
Output_FPU_Threatened___Endangered_Values_Shapefile__ex__CA_CA_001_Tes_shp_, "Value")

# Process: Add Field (VALSTXT) (6)
arcpy.AddField_management(Output_FPU_Threatened___Endangered_Values_Shapefile__ex__CA_CA_
001_Tes_shp_, "VALSTXT", "TEXT", "", "", "10", "", "NULLABLE", "NON_REQUIRED", "")

# Process: Calculate Field (VALSTXT) (6)
```

```
arcpy.CalculateField_management(teconusfpu_1pt_shp__VALSTXT_, "VALSTXT", "[GRID_CODE]", "VB",
"")

# Process: Add Field (VALSTXTT) (6)
arcpy.AddField_management(teconusfpu_1pt_shp__VALSTXT_calc_, "VALSTXTT", "TEXT", "", "", "10",
"", "NULLABLE", "NON_REQUIRED", "")

# Process: Calculate Field (VALSTXTT) (6)
arcpy.CalculateField_management(teconusfpu_1pt_shp__VALSTXTT_, "VALSTXTT", "Right(
\"00000000\" &  CStr([VALSTXT]),8 )", "VB", "")

# Process: Delete Field (VALSTXT) (6)
arcpy.DeleteField_management(teconusfpu_1pt_shp__VALSTXTT_calc_, "VALSTXT")

# Process: Delete (6)
arcpy.Delete_management(FPU_Buffer_shp, "")

# Process: Clip (dkwconus)
arcpy.Clip_management(dkwconus, "-2354719.4099919 312634.153254099 2179660.5900081
3161944.1532541", dkwconusfpu, FPU_Buffer_shp, "-2147483647", "ClippingGeometry",
"NO_MAINTAIN_EXTENT")

# Process: Raster to Point (dkwconusfpu)
arcpy.RasterToPoint_conversion(dkwconusfpu,
Output_FPU_Drinking_Water_Values_Shapefile__ex__CA_CA_001_Dkw_shp_, "Value")

# Process: Add Field (VALSTXT) (7)
arcpy.AddField_management(Output_FPU_Drinking_Water_Values_Shapefile__ex__CA_CA_001_Dkw_s
hp_, "VALSTXT", "TEXT", "", "", "10", "", "NULLABLE", "NON_REQUIRED", "")

# Process: Calculate Field (VALSTXT) (7)
arcpy.CalculateField_management(dkwconusfpu_1pt_shp__VALSTXT_, "VALSTXT", "[GRID_CODE]",
"VB", "")

# Process: Add Field (VALSTXTT) (7)
arcpy.AddField_management(dkwconusfpu_1pt_shp__VALSTXT_calc_, "VALSTXTT", "TEXT", "", "", "10",
"", "NULLABLE", "NON_REQUIRED", "")

# Process: Calculate Field (VALSTXTT) (7)
arcpy.CalculateField_management(dkwconusfpu_1pt_shp__VALSTXTT_, "VALSTXTT", "Right(
\"00000000\" &  CStr([VALSTXT]),8 )", "VB", "")

# Process: Delete Field (VALSTXT) (7)
arcpy.DeleteField_management(dkwconusfpu_1pt_shp__VALSTXTT_calc_, "VALSTXT")

# Process: Delete (7)
arcpy.Delete_management(FPU_Buffer_shp, "")
```

```
# Process: Clip (sagconus)
arcpy.Clip_management(sagconus, "-2354719.4099919 312634.153254099 2179660.5900081
3161944.1532541", sagconusfpu, FPU_Buffer_shp, "-2147483647", "ClippingGeometry",
"NO_MAINTAIN_EXTENT")

# Process: Raster to Point (sagconusfpu)
arcpy.RasterToPoint_conversion(sagconusfpu,
Output_FPU_Sage_Grouse_Values_Shapefile__ex__CA_CA_001_Sag_shp_, "Value")

# Process: Add Field (VALSTXT) (8)
arcpy.AddField_management(Output_FPU_Sage_Grouse_Values_Shapefile__ex__CA_CA_001_Sag_shp
_, "VALSTXT", "TEXT", "", "", "10", "", "NULLABLE", "NON_REQUIRED", "")

# Process: Calculate Field (VALSTXT) (8)
arcpy.CalculateField_management(sagconusfpu_1pt_shp__VALSTXT_, "VALSTXT", "[GRID_CODE]", "VB",
"")

# Process: Add Field (VALSTXTT) (8)
arcpy.AddField_management(sagconusfpu_1pt_shp__VALSTXT_calc_, "VALSTXTT", "TEXT", "", "", "10",
"", "NULLABLE", "NON_REQUIRED", "")

# Process: Calculate Field (VALSTXTT) (8)
arcpy.CalculateField_management(sagconusfpu_1pt_shp__VALSTXTT_, "VALSTXTT", "Right(
\"00000000\" &  CStr([VALSTXT]),8 )", "VB", "")

# Process: Delete Field (VALSTXT) (8)
arcpy.DeleteField_management(sagconusfpu_1pt_shp__VALSTXTT_calc_, "VALSTXT")

# Process: Delete (8)
arcpy.Delete_management(FPU_Buffer_shp, "")
```

Appendix D

SAS Program for Step 2:

The code for this analysis was generated using SAS software, Version 9.3 of the SAS System for Windows. Copyright © 2002-2010 SAS Institute Inc. SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc., Cary, NC, USA.

```
%MACRO readfil(FPU);
data null1;
      format GA $20.;
      if substr("&fpu",1,2)="AK" then GA="Alaska";
      if substr("&fpu",1,2)="CA" then GA="California";
      if substr("&fpu",1,2)="EA" then GA="Eastern_Area";
      if substr("&fpu",1,2)="GB" then GA="Great_Basin";
      if substr("&fpu",1,2)="NR" then GA="Northern_Rockies";
      if substr("&fpu",1,2)="NW" then GA="Northwest";
      if substr("&fpu",1,2)="RM" then GA="Rocky_Mountain";
      if substr("&fpu",1,2)="SA" then GA="Southern_Area";
      if substr("&fpu",1,2)="SW" then GA="Southwest";
      in1=
CATS("S:\Phase_2\5100_Fire_Management\GIS\6_Year_2015_Analysis\",ga,"_GA\","&
fpu","\FSim_Results\","&fpu","_standard_results\","&fpu","_STANDARD_FLP.txt")
;
      dirnm=CATS("S:\Phase_2\5100_Fire_Management\GIS\2_GISProjects\Risk-
Based_Wildland_Fire_Mgmt\Documents\Tables\",ga,"_GA\","&fpu");
      out1 =CATS(dirnm,"\","&fpu","_FIL.dbf");
      call symputx('in1',in1);
      call symputx('dirnm',dirnm);
      call symputx('out1',out1);
run;
DATA std_fil;
INFILE "&in1" DLM='2C'x FIRSTOBS=2 DSD;
      INPUT xpos ypos pburn fil1 fil2 fil3 fil4 fil5 fil6;
RUN;
data std_fil2;
      set std_fil;
      abv4ft=fil3+fil4+fil5+fil6;if abv4ft>1 then abv4ft=1;
      bel4ft=fil1+fil2;if bel4ft>1 then bel4ft=1;
      pbabv4ft=pburn*abv4ft;
      pbbel4ft=pburn*bel4ft;
run;
OPTIONS NOXWAIT;
DATA _NULL_;
         X "md &dirnm";
RUN;
PROC EXPORT DATA= std_fil2
            OUTFILE="&out1"
            DBMS=DBF REPLACE;
RUN;
proc datasets lib=work kill nolist memtype=data;
quit;
%MEND readfil;
/*AK GA*/
%readfil(AK_AK_001);
```

```
/*CA GA*/
%readfil(CA_CA_001);%readfil(CA_CA_002);%readfil(CA_CA_003);%readfil(CA_CA_00
4);%readfil(CA_CA_005);%readfil(CA_CA_006);%readfil(CA_CA_007);%readfil(CA_CA
_008);%readfil(CA_CA_009);%readfil(CA_CA_010);%readfil(CA_CA_011);%readfil(CA
_CA_012);%readfil(CA_CA_013);%readfil(CA_CA_014);%readfil(CA_CA_015);
/*EA GA*/
%readfil(EA_IA_001);%readfil(EA_IL_001);%readfil(EA_IN_001);%readfil(EA_MI_00
1);%readfil(EA_MI_002);%readfil(EA_MN_001);%readfil(EA_MN_002);%readfil(EA_MO
_001);%readfil(EA_NH_001);%readfil(EA_NJ_001);%readfil(EA_OH_001);%readfil(EA
_PA_001);%readfil(EA_WI_001);%readfil(EA_WI_002);%readfil(EA_WV_001);
/*GB GA*/
%readfil(GB_ID_001);%readfil(GB_ID_002);%readfil(GB_ID_003);%readfil(GB_ID_00
4);%readfil(GB_NV_001);%readfil(GB_NV_002);%readfil(GB_NV_003);%readfil(GB_NV
_004);%readfil(GB_NV_005);%readfil(GB_NV_006);%readfil(GB_UT_001);%readfil(GB
_UT_002);%readfil(GB_UT_003);%readfil(GB_UT_004);%readfil(GB_UT_005);%readfil
(GB_WY_001);
/*NR GA*/
%readfil(NR_ID_001);%readfil(NR_MT_001);%readfil(NR_MT_002);%readfil(NR_MT_00
3);%readfil(NR_MT_004);%readfil(NR_MT_005);%readfil(NR_MT_006);%readfil(NR_MT
_007);%readfil(NR_MT_008);%readfil(NR_MT_009);%readfil(NR_MT_010);%readfil(NR
_MT_011);%readfil(NR_ND_001);
/*NW GA*/
%readfil(NW_OR_001);%readfil(NW_OR_002);%readfil(NW_OR_003);%readfil(NW_OR_00
4);%readfil(NW_OR_005);%readfil(NW_OR_006);%readfil(NW_OR_007);%readfil(NW_OR
_008);%readfil(NW_OR_009);%readfil(NW_OR_010);%readfil(NW_OR_011);%readfil(NW
_WA_001);%readfil(NW_WA_002);%readfil(NW_WA_003);%readfil(NW_WA_004);%readfil
(NW_WA_005);%readfil(NW_WA_007);%readfil(NW_WA_008);
/*RM GA*/
%readfil(RM_CO_001);%readfil(RM_CO_002);%readfil(RM_CO_003);%readfil(RM_CO_00
4);%readfil(RM_CO_005);%readfil(RM_CO_006);%readfil(RM_CO_007);%readfil(RM_CO
_008);%readfil(RM_KS_001);%readfil(RM_NE_001);%readfil(RM_SD_001);%readfil(RM
_SD_002);%readfil(RM_SD_002);%readfil(RM_SD_003);%readfil(RM_WY_002);%readfil
(RM_WY_003);
/*SA GA*/
%readfil(SA_AL_001);%readfil(SA_AR_001);%readfil(SA_FL_001);%readfil(SA_FL_00
2);%readfil(SA_FL_003);%readfil(SA_FL_004);%readfil(SA_GA_001);%readfil(SA_KY
_001);%readfil(SA_LA_001);%readfil(SA_LA_003);%readfil(SA_MD_001);%readfil(SA
_MS_001);%readfil(SA_MS_002);%readfil(SA_NC_001);%readfil(SA_NC_002);%readfil
(SA_OK_001);%readfil(SA_OK_002);%readfil(SA_OK_003);%readfil(SA_OK_005);%read
fil(SA_PR_001);%readfil(SA_SC_001);%readfil(SA_TN_001);%readfil(SA_TX_001);%r
eadfil(SA_TX_002);%readfil(SA_TX_003);%readfil(SA_TX_004);%readfil(SA_TX_005)
;%readfil(SA_VA_001);
/*SW GA*/
%readfil(SW_AZ_001);%readfil(SW_AZ_002);%readfil(SW_AZ_003);%readfil(SW_AZ_00
4);%readfil(SW_AZ_005);%readfil(SW_AZ_006);%readfil(SW_NM_001);%readfil(SW_NM
_002);%readfil(SW_NM_003);%readfil(SW_NM_004);%readfil(SW_NM_005);%readfil(SW
_NM_007);%readfil(SW_TX_002);%readfil(SW_TX_004);
```

Python Script for Step 2:

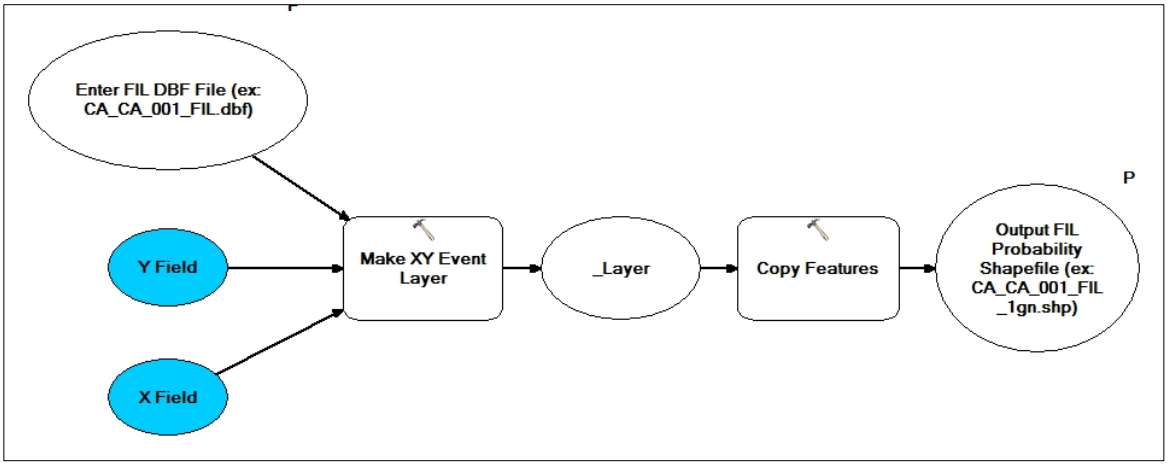


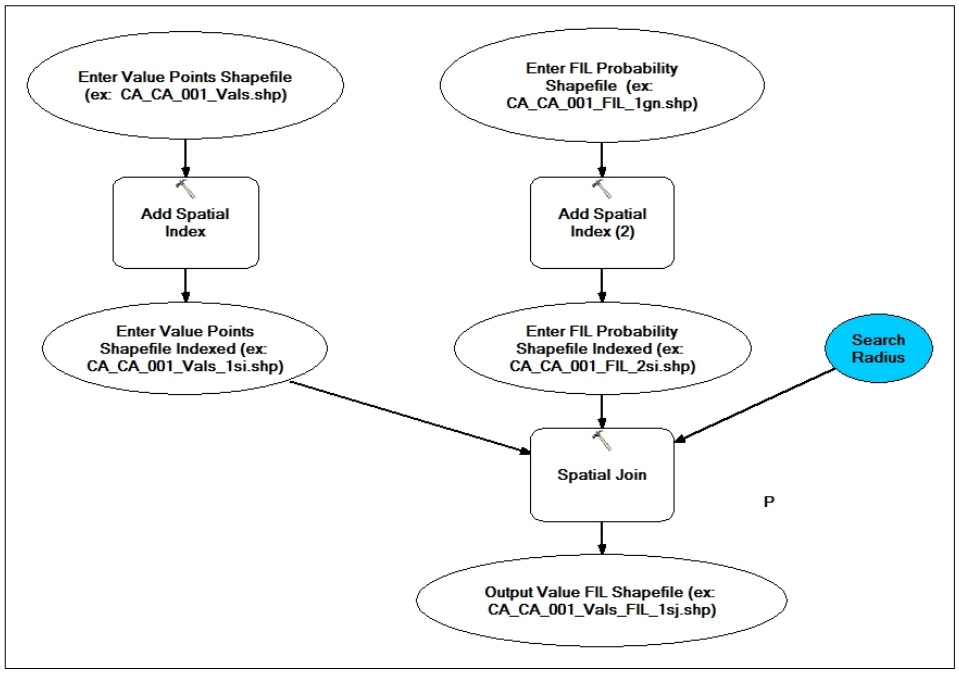
Figure A- 3: Step 2 process flow diagram - FIL DBF to point feature shapefile

```
# ---------------------------------------------------------------------------
# FIL_DBF_to_FIL_Prob_Shapefile.py
# Created on: 2015-02-06 11:33:29.00000
# Usage: FIL_DBF_to_FIL_Prob_Shapefile <Enter_FIL_DBF_File__ex__CA_CA_001_FIL_dbf_>
<Output_FIL_Probability_Shapefile__ex__CA_CA_001_FIL_1gn_shp_>
# Description: FIL DBF file to Point Feature Shapefile
# ---------------------------------------------------------------------------

# Import arcpy module
import arcpy

# Script arguments
Enter_FIL_DBF_File__ex__CA_CA_001_FIL_dbf_ = arcpy.GetParameterAsText(0)

Output_FIL_Probability_Shapefile__ex__CA_CA_001_FIL_1gn_shp_ = arcpy.GetParameterAsText(1)
if Output_FIL_Probability_Shapefile__ex__CA_CA_001_FIL_1gn_shp_ == '#' or not
Output_FIL_Probability_Shapefile__ex__CA_CA_001_FIL_1gn_shp_:
    Output_FIL_Probability_Shapefile__ex__CA_CA_001_FIL_1gn_shp_ = "in_memory\\CopyFeatures" #
provide a default value if unspecified

# Local variables:
X_Field = "xpos"
Y_Field = "ypos"
v_Layer = Enter_FIL_DBF_File__ex__CA_CA_001_FIL_dbf_

# Process: Make XY Event Layer
arcpy.MakeXYEventLayer_management(Enter_FIL_DBF_File__ex__CA_CA_001_FIL_dbf_, X_Field,
Y_Field, v_Layer,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
```

```
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],PARAMETER['Central_Meridian',-96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Latitude_Of_Origin',23.0],UNIT['Meter',1.0]];-16901100 -6972200 10000;-100000 10000;-100000 10000;0.001;0.001;0.001;IsHighPrecision", "")

# Process: Copy Features
arcpy.CopyFeatures_management(v_Layer, Output_FIL_Probability_Shapefile__ex__CA_CA_001_FIL_1gn_shp_, "", "0", "0", "0")
```

Appendix E

Python Script for Step 3:




```
# ---------------------------------------------------------------------------
# SpatIndx_SpatJoin_ValPts_to_FIL_Probs.py
# Created on: 2015-02-06 11:34:26.00000
# Usage: SpatIndx_SpatJoin_ValPts_to_FIL_Probs
<Enter_Value_Points_Shapefile__ex___CA_CA_001_Vals_shp_>
<Enter_FIL_Probability_Shapefile___ex___CA_CA_001_FIL_1gn_shp_>
<Output_Value_FIL_Shapefile__ex___CA_CA_001_Vals_FIL_1sj_shp_>
# Description: Perform Spatial Index and Spatial Join NLPR points to FIL Probability Points
# ---------------------------------------------------------------------------

# Import arcpy module
import arcpy

# Script arguments
Enter_Value_Points_Shapefile__ex___CA_CA_001_Vals_shp_ = arcpy.GetParameterAsText(0)

Enter_FIL_Probability_Shapefile___ex___CA_CA_001_FIL_1gn_shp_ = arcpy.GetParameterAsText(1)

Output_Value_FIL_Shapefile__ex___CA_CA_001_Vals_FIL_1sj_shp_ = arcpy.GetParameterAsText(2)

# Local variables:
```

```
Enter_Value_Points_Shapefile_Indexed__ex___CA_CA_001_Vals_1si_shp_ =
Enter_Value_Points_Shapefile__ex___CA_CA_001_Vals_shp_
Enter_FIL_Probability_Shapefile_Indexed__ex___CA_CA_001_FIL_2si_shp_ =
Enter_FIL_Probability_Shapefile___ex___CA_CA_001_FIL_1gn_shp_
Search_Radius = "150 Meters"

# Process: Add Spatial Index
arcpy.AddSpatialIndex_management(Enter_Value_Points_Shapefile__ex___CA_CA_001_Vals_shp_, "0",
"0", "0")

# Process: Add Spatial Index (2)
arcpy.AddSpatialIndex_management(Enter_FIL_Probability_Shapefile___ex___CA_CA_001_FIL_1gn_shp
_, "0", "0", "0")

# Process: Spatial Join
arcpy.SpatialJoin_analysis(Enter_Value_Points_Shapefile_Indexed__ex___CA_CA_001_Vals_1si_shp_,
Enter_FIL_Probability_Shapefile_Indexed__ex___CA_CA_001_FIL_2si_shp_,
Output_Value_FIL_Shapefile__ex___CA_CA_001_Vals_FIL_1sj_shp_, "JOIN_ONE_TO_ONE", "KEEP_ALL",
"", "INTERSECT", Search_Radius, "")
```

Appendix F

SAS Program for Step 4:

```
%MACRO readfil(FPU);
data null1;
      format GA $20.;
      if substr("&fpu",1,2)="AK" then GA="Alaska";
      if substr("&fpu",1,2)="CA" then GA="California";
      if substr("&fpu",1,2)="EA" then GA="Eastern_Area";
      if substr("&fpu",1,2)="GB" then GA="Great_Basin";
      if substr("&fpu",1,2)="NR" then GA="Northern_Rockies";
      if substr("&fpu",1,2)="NW" then GA="Northwest";
      if substr("&fpu",1,2)="RM" then GA="Rocky_Mountains";
      if substr("&fpu",1,2)="SA" then GA="Southern_Area";
      if substr("&fpu",1,2)="SW" then GA="Southwest";
      in1= CATS("T:\2_GISProjects\Risk-
Based_Wildland_Fire_Mgmt\Data\Generated\Shapefiles_ver3\",ga,"_GA\","&fpu","\
","&fpu","_Vals_FIL_1sj.dbf");
      out1=CATS("T:\2_GISProjects\Risk-
Based_Wildland_Fire_Mgmt\Documents\Tables_ver6\",ga,"_GA\","&fpu","\","&fpu",
"_Val_.dbf");
      out2=CATS("T:\2_GISProjects\Risk-
Based_Wildland_Fire_Mgmt\Data\Generated\Shapefiles_ver6\",ga,"_GA\","&fpu","\
","&fpu","_Val__1gn.shp");
      call symputx('in1',in1);
      call symputx('out1',out1);
      call symputx('out2',out2);
run;
PROC IMPORT OUT=indbf
      DATAFILE="&in1"
      DBMS=DBF REPLACE;
RUN;
data indbf2 (keep=/*target_fid objectid pointid valstxt2*/ xpos ypos
            sag dkw tes ftp rip rec inf dev);
      set indbf;
      format grdcode $char8.;
      grdcode=valstxtt;
      if substr(grdcode,1,1)="1" then sage=1;
      if substr(grdcode,2,1)="1" then drink=1;
      if substr(grdcode,3,1)="1" then threat=1;
      if substr(grdcode,4,1)="1" then forest=1;
      if substr(grdcode,5,1)="1" then riparian=1;
      if substr(grdcode,6,1)="1" then recreation=1;
      if substr(grdcode,7,1)="1" then critical=1;
      if substr(grdcode,8,1)="1" then develop=1;
      sag=sage*sum(of fil1-fil6);
      dkw=drink*sum(of fil1-fil6);
      tes=threat*sum(of fil1-fil6);
      ftp=forest*sum(of fil1-fil6);
      rip=riparian*sum(of fil1-fil6);
      rec=recreation*sum(of fil1-fil6);
      inf=critical*sum(of fil1-fil6);
      dev=develop*sum(of fil1-fil6);
run;
data indbf3;
```

```
      set indbf2;
      sag=round(sag,1);
      dkw=round(dkw,1);
      tes=round(tes,1);
      ftp=round(ftp,1);
      rip=round(rip,1);
      rec=round(rec,1);
      inf=round(inf,1);
      dev=round(dev,1);
      if max(sag,dkw,tes,ftp,rip,rec,inf,dev)=0 then delete;
      else if max(sag,dkw,tes,ftp,rip,rec,inf,dev)=. then delete;
run;

PROC EXPORT DATA= indbf3
            OUTFILE= "&out1"
            DBMS=DBF REPLACE;
RUN;
proc datasets lib=work nolist memtype=data;
      delete indbf indbf2 indbf3 null1;
quit;
data arcmp2;
      step4_1="&out1";step4_2="&out2";
      output;
run;
proc datasets library=work nolist;
      append base=arcmp data=arcmp2;
run;quit;
%MEND readfil;
data arcmp;
format step4_1 $250.;
format step4_2 $250.;
run;quit;
/*AK GA*/
%readfil(AK_AK_001);
/*CA GA*/
%readfil(CA_CA_001);%readfil(CA_CA_002);%readfil(CA_CA_003);%readfil(CA_CA_00
4);%readfil(CA_CA_005);%readfil(CA_CA_006);%readfil(CA_CA_007);%readfil(CA_CA
_008);%readfil(CA_CA_009);%readfil(CA_CA_010);%readfil(CA_CA_011);%readfil(CA
_CA_012);%readfil(CA_CA_013);%readfil(CA_CA_014);%readfil(CA_CA_015);
/*EA GA*/
%readfil(EA_IA_001);%readfil(EA_IL_001);%readfil(EA_IN_001);%readfil(EA_MI_00
1);%readfil(EA_MI_002);%readfil(EA_MN_001);%readfil(EA_MN_002);%readfil(EA_MO
_001);%readfil(EA_NH_001);%readfil(EA_NJ_001);%readfil(EA_OH_001);%readfil(EA
_PA_001);%readfil(EA_WI_001);%readfil(EA_WI_002);%readfil(EA_WV_001);
/*GB GA*/
%readfil(GB_ID_001);%readfil(GB_ID_002);%readfil(GB_ID_003);%readfil(GB_ID_00
4);%readfil(GB_NV_001);%readfil(GB_NV_002);%readfil(GB_NV_003);%readfil(GB_NV
_004);%readfil(GB_NV_005);%readfil(GB_NV_006);%readfil(GB_UT_001);%readfil(GB
_UT_002);%readfil(GB_UT_003);%readfil(GB_UT_004);%readfil(GB_UT_005);%readfil
(GB_WY_001);
/*NR GA*/
%readfil(NR_ID_001);%readfil(NR_MT_001);%readfil(NR_MT_002);%readfil(NR_MT_00
3);%readfil(NR_MT_004);%readfil(NR_MT_005);%readfil(NR_MT_006);%readfil(NR_MT
_007);%readfil(NR_MT_008);%readfil(NR_MT_009);%readfil(NR_MT_010);%readfil(NR
_MT_011);%readfil(NR_ND_001);
/*NW GA*/
```

```
%readfil(NW_OR_001);%readfil(NW_OR_002);%readfil(NW_OR_003);%readfil(NW_OR_00
4);%readfil(NW_OR_005);%readfil(NW_OR_006);%readfil(NW_OR_007);%readfil(NW_OR
_008);%readfil(NW_OR_009);%readfil(NW_OR_010);%readfil(NW_OR_011);%readfil(NW
_WA_001);%readfil(NW_WA_002);%readfil(NW_WA_003);%readfil(NW_WA_004);%readfil
(NW_WA_005);%readfil(NW_WA_007);%readfil(NW_WA_008);
/*RM GA*/
%readfil(RM_CO_001);%readfil(RM_CO_002);%readfil(RM_CO_003);%readfil(RM_CO_00
4);%readfil(RM_CO_005);%readfil(RM_CO_006);%readfil(RM_CO_007);%readfil(RM_CO
_008);%readfil(RM_KS_001);%readfil(RM_NE_001);%readfil(RM_SD_001);%readfil(RM
_SD_002);%readfil(RM_SD_002);%readfil(RM_SD_003);%readfil(RM_WY_002);%readfil
(RM_WY_003);
/*SA GA*/
%readfil(SA_AL_001);%readfil(SA_AR_001);%readfil(SA_FL_001);%readfil(SA_FL_00
2);%readfil(SA_FL_003);%readfil(SA_FL_004);%readfil(SA_GA_001);%readfil(SA_KY
_001);%readfil(SA_LA_001);%readfil(SA_LA_003);%readfil(SA_MD_001);%readfil(SA
_MS_001);%readfil(SA_MS_002);%readfil(SA_NC_001);%readfil(SA_NC_002);%readfil
(SA_OK_001);%readfil(SA_OK_002);%readfil(SA_OK_003);%readfil(SA_OK_005);%read
fil(SA_PR_001);%readfil(SA_SC_001);%readfil(SA_TN_001);%readfil(SA_TX_001);%r
eadfil(SA_TX_002);%readfil(SA_TX_003);%readfil(SA_TX_004);%readfil(SA_TX_005)
;%readfil(SA_VA_001);
/*SW GA*/
%readfil(SW_AZ_001);%readfil(SW_AZ_002);%readfil(SW_AZ_003);%readfil(SW_AZ_00
4);%readfil(SW_AZ_005);%readfil(SW_AZ_006);%readfil(SW_NM_001);%readfil(SW_NM
_002);%readfil(SW_NM_003);%readfil(SW_NM_004);%readfil(SW_NM_005);
%readfil(SW_NM_007);%readfil(SW_TX_002);%readfil(SW_TX_004);
```

Python Script for Step 4:

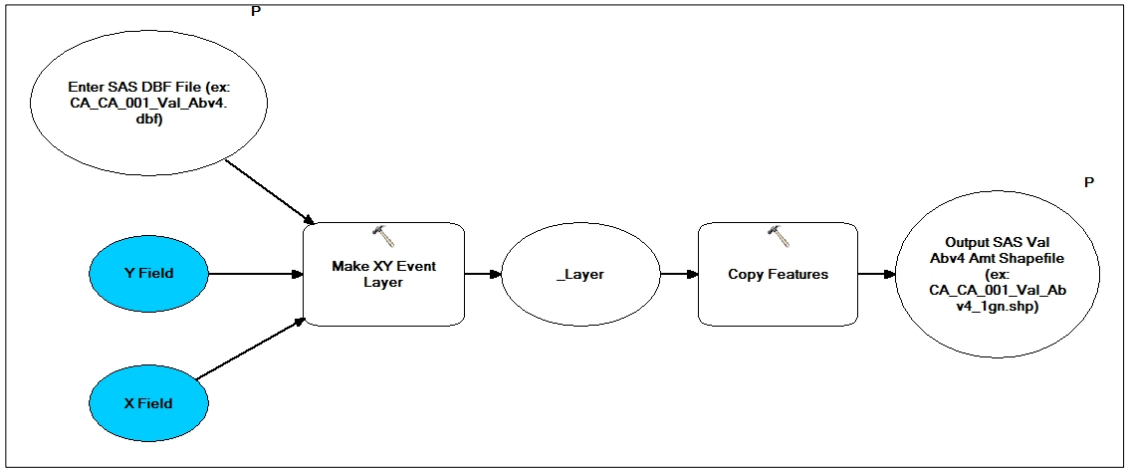


Figure A- 5: Step 4 process flow diagram - shapefile created from SAS DBF file

```
# ---------------------------------------------------------------------------
# SAS_DBF_to_Shapefile.py
# Created on: 2015-02-06 11:34:52.00000
# Usage: SAS_DBF_to_Shapefile <Enter_SAS_DBF_File__ex__CA_CA_001_Val_Abv4_dbf_>
<Output_SAS_Val_Abv4_Amt_Shapefile__ex__CA_CA_001_Val_Abv4_1gn_shp_>
# Description: Shapefile created from SAS DBF file
# ---------------------------------------------------------------------------
```

```
# Import arcpy module
import arcpy

# Script arguments
Enter_SAS_DBF_File__ex__CA_CA_001_Val_Abv4_dbf_ = arcpy.GetParameterAsText(0)

Output_SAS_Val_Abv4_Amt_Shapefile__ex__CA_CA_001_Val_Abv4_1gn_shp_ =
arcpy.GetParameterAsText(1)
if Output_SAS_Val_Abv4_Amt_Shapefile__ex__CA_CA_001_Val_Abv4_1gn_shp_ == '#' or not
Output_SAS_Val_Abv4_Amt_Shapefile__ex__CA_CA_001_Val_Abv4_1gn_shp_:
    Output_SAS_Val_Abv4_Amt_Shapefile__ex__CA_CA_001_Val_Abv4_1gn_shp_ =
"in_memory\\CopyFeatures" # provide a default value if unspecified

# Local variables:
X_Field = "xpos"
Y_Field = "ypos"
v_Layer = Enter_SAS_DBF_File__ex__CA_CA_001_Val_Abv4_dbf_

# Process: Make XY Event Layer
arcpy.MakeXYEventLayer_management(Enter_SAS_DBF_File__ex__CA_CA_001_Val_Abv4_dbf_,
X_Field, Y_Field, v_Layer,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]];-16901100 -6972200 10000;-100000 10000;-100000
10000;0.001;0.001;0.001;IsHighPrecision", "")

# Process: Copy Features
arcpy.CopyFeatures_management(v_Layer,
Output_SAS_Val_Abv4_Amt_Shapefile__ex__CA_CA_001_Val_Abv4_1gn_shp_, "", "0", "0", "0")
```

Appendix G

Python Script for Step 5 and 6:

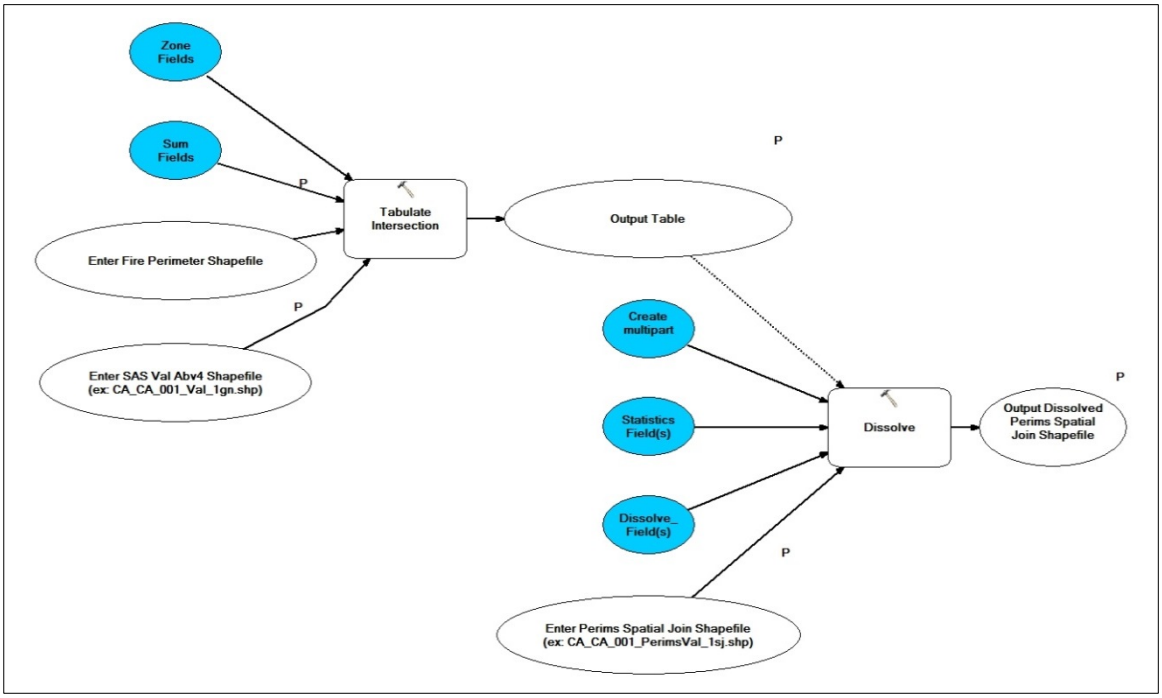


Figure A- 6: Step 5/6 process flow diagram - Tabular Intersect of FSim modeled Fire Perimeters and associated Fire Start Location point to NLPR point features (single entry)

```
# ---------------------------------------------------------------------------

# 5_6_TabularInteresect_FSimPerims_to SAS_ValPts_wValAmts.py

# Created on: 2015-04-28 16:56:51.00000

# Usage: 5_6_TabularInteresect_FSimPerims_to SAS_ValPts_wValAmts
<Enter_Fire_Perimeter_Shapefile> <Enter_SAS_Val_Abv4_Shapefile__ex__CA_CA_001_Val_1gn_shp_>
<Output_Table>
<Output_Dissolved_Perims_Spatial_Join_Shapefile__ex__CA_CA_001_PerimsVal_2dv_shp_>
<Enter_Perims_Spatial_Join_Shapefile__ex__CA_CA_001_PerimsVal_1sj_shp_>

# Description:

# ---------------------------------------------------------------------------

# Set the necessary product code

# import arcinfo
```

```
# Import arcpy module

import arcpy

# Script arguments

Enter_Fire_Perimeter_Shapefile = arcpy.GetParameterAsText(0)

Enter_SAS_Val_Abv4_Shapefile__ex__CA_CA_001_Val_1gn_shp_ = arcpy.GetParameterAsText(1)

Output_Table = arcpy.GetParameterAsText(2)

Output_Dissolved_Perims_Spatial_Join_Shapefile__ex__CA_CA_001_PerimsVal_2dv_shp_ =
arcpy.GetParameterAsText(3)

Enter_Perims_Spatial_Join_Shapefile__ex__CA_CA_001_PerimsVal_1sj_shp_ =
arcpy.GetParameterAsText(4)


# Local variables:

Sum_Fields = "sg;dw;te;ft;rp;rc;nf;dv"

Zone_Fields = "FID"

Dissolve_Field_s_ = "FIRE_NUMBE;Xcoord;Ycoord"

Statistics_Field_s_ = "dv SUM;nf SUM;rc SUM;rp SUM;ft SUM;te SUM;dw SUM;sg SUM"

Create_multipart_features = "true"

# Process: Tabulate Intersection

arcpy.TabulateIntersection_analysis(Enter_Fire_Perimeter_Shapefile, Zone_Fields,
Enter_SAS_Val_Abv4_Shapefile__ex__CA_CA_001_Val_1gn_shp_, Output_Table, "", Sum_Fields, "",
"UNKNOWN")

# Process: Dissolve

arcpy.Dissolve_management(Enter_Perims_Spatial_Join_Shapefile__ex__CA_CA_001_PerimsVal_1sj_sh
p_, Output_Dissolved_Perims_Spatial_Join_Shapefile__ex__CA_CA_001_PerimsVal_2dv_shp_,
Dissolve_Field_s_, Statistics_Field_s_, Create_multipart_features, "DISSOLVE_LINES")
```

Appendix I

Python Script for Step 7:

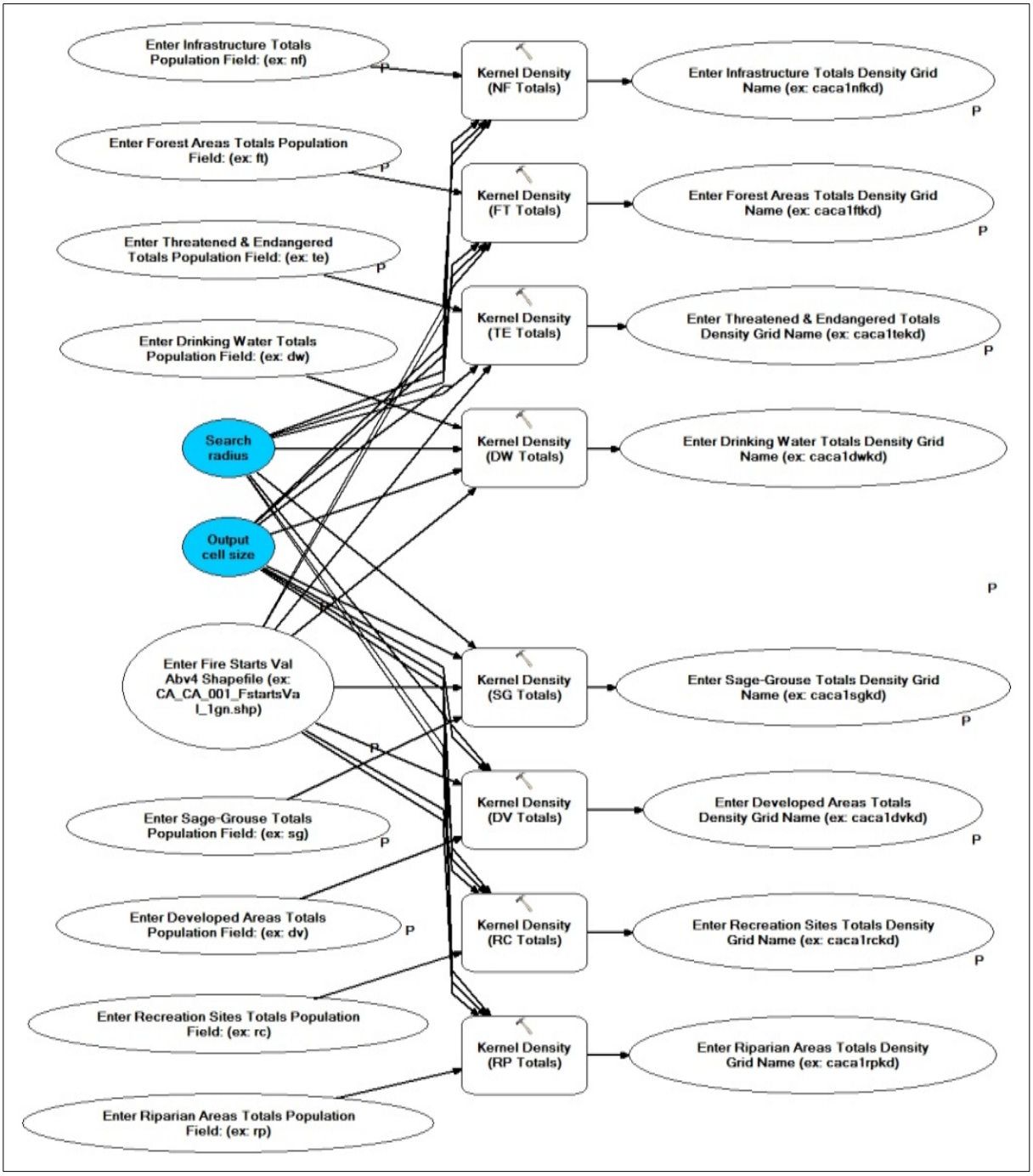


Figure A- 7: Step 7 process flow diagram - create density grid for each NLPR category

```
# ----------------------------------------------------------------------------
# Create_DenGrids_ValCat.py
# Created on: 2015-02-06 11:37:32.00000
```

```
# Usage: Create_DenGrids_ValCat
<Enter_Fire_Starts_Val_Abv4_Shapefile__ex__CA_CA_001_FstartsVal_1gn_shp_>
<Enter_Developed_Areas_Totals_Population_Field___ex__dv_>
<Enter_Developed_Areas_Totals_Density_Grid_Name__ex__caca1dvkd_>
<Enter_Infrastructure_Totals_Population_Field___ex__nf_>
<Enter_Infrastructure_Totals_Density_Grid_Name__ex__caca1nfkd_>
<Enter_Recreation_Sites_Totals_Population_Field___ex__rc_>
<Enter_Recreation_Sites_Totals_Density_Grid_Name__ex__caca1rckd_>
<Enter_Riparian_Areas_Totals_Population_Field___ex__rp_>
<Enter_Riparian_Areas_Totals_Density_Grid_Name__ex__caca1rpkd_>
<Enter_Forest_Areas_Totals_Population_Field___ex__ft_>
<Enter_Forest_Areas_Totals_Density_Grid_Name__ex__caca1ftkd_>
<Enter_Threatened___Endangered_Totals_Population_Field___ex__te_>
<Enter_Threatened___Endangered_Totals_Density_Grid_Name__ex__caca1tekd_>
<Enter_Drinking_Water_Totals_Population_Field___ex__dw_>
<Enter_Drinking_Water_Totals_Density_Grid_Name__ex__caca1dwkd_> <Enter_Sage-
Grouse_Totals_Population_Field___ex__sg_> <Enter_Sage-
Grouse_Totals_Density_Grid_Name__ex__caca1sgkd_>
# Description: Create Density Grid for each NLPR category
# ---------------------------------------------------------------------------

# Import arcpy module
import arcpy

# Check out any necessary licenses
arcpy.CheckOutExtension("spatial")

# Script arguments
Enter_Fire_Starts_Val_Abv4_Shapefile__ex__CA_CA_001_FstartsVal_1gn_shp_ =
arcpy.GetParameterAsText(0)

Enter_Developed_Areas_Totals_Population_Field___ex__dv_ = arcpy.GetParameterAsText(1)

Enter_Developed_Areas_Totals_Density_Grid_Name__ex__caca1dvkd_ = arcpy.GetParameterAsText(2)

Enter_Infrastructure_Totals_Population_Field___ex__nf_ = arcpy.GetParameterAsText(3)

Enter_Infrastructure_Totals_Density_Grid_Name__ex__caca1nfkd_ = arcpy.GetParameterAsText(4)

Enter_Recreation_Sites_Totals_Population_Field___ex__rc_ = arcpy.GetParameterAsText(5)

Enter_Recreation_Sites_Totals_Density_Grid_Name__ex__caca1rckd_ = arcpy.GetParameterAsText(6)

Enter_Riparian_Areas_Totals_Population_Field___ex__rp_ = arcpy.GetParameterAsText(7)

Enter_Riparian_Areas_Totals_Density_Grid_Name__ex__caca1rpkd_ = arcpy.GetParameterAsText(8)

Enter_Forest_Areas_Totals_Population_Field___ex__ft_ = arcpy.GetParameterAsText(9)
```

```
Enter_Forest_Areas_Totals_Density_Grid_Name__ex__caca1ftkd_ = arcpy.GetParameterAsText(10)

Enter_Threatened___Endangered_Totals_Population_Field___ex__te_ = arcpy.GetParameterAsText(11)

Enter_Threatened___Endangered_Totals_Density_Grid_Name__ex__caca1tekd_ =
arcpy.GetParameterAsText(12)

Enter_Drinking_Water_Totals_Population_Field___ex__dw_ = arcpy.GetParameterAsText(13)

Enter_Drinking_Water_Totals_Density_Grid_Name__ex__caca1dwkd_ = arcpy.GetParameterAsText(14)

Enter_Sage-Grouse_Totals_Population_Field___ex__sg_ = arcpy.GetParameterAsText(15)

Enter_Sage-Grouse_Totals_Density_Grid_Name__ex__caca1sgkd_ = arcpy.GetParameterAsText(16)

# Local variables:
Output_cell_size = "270"
Search_radius = "10000"

# Process: Kernel Density (RP Totals)
arcpy.gp.KernelDensity_sa(Enter_Fire_Starts_Val_Abv4_Shapefile__ex__CA_CA_001_FstartsVal_1gn_sh
p_, Enter_Riparian_Areas_Totals_Population_Field___ex__rp_,
Enter_Riparian_Areas_Totals_Density_Grid_Name__ex__caca1rpkd_, Output_cell_size, Search_radius,
"SQUARE_MAP_UNITS")

# Process: Kernel Density (RC Totals)
arcpy.gp.KernelDensity_sa(Enter_Fire_Starts_Val_Abv4_Shapefile__ex__CA_CA_001_FstartsVal_1gn_sh
p_, Enter_Recreation_Sites_Totals_Population_Field___ex__rc_,
Enter_Recreation_Sites_Totals_Density_Grid_Name__ex__caca1rckd_, Output_cell_size, Search_radius,
"SQUARE_MAP_UNITS")

# Process: Kernel Density (DV Totals)
arcpy.gp.KernelDensity_sa(Enter_Fire_Starts_Val_Abv4_Shapefile__ex__CA_CA_001_FstartsVal_1gn_sh
p_, Enter_Developed_Areas_Totals_Population_Field___ex__dv_,
Enter_Developed_Areas_Totals_Density_Grid_Name__ex__caca1dvkd_, Output_cell_size,
Search_radius, "SQUARE_MAP_UNITS")

# Process: Kernel Density (SG Totals)
arcpy.gp.KernelDensity_sa(Enter_Fire_Starts_Val_Abv4_Shapefile__ex__CA_CA_001_FstartsVal_1gn_sh
p_, Enter_Sage-Grouse_Totals_Population_Field___ex__sg_, Enter_Sage-
Grouse_Totals_Density_Grid_Name__ex__caca1sgkd_, Output_cell_size, Search_radius,
"SQUARE_MAP_UNITS")

# Process: Kernel Density (DW Totals)
arcpy.gp.KernelDensity_sa(Enter_Fire_Starts_Val_Abv4_Shapefile__ex__CA_CA_001_FstartsVal_1gn_sh
p_, Enter_Drinking_Water_Totals_Population_Field___ex__dw_,
```

```
Enter_Drinking_Water_Totals_Density_Grid_Name__ex__caca1dwkd_, Output_cell_size, Search_radius, "SQUARE_MAP_UNITS")

# Process: Kernel Density (TE Totals)
arcpy.gp.KernelDensity_sa(Enter_Fire_Starts_Val_Abv4_Shapefile__ex__CA_CA_001_FstartsVal_1gn_shp_, Enter_Threatened___Endangered_Totals_Population_Field___ex__te_, Enter_Threatened___Endangered_Totals_Density_Grid_Name__ex__caca1tekd_, Output_cell_size, Search_radius, "SQUARE_MAP_UNITS")

# Process: Kernel Density (FT Totals)
arcpy.gp.KernelDensity_sa(Enter_Fire_Starts_Val_Abv4_Shapefile__ex__CA_CA_001_FstartsVal_1gn_shp_, Enter_Forest_Areas_Totals_Population_Field___ex__ft_, Enter_Forest_Areas_Totals_Density_Grid_Name__ex__caca1ftkd_, Output_cell_size, Search_radius, "SQUARE_MAP_UNITS")

# Process: Kernel Density (NF Totals)
arcpy.gp.KernelDensity_sa(Enter_Fire_Starts_Val_Abv4_Shapefile__ex__CA_CA_001_FstartsVal_1gn_shp_, Enter_Infrastructure_Totals_Population_Field___ex__nf_, Enter_Infrastructure_Totals_Density_Grid_Name__ex__caca1nfkd_, Output_cell_size, Search_radius, "SQUARE_MAP_UNITS")
```

Appendix J

Python Script for Step 8:

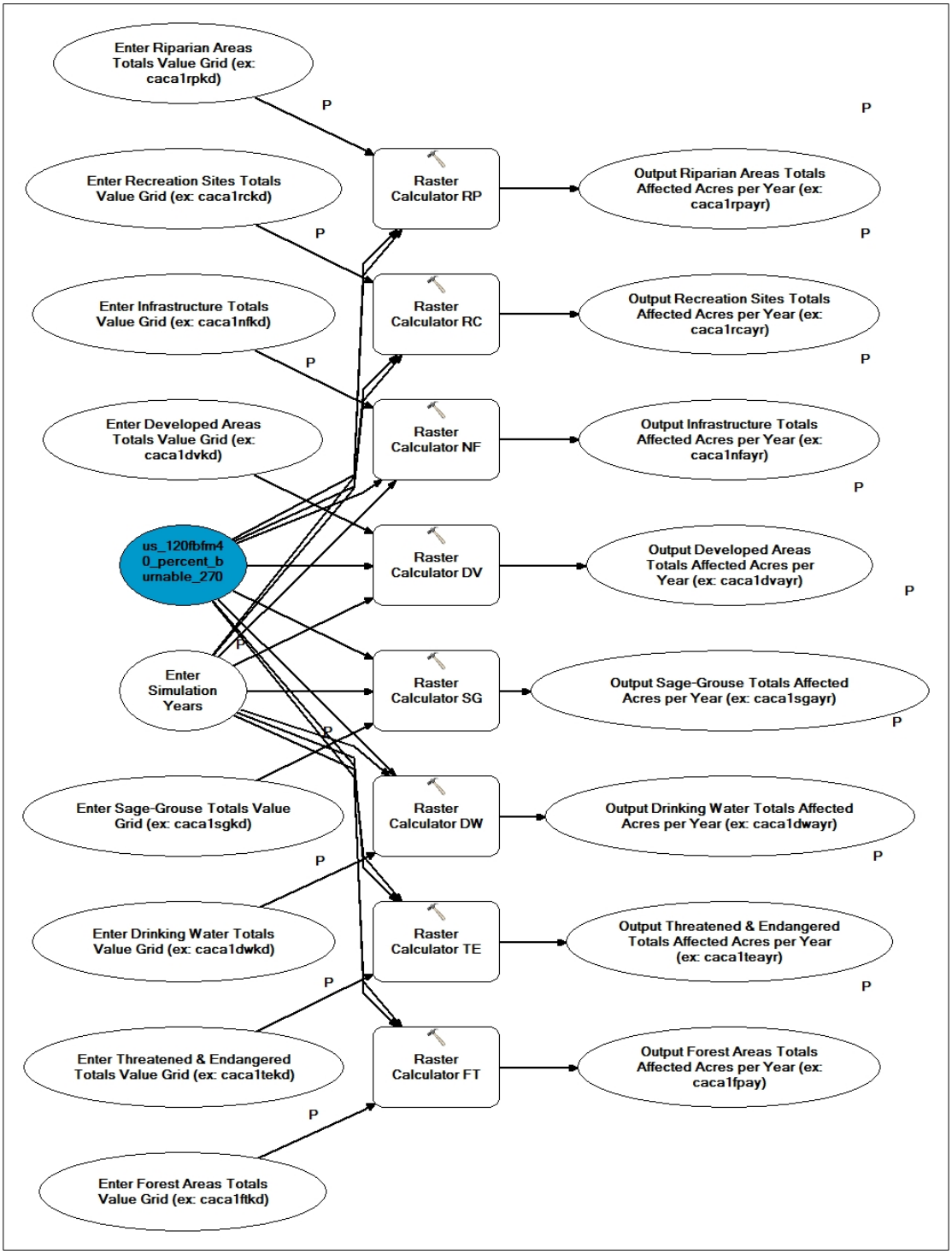


Figure A- 8: Step 8 process flow diagram - calculate NLPR acres burned per FSim simulation year

```
# ---------------------------------------------------------------------------

# 8b_Calc_ValAc_per_FSim_SimYr_wBurnPct.py

# Created on: 2015-04-28 17:05:22.00000

# Usage: 8b_Calc_ValAc_per_FSim_SimYr_wBurnPct <Enter_Simulation_Years>
<Enter_Developed_Areas_Totals_Value_Grid__ex__caca1dvkd_>
<Output_Developed_Areas_Totals_Affected_Acres_per_Year__ex__caca1dvayr_>
<Enter_Infrastructure_Totals_Value_Grid__ex__caca1nfkd_>
<Output_Infrastructure_Totals_Affected_Acres_per_Year__ex__caca1nfayr_>
<Enter_Recreation_Sites_Totals_Value_Grid__ex__caca1rckd_>
<Output_Recreation_Sites_Totals_Affected_Acres_per_Year__ex__caca1rcayr_>
<Enter_Riparian_Areas_Totals_Value_Grid__ex__caca1rpkd_>
<Output_Riparian_Areas_Totals_Affected_Acres_per_Year__ex__caca1rpayr_>
<Enter_Forest_Areas_Totals_Value_Grid__ex__caca1ftkd_>
<Output_Forest_Areas_Totals_Affected_Acres_per_Year__ex__caca1fpay_>
<Enter_Threatened___Endangered_Totals_Value_Grid__ex__caca1tekd_>
<Output_Threatened___Endangered_Totals_Affected_Acres_per_Year__ex__caca1teayr_>
<Enter_Drinking_Water_Totals_Value_Grid__ex__caca1dwkd_>
<Output_Drinking_Water_Totals_Affected_Acres_per_Year__ex__caca1dwayr_> <Enter_Sage-
Grouse_Totals_Value_Grid__ex__caca1sgkd_> <Output_Sage-
Grouse_Totals_Affected_Acres_per_Year__ex__caca1sgayr_>

# Description:

# ---------------------------------------------------------------------------

# Import arcpy module

import arcpy

# Check out any necessary licenses

arcpy.CheckOutExtension("spatial")

# Script arguments

Enter_Simulation_Years = arcpy.GetParameterAsText(0)

Enter_Developed_Areas_Totals_Value_Grid__ex__caca1dvkd_ = arcpy.GetParameterAsText(1)

Output_Developed_Areas_Totals_Affected_Acres_per_Year__ex__caca1dvayr_ =
arcpy.GetParameterAsText(2)

Enter_Infrastructure_Totals_Value_Grid__ex__caca1nfkd_ = arcpy.GetParameterAsText(3)

Output_Infrastructure_Totals_Affected_Acres_per_Year__ex__caca1nfayr_ =
arcpy.GetParameterAsText(4)
```

```
Enter_Recreation_Sites_Totals_Value_Grid__ex__caca1rckd_ = arcpy.GetParameterAsText(5)

Output_Recreation_Sites_Totals_Affected_Acres_per_Year__ex__caca1rcayr_ =
arcpy.GetParameterAsText(6)

Enter_Riparian_Areas_Totals_Value_Grid__ex__caca1rpkd_ = arcpy.GetParameterAsText(7)

Output_Riparian_Areas_Totals_Affected_Acres_per_Year__ex__caca1rpayr_ =
arcpy.GetParameterAsText(8)

Enter_Forest_Areas_Totals_Value_Grid__ex__caca1ftkd_ = arcpy.GetParameterAsText(9)

Output_Forest_Areas_Totals_Affected_Acres_per_Year__ex__caca1fpay_ =
arcpy.GetParameterAsText(10)

Enter_Threatened___Endangered_Totals_Value_Grid__ex__caca1tekd_ =
arcpy.GetParameterAsText(11)

Output_Threatened___Endangered_Totals_Affected_Acres_per_Year__ex__caca1teayr_ =
arcpy.GetParameterAsText(12)

Enter_Drinking_Water_Totals_Value_Grid__ex__caca1dwkd_ = arcpy.GetParameterAsText(13)

Output_Drinking_Water_Totals_Affected_Acres_per_Year__ex__caca1dwayr_ =
arcpy.GetParameterAsText(14)

Enter_Sage-Grouse_Totals_Value_Grid__ex__caca1sgkd_ = arcpy.GetParameterAsText(15)

Output_Sage-Grouse_Totals_Affected_Acres_per_Year__ex__caca1sgayr_ =
arcpy.GetParameterAsText(16)

# Local variables:

us_120fbfm40_percent_burnable_270m =
"S:\\Phase_2\\5100_Fire_Management\\GIS\\2_GISProjects\\Risk-
Based_Wildland_Fire_Mgmt\\Data\\Grids\\PercentBurnable\\PercentBurnable.gdb\\us_120fbfm40_pe
rcent_burnable_270m"

# Process: Raster Calculator FT

arcpy.gp.RasterCalculator_sa("(\"%Enter Forest Areas Totals Value Grid (ex: caca1ftkd)%\" *
Square(270) * 18.0139)/%Enter Simulation Years% * \"%us_120fbfm40_percent_burnable_270m%\" /
0.89", Output_Forest_Areas_Totals_Affected_Acres_per_Year__ex__caca1fpay_)

# Process: Raster Calculator TE

arcpy.gp.RasterCalculator_sa("(\"%Enter Threatened & Endangered Totals Value Grid (ex: caca1tekd)%\"
* Square(270) * 18.0139)/%Enter Simulation Years% * \"%us_120fbfm40_percent_burnable_270m%\" /
0.89", Output_Threatened___Endangered_Totals_Affected_Acres_per_Year__ex__caca1teayr_)

# Process: Raster Calculator DW
```

```
arcpy.gp.RasterCalculator_sa("(\"%Enter Drinking Water Totals Value Grid (ex: caca1dwkd)%\" *
Square(270) * 18.0139)/%Enter Simulation Years% * \"%us_120fbfm40_percent_burnable_270m%\" /
0.89", Output_Drinking_Water_Totals_Affected_Acres_per_Year__ex__caca1dwayr_)

# Process: Raster Calculator SG

arcpy.gp.RasterCalculator_sa("(\"%Enter Sage-Grouse Totals Value Grid (ex: caca1sgkd)%\" *
Square(270) * 18.0139)/%Enter Simulation Years% * \"%us_120fbfm40_percent_burnable_270m%\" /
0.89", Output_Sage-Grouse_Totals_Affected_Acres_per_Year__ex__caca1sgayr_)

# Process: Raster Calculator DV

arcpy.gp.RasterCalculator_sa("(\"%Enter Developed Areas Totals Value Grid (ex: caca1dvkd)%\" *
Square(270) * 18.0139)/%Enter Simulation Years% * \"%us_120fbfm40_percent_burnable_270m%\" /
0.89", Output_Developed_Areas_Totals_Affected_Acres_per_Year__ex__caca1dvayr_)

# Process: Raster Calculator NF

arcpy.gp.RasterCalculator_sa("(\"%Enter Infrastructure Totals Value Grid (ex: caca1nfkd)%\" *
Square(270) * 18.0139)/%Enter Simulation Years% * \"%us_120fbfm40_percent_burnable_270m%\" /
0.89", Output_Infrastructure_Totals_Affected_Acres_per_Year__ex__caca1nfayr_)

# Process: Raster Calculator RC

arcpy.gp.RasterCalculator_sa("(\"%Enter Recreation Sites Totals Value Grid (ex: caca1rckd)%\" *
Square(270) * 18.0139)/%Enter Simulation Years% * \"%us_120fbfm40_percent_burnable_270m%\" /
0.89", Output_Recreation_Sites_Totals_Affected_Acres_per_Year__ex__caca1rcayr_)

# Process: Raster Calculator RP

arcpy.gp.RasterCalculator_sa("(\"%Enter Riparian Areas Totals Value Grid (ex: caca1rpkd)%\" *
Square(270) * 18.0139)/%Enter Simulation Years% * \"%us_120fbfm40_percent_burnable_270m%\" /
0.89", Output_Riparian_Areas_Totals_Affected_Acres_per_Year__ex__caca1rpayr_)
```

Appendix K

Python Script for Step 9:

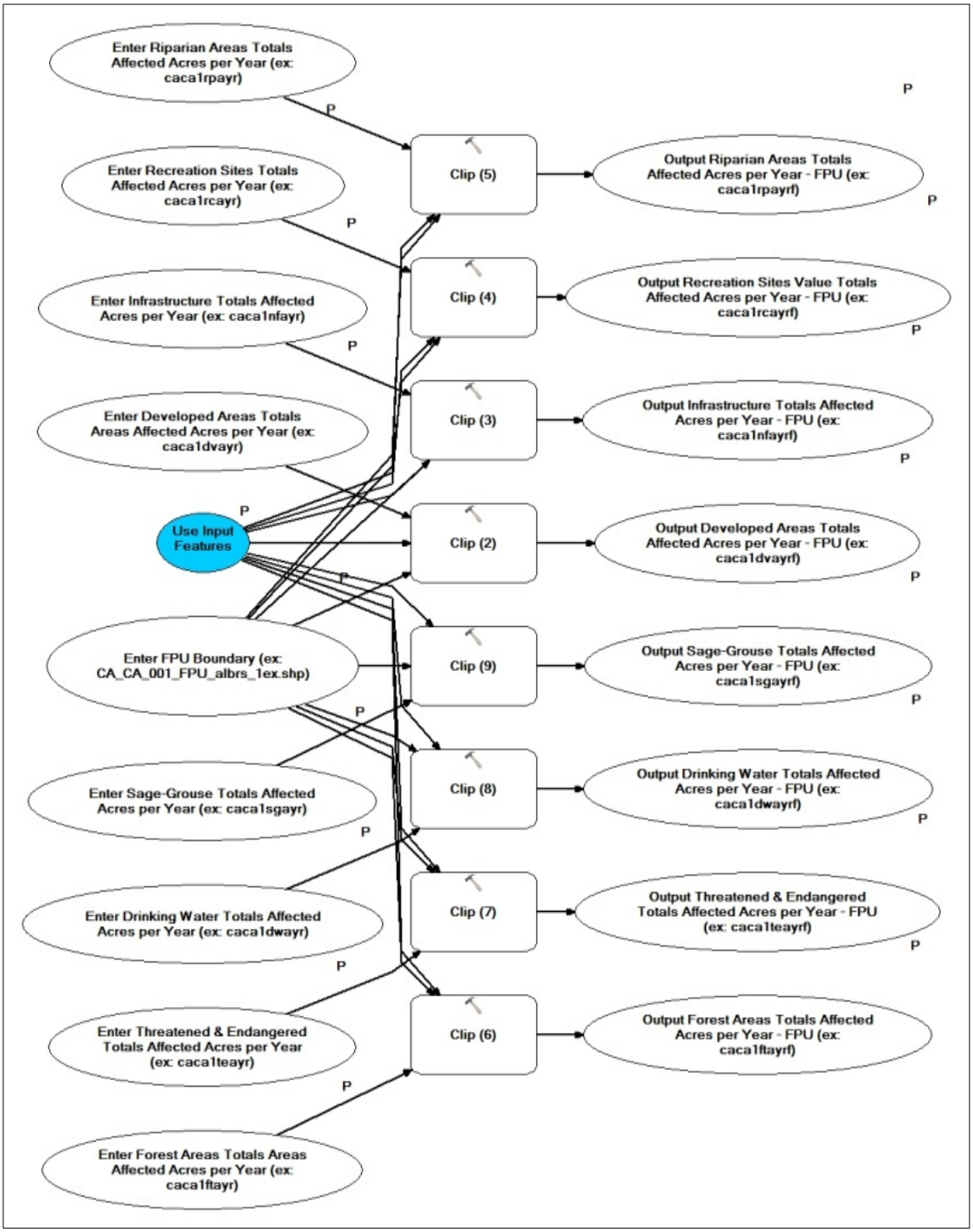


Figure A- 9: Step 9 process flow diagram - clip density grid to FPU boundary

```
# ---------------------------------------------------------------------------
# Clip_DenGridVal_perAc_perYr_FPU.py
# Created on: 2015-02-06 11:39:07.00000
# Usage: Clip_DenGridVal_perAc_perYr_FPU
<Enter_FPU_Boundary__ex__CA_CA_001_FPU_albrs_1ex_shp_>
<Use_Input_Features_for_Clipping_Geometry>
<Enter_Developed_Areas_Totals_Areas_Affected_Acres_per_Year__ex__caca1dvayr_>
<Output_Developed_Areas_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1dvayrf_>
<Enter_Infrastructure_Totals_Affected_Acres_per_Year__ex__caca1nfayr_>
<Output_Infrastructure_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1nfayrf_>
<Enter_Recreation_Sites_Totals_Affected_Acres_per_Year__ex__caca1rcayr_>
<Output_Recreation_Sites_Value_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1rcayrf_>
<Enter_Riparian_Areas_Totals_Affected_Acres_per_Year__ex__caca1rpayr_>
<Output_Riparian_Areas_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1rpayrf_>
<Enter_Forest_Areas_Totals_Areas_Affected_Acres_per_Year__ex__caca1ftayr_>
<Output_Forest_Areas_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1ftayrf_>
<Enter_Threatened___Endangered_Totals_Affected_Acres_per_Year__ex__caca1teayr_>
<Output_Threatened___Endangered_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1teayrf_>
<Enter_Drinking_Water_Totals_Affected_Acres_per_Year__ex__caca1dwayr_>
<Output_Drinking_Water_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1dwayrf_> <Enter_Sage-
Grouse_Totals_Affected_Acres_per_Year__ex__caca1sgayr_> <Output_Sage-
Grouse_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1sgayrf_>
# Description: Clip Density Grid to FPU Boundary
# ---------------------------------------------------------------------------

# Import arcpy module
import arcpy

# Script arguments
Enter_FPU_Boundary__ex__CA_CA_001_FPU_albrs_1ex_shp_ = arcpy.GetParameterAsText(0)

Use_Input_Features_for_Clipping_Geometry = arcpy.GetParameterAsText(1)
if Use_Input_Features_for_Clipping_Geometry == '#' or not
Use_Input_Features_for_Clipping_Geometry:
    Use_Input_Features_for_Clipping_Geometry = "false" # provide a default value if unspecified

Enter_Developed_Areas_Totals_Areas_Affected_Acres_per_Year__ex__caca1dvayr_ =
arcpy.GetParameterAsText(2)

Output_Developed_Areas_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1dvayrf_ =
arcpy.GetParameterAsText(3)

Enter_Infrastructure_Totals_Affected_Acres_per_Year__ex__caca1nfayr_ =
arcpy.GetParameterAsText(4)

Output_Infrastructure_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1nfayrf_ =
arcpy.GetParameterAsText(5)
```

```
Enter_Recreation_Sites_Totals_Affected_Acres_per_Year__ex__caca1rcayr_ =
arcpy.GetParameterAsText(6)

Output_Recreation_Sites_Value_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1rcayrf_ =
arcpy.GetParameterAsText(7)

Enter_Riparian_Areas_Totals_Affected_Acres_per_Year__ex__caca1rpayr_ =
arcpy.GetParameterAsText(8)

Output_Riparian_Areas_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1rpayrf_ =
arcpy.GetParameterAsText(9)

Enter_Forest_Areas_Totals_Areas_Affected_Acres_per_Year__ex__caca1ftayr_ =
arcpy.GetParameterAsText(10)

Output_Forest_Areas_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1ftayrf_ =
arcpy.GetParameterAsText(11)

Enter_Threatened___Endangered_Totals_Affected_Acres_per_Year__ex__caca1teayr_ =
arcpy.GetParameterAsText(12)

Output_Threatened___Endangered_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1teayrf_ =
arcpy.GetParameterAsText(13)

Enter_Drinking_Water_Totals_Affected_Acres_per_Year__ex__caca1dwayr_ =
arcpy.GetParameterAsText(14)

Output_Drinking_Water_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1dwayrf_ =
arcpy.GetParameterAsText(15)

Enter_Sage-Grouse_Totals_Affected_Acres_per_Year__ex__caca1sgayr_ =
arcpy.GetParameterAsText(16)

Output_Sage-Grouse_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1sgayrf_ =
arcpy.GetParameterAsText(17)

# Local variables:

# Process: Clip (6)
arcpy.Clip_management(Enter_Forest_Areas_Totals_Areas_Affected_Acres_per_Year__ex__caca1ftayr_
, "-2259055.04144794 2536789.54939306 -1990405.04144794 2921269.54939306",
Output_Forest_Areas_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1ftayrf_,
Enter_FPU_Boundary__ex__CA_CA_001_FPU_albrs_1ex_shp_, "-3.402823e+038",
Use_Input_Features_for_Clipping_Geometry, "NO_MAINTAIN_EXTENT")

# Process: Clip (7)
arcpy.Clip_management(Enter_Threatened___Endangered_Totals_Affected_Acres_per_Year__ex__cac
a1teayr_, "-2259055.04144794 2536789.54939306 -1990405.04144794 2921269.54939306",
```

```
Output_Threatened___Endangered_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1teayrf_,
Enter_FPU_Boundary__ex__CA_CA_001_FPU_albrs_1ex_shp_, "-3.402823e+038",
Use_Input_Features_for_Clipping_Geometry, "NO_MAINTAIN_EXTENT")

# Process: Clip (8)
arcpy.Clip_management(Enter_Drinking_Water_Totals_Affected_Acres_per_Year__ex__caca1dwayr_,
"-2259055.04144794 2536789.54939306 -1990405.04144794 2921269.54939306",
Output_Drinking_Water_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1dwayrf_,
Enter_FPU_Boundary__ex__CA_CA_001_FPU_albrs_1ex_shp_, "-3.402823e+038",
Use_Input_Features_for_Clipping_Geometry, "NO_MAINTAIN_EXTENT")

# Process: Clip (9)
arcpy.Clip_management(Enter_Sage-Grouse_Totals_Affected_Acres_per_Year__ex__caca1sgayr_, "-
2259055.04144794 2536789.54939306 -1990405.04144794 2921269.54939306", Output_Sage-
Grouse_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1sgayrf_,
Enter_FPU_Boundary__ex__CA_CA_001_FPU_albrs_1ex_shp_, "-3.402823e+038",
Use_Input_Features_for_Clipping_Geometry, "NO_MAINTAIN_EXTENT")

# Process: Clip (2)
arcpy.Clip_management(Enter_Developed_Areas_Totals_Areas_Affected_Acres_per_Year__ex__caca1d
vayr_, "-2259055.04144794 2536789.54939306 -1990405.04144794 2921269.54939306",
Output_Developed_Areas_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1dvayrf_,
Enter_FPU_Boundary__ex__CA_CA_001_FPU_albrs_1ex_shp_, "-3.402823e+038",
Use_Input_Features_for_Clipping_Geometry, "NO_MAINTAIN_EXTENT")

# Process: Clip (3)
arcpy.Clip_management(Enter_Infrastructure_Totals_Affected_Acres_per_Year__ex__caca1nfayr_, "-
2259055.04144794 2536789.54939306 -1990405.04144794 2921269.54939306",
Output_Infrastructure_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1nfayrf_,
Enter_FPU_Boundary__ex__CA_CA_001_FPU_albrs_1ex_shp_, "-3.402823e+038",
Use_Input_Features_for_Clipping_Geometry, "NO_MAINTAIN_EXTENT")

# Process: Clip (4)
arcpy.Clip_management(Enter_Recreation_Sites_Totals_Affected_Acres_per_Year__ex__caca1rcayr_, "-
2259055.04144794 2536789.54939306 -1990405.04144794 2921269.54939306",
Output_Recreation_Sites_Value_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1rcayrf_,
Enter_FPU_Boundary__ex__CA_CA_001_FPU_albrs_1ex_shp_, "-3.402823e+038",
Use_Input_Features_for_Clipping_Geometry, "NO_MAINTAIN_EXTENT")

# Process: Clip (5)
arcpy.Clip_management(Enter_Riparian_Areas_Totals_Affected_Acres_per_Year__ex__caca1rpayr_, "-
2259055.04144794 2536789.54939306 -1990405.04144794 2921269.54939306",
Output_Riparian_Areas_Totals_Affected_Acres_per_Year_-_FPU__ex__caca1rpayrf_,
Enter_FPU_Boundary__ex__CA_CA_001_FPU_albrs_1ex_shp_, "-3.402823e+038",
Use_Input_Features_for_Clipping_Geometry, "NO_MAINTAIN_EXTENT")
```

Appendix L

Python Script for Step 10:

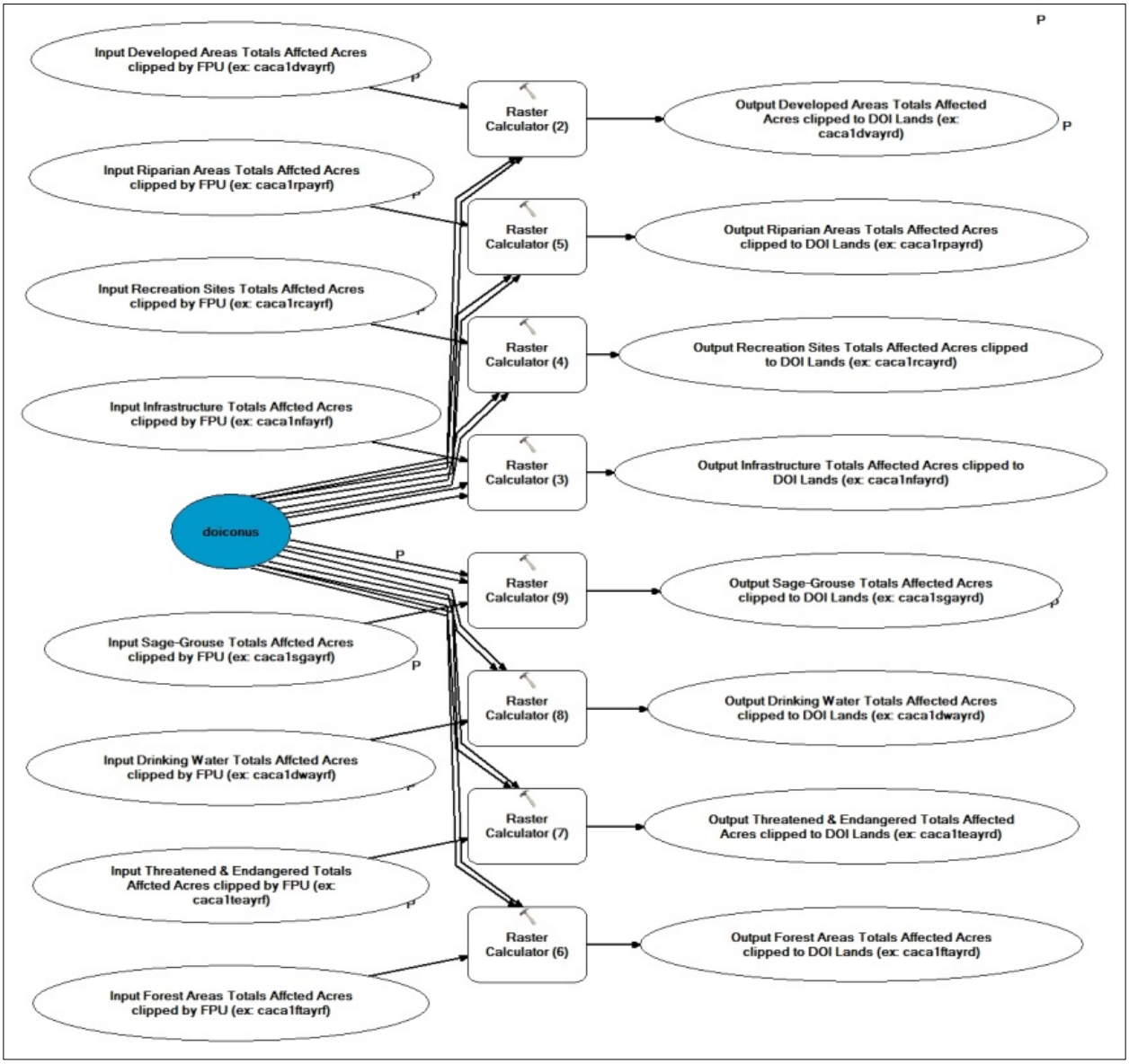


Figure A- 10: Step 10 process flow diagram - clip density grid to DOI wildfire-fighting Bureau boundaries

```
# ---------------------------------------------------------------------------

# Clip_DenGridVal_perAc_perYr_DOI.py
# Created on: 2015-02-06 11:39:40.00000
# Usage: Clip_DenGridVal_perAc_perYr_DOI
<Input_Developed_Areas_Totals_Affcted_Acres_clipped_by_FPU__ex__caca1dvayrf_>
<Output_Developed_Areas_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1dvayrd_>
<Input_Infrastructure_Totals_Affcted_Acres_clipped_by_FPU__ex__caca1nfayrf_>
<Output_Infrastructure_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1nfayrd_>
```

```
<Input_Recreation_Sites_Totals_Affcted_Acres_clipped_by_FPU__ex__caca1rcayrf_>
<Output_Recreation_Sites_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1rcayrd_>
<Input_Riparian_Areas_Totals_Affcted_Acres_clipped_by_FPU__ex__caca1rpayrf_>
<Output_Riparian_Areas_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1rpayrd_>
<Input_Forest_Areas_Totals_Affcted_Acres_clipped_by_FPU__ex__caca1ftayrf_>
<Output_Forest_Areas_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1ftayrd_>
<Input_Threatened___Endangered_Totals_Affcted_Acres_clipped_by_FPU__ex__caca1teayrf_>
<Output_Threatened___Endangered_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1teayrd
_> <Input_Drinking_Water_Totals_Affcted_Acres_clipped_by_FPU__ex__caca1dwayrf_>
<Output_Drinking_Water_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1dwayrd_>
<Input_Sage-Grouse_Totals_Affcted_Acres_clipped_by_FPU__ex__caca1sgayrf_> <Output_Sage-
Grouse_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1sgayrd_>
# Description: Clip Density Grid to DOI wildfire-fighting Bureau boundaries
# ---------------------------------------------------------------------------

# Import arcpy module
import arcpy

# Check out any necessary licenses
arcpy.CheckOutExtension("spatial")

# Script arguments
Input_Developed_Areas_Totals_Affcted_Acres_clipped_by_FPU__ex__caca1dvayrf_ =
arcpy.GetParameterAsText(0)

Output_Developed_Areas_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1dvayrd_ =
arcpy.GetParameterAsText(1)

Input_Infrastructure_Totals_Affcted_Acres_clipped_by_FPU__ex__caca1nfayrf_ =
arcpy.GetParameterAsText(2)

Output_Infrastructure_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1nfayrd_ =
arcpy.GetParameterAsText(3)

Input_Recreation_Sites_Totals_Affcted_Acres_clipped_by_FPU__ex__caca1rcayrf_ =
arcpy.GetParameterAsText(4)

Output_Recreation_Sites_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1rcayrd_ =
arcpy.GetParameterAsText(5)

Input_Riparian_Areas_Totals_Affcted_Acres_clipped_by_FPU__ex__caca1rpayrf_ =
arcpy.GetParameterAsText(6)

Output_Riparian_Areas_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1rpayrd_ =
arcpy.GetParameterAsText(7)

Input_Forest_Areas_Totals_Affcted_Acres_clipped_by_FPU__ex__caca1ftayrf_ =
arcpy.GetParameterAsText(8)
```

```
Output_Forest_Areas_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1ftayrd_ =
arcpy.GetParameterAsText(9)

Input_Threatened___Endangered_Totals_Affcted_Acres_clipped_by_FPU__ex__caca1teayrf_ =
arcpy.GetParameterAsText(10)

Output_Threatened___Endangered_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1teayrd_
= arcpy.GetParameterAsText(11)

Input_Drinking_Water_Totals_Affcted_Acres_clipped_by_FPU__ex__caca1dwayrf_ =
arcpy.GetParameterAsText(12)

Output_Drinking_Water_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1dwayrd_ =
arcpy.GetParameterAsText(13)

Input_Sage-Grouse_Totals_Affcted_Acres_clipped_by_FPU__ex__caca1sgayrf_ =
arcpy.GetParameterAsText(14)

Output_Sage-Grouse_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1sgayrd_ =
arcpy.GetParameterAsText(15)

# Local variables:
doiconus = "T:\\2_GISProjects\\Risk-Based_Wildland_Fire_Mgmt\\Data\\Grids\\doiconus"

# Process: Raster Calculator (6)
arcpy.gp.RasterCalculator_sa("(\"%Input Forest Areas Totals Affcted Acres clipped by FPU (ex:
caca1ftayrf)%\" + \"%doiconus%\") - \"%doiconus%\"",
Output_Forest_Areas_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1ftayrd_)

# Process: Raster Calculator (7)
arcpy.gp.RasterCalculator_sa("(\"%Input Threatened & Endangered Totals Affcted Acres clipped by FPU
(ex: caca1teayrf)%\" + \"%doiconus%\") - \"%doiconus%\"",
Output_Threatened___Endangered_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1teayrd_)

# Process: Raster Calculator (8)
arcpy.gp.RasterCalculator_sa("(\"%Input Drinking Water Totals Affcted Acres clipped by FPU (ex:
caca1dwayrf)%\" + \"%doiconus%\") - \"%doiconus%\"",
Output_Drinking_Water_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1dwayrd_)

# Process: Raster Calculator (9)
arcpy.gp.RasterCalculator_sa("(\"%Input Sage-Grouse Totals Affcted Acres clipped by FPU (ex:
caca1sgayrf)%\" + \"%doiconus%\") - \"%doiconus%\"", Output_Sage-
Grouse_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1sgayrd_)

# Process: Raster Calculator (3)
```

```
arcpy.gp.RasterCalculator_sa("(\"%Input Infrastructure Totals Affcted Acres clipped by FPU (ex:
caca1nfayrf)%\" + \"%doiconus%\") - \"%doiconus%\"",
Output_Infrastructure_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1nfayrd_)

# Process: Raster Calculator (4)
arcpy.gp.RasterCalculator_sa("(\"%Input Recreation Sites Totals Affcted Acres clipped by FPU (ex:
caca1rcayrf)%\" + \"%doiconus%\") - \"%doiconus%\"",
Output_Recreation_Sites_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1rcayrd_)

# Process: Raster Calculator (5)
arcpy.gp.RasterCalculator_sa("(\"%Input Riparian Areas Totals Affcted Acres clipped by FPU (ex:
caca1rpayrf)%\" + \"%doiconus%\") - \"%doiconus%\"",
Output_Riparian_Areas_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1rpayrd_)

# Process: Raster Calculator (2)
arcpy.gp.RasterCalculator_sa("(\"%Input Developed Areas Totals Affcted Acres clipped by FPU (ex:
caca1dvayrf)%\"  + \"%doiconus%\") - \"%doiconus%\"",
Output_Developed_Areas_Totals_Affected_Acres_clipped_to_DOI_Lands__ex__caca1dvayrd_)
```

Appendix M

Python Script for Step 11:

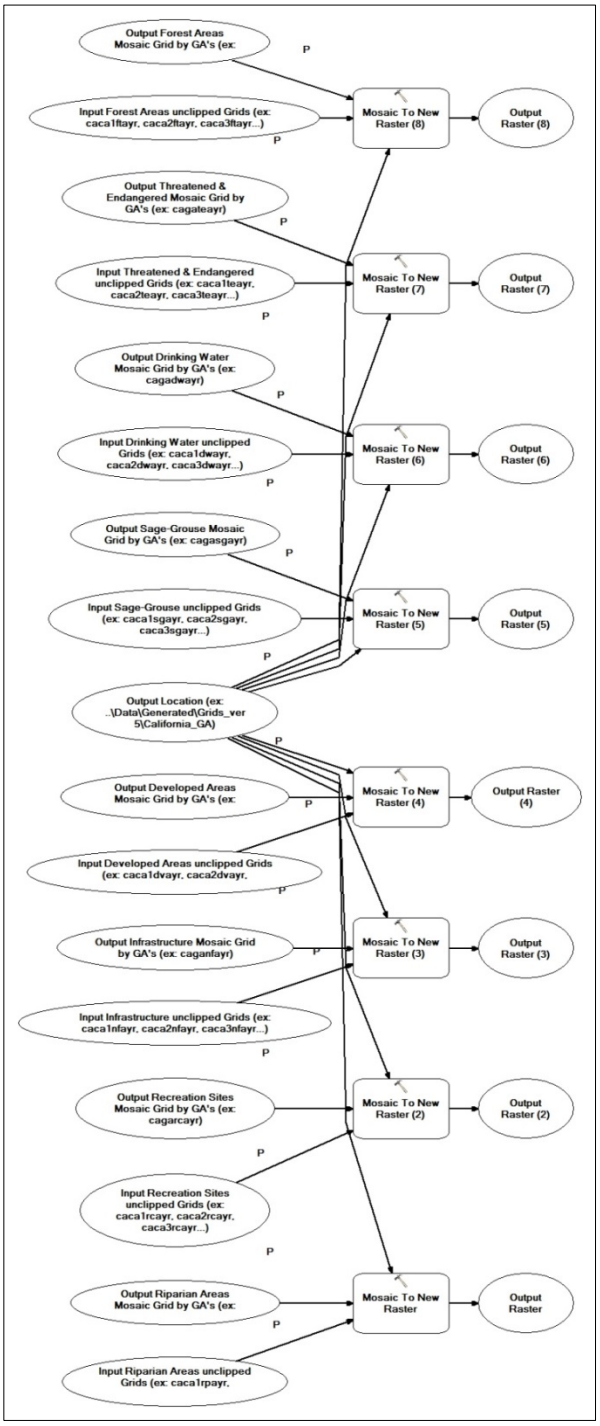


Figure A- 11: Step 11 process flow diagram - mosaic FPU clipped density grids to form a new raster grid feature extending to GA borders

```
# ---------------------------------------------------------------------------
# Mosaic_unClip_DenGrid_by_GA.py
# Created on: 2015-02-06 11:40:48.00000
# Usage: Mosaic_unClip_DenGrid_by_GA
<Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_>
<Input_Developed_Areas_unclipped_Grids__ex__caca1dvayr__caca2dvayr__caca3dvayr____>
<Output_Developed_Areas_Mosaic_Grid_by_GA_s__ex__cagadvayr_>
<Input_Infrastructure_unclipped_Grids__ex__caca1nfayr__caca2nfayr__caca3nfayr____>
<Output_Infrastructure_Mosaic_Grid_by_GA_s__ex__caganfayr_>
<Input_Recreation_Sites_unclipped_Grids__ex__caca1rcayr__caca2rcayr__caca3rcayr____>
<Output_Recreation_Sites_Mosaic_Grid_by_GA_s__ex__cagarcayr_>
<Input_Riparian_Areas_unclipped_Grids__ex__caca1rpayr__caca2rpayr__caca3rpayr____>
<Output_Riparian_Areas_Mosaic_Grid_by_GA_s__ex__cagarpayr_>
<Input_Forest_Areas_unclipped_Grids__ex__caca1ftayr__caca2ftayr__caca3ftayr____>
<Output_Forest_Areas_Mosaic_Grid_by_GA_s__ex__cagaftayr_>
<Input_Threatened___Endangered_unclipped_Grids__ex__caca1teayr__caca2teayr__caca3teayr____>
<Output_Threatened___Endangered_Mosaic_Grid_by_GA_s__ex__cagateayr_>
<Input_Drinking_Water_unclipped_Grids__ex__caca1dwayr__caca2dwayr__caca3dwayr____>
<Output_Drinking_Water_Mosaic_Grid_by_GA_s__ex__cagadwayr_> <Input_Sage-
Grouse_unclipped_Grids__ex__caca1sgayr__caca2sgayr__caca3sgayr____> <Output_Sage-
Grouse_Mosaic_Grid_by_GA_s__ex__cagasgayr_>
# Description: Mosaic FPU clipped Density Grids to form new raster grid feature extending to
#                    Geographic Area (GA) borders
#---------------------------------------------------------------------------

# Import arcpy module
import arcpy

# Script arguments
Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_ =
arcpy.GetParameterAsText(0)

Input_Developed_Areas_unclipped_Grids__ex__caca1dvayr__caca2dvayr__caca3dvayr____ =
arcpy.GetParameterAsText(1)

Output_Developed_Areas_Mosaic_Grid_by_GA_s__ex__cagadvayr_ = arcpy.GetParameterAsText(2)

Input_Infrastructure_unclipped_Grids__ex__caca1nfayr__caca2nfayr__caca3nfayr____ =
arcpy.GetParameterAsText(3)

Output_Infrastructure_Mosaic_Grid_by_GA_s__ex__caganfayr_ = arcpy.GetParameterAsText(4)

Input_Recreation_Sites_unclipped_Grids__ex__caca1rcayr__caca2rcayr__caca3rcayr____ =
arcpy.GetParameterAsText(5)

Output_Recreation_Sites_Mosaic_Grid_by_GA_s__ex__cagarcayr_ = arcpy.GetParameterAsText(6)
```

```
Input_Riparian_Areas_unclipped_Grids__ex__caca1rpayr__caca2rpayr__caca3rpayr____ =
arcpy.GetParameterAsText(7)

Output_Riparian_Areas_Mosaic_Grid_by_GA_s__ex__cagarpayr_ = arcpy.GetParameterAsText(8)

Input_Forest_Areas_unclipped_Grids__ex__caca1ftayr__caca2ftayr__caca3ftayr____ =
arcpy.GetParameterAsText(9)

Output_Forest_Areas_Mosaic_Grid_by_GA_s__ex__cagaftayr_ = arcpy.GetParameterAsText(10)

Input_Threatened___Endangered_unclipped_Grids__ex__caca1teayr__caca2teayr__caca3teayr____ =
arcpy.GetParameterAsText(11)

Output_Threatened___Endangered_Mosaic_Grid_by_GA_s__ex__cagateayr_ =
arcpy.GetParameterAsText(12)

Input_Drinking_Water_unclipped_Grids__ex__caca1dwayr__caca2dwayr__caca3dwayr____ =
arcpy.GetParameterAsText(13)

Output_Drinking_Water_Mosaic_Grid_by_GA_s__ex__cagadwayr_ = arcpy.GetParameterAsText(14)

Input_Sage-Grouse_unclipped_Grids__ex__caca1sgayr__caca2sgayr__caca3sgayr____ =
arcpy.GetParameterAsText(15)

Output_Sage-Grouse_Mosaic_Grid_by_GA_s__ex__cagasgayr_ = arcpy.GetParameterAsText(16)

# Local variables:
Output_Raster =
Input_Riparian_Areas_unclipped_Grids__ex__caca1rpayr__caca2rpayr__caca3rpayr____
Output_Raster__2_ = Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_
Output_Raster__3_ = Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_
Output_Raster__4_ = Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_
Output_Raster__5_ = Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_
Output_Raster__6_ = Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_
Output_Raster__7_ = Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_
Output_Raster__8_ = Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_

# Process: Mosaic To New Raster
arcpy.MosaicToNewRaster_management(Input_Riparian_Areas_unclipped_Grids__ex__caca1rpayr__ca
ca2rpayr__caca3rpayr____, Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_,
Output_Riparian_Areas_Mosaic_Grid_by_GA_s__ex__cagarpayr_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")
```

```
# Process: Mosaic To New Raster (2)
arcpy.MosaicToNewRaster_management(Input_Recreation_Sites_unclipped_Grids__ex__caca1rcayr__c
aca2rcayr__caca3rcayr____, Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_,
Output_Recreation_Sites_Mosaic_Grid_by_GA_s__ex__cagarcayr_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")

# Process: Mosaic To New Raster (3)
arcpy.MosaicToNewRaster_management(Input_Infrastructure_unclipped_Grids__ex__caca1nfayr__cac
a2nfayr__caca3nfayr____, Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_,
Output_Infrastructure_Mosaic_Grid_by_GA_s__ex__caganfayr_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")

# Process: Mosaic To New Raster (4)
arcpy.MosaicToNewRaster_management(Input_Developed_Areas_unclipped_Grids__ex__caca1dvayr__
caca2dvayr__caca3dvayr____, Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_,
Output_Developed_Areas_Mosaic_Grid_by_GA_s__ex__cagadvayr_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")

# Process: Mosaic To New Raster (5)
arcpy.MosaicToNewRaster_management(Input_Sage-
Grouse_unclipped_Grids__ex__caca1sgayr__caca2sgayr__caca3sgayr____,
Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_, Output_Sage-
Grouse_Mosaic_Grid_by_GA_s__ex__cagasgayr_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")

# Process: Mosaic To New Raster (6)
arcpy.MosaicToNewRaster_management(Input_Drinking_Water_unclipped_Grids__ex__caca1dwayr__c
aca2dwayr__caca3dwayr____, Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_,
```

```
Output_Drinking_Water_Mosaic_Grid_by_GA_s__ex__cagadwayr_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")

# Process: Mosaic To New Raster (7)
arcpy.MosaicToNewRaster_management(Input_Threatened___Endangered_unclipped_Grids__ex__cac
a1teayr__caca2teayr__caca3teayr____,
Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_,
Output_Threatened___Endangered_Mosaic_Grid_by_GA_s__ex__cagateayr_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")

# Process: Mosaic To New Raster (8)
arcpy.MosaicToNewRaster_management(Input_Forest_Areas_unclipped_Grids__ex__caca1ftayr__caca
2ftayr__caca3ftayr____, Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_,
Output_Forest_Areas_Mosaic_Grid_by_GA_s__ex__cagaftayr_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")
```

Appendix N

Python Script for Step 12:

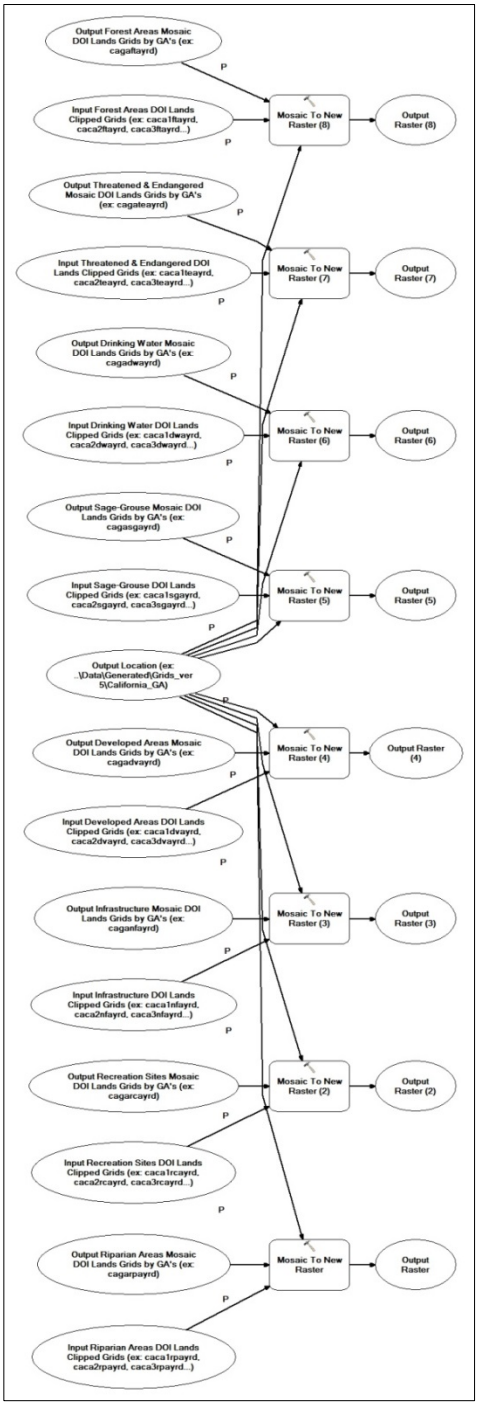


Figure A- 12: Step 12 process flow diagram - mosaic DOI wildfire-fighting Bureau boundary clipped density grids to form new raster grid feature extending to GA borders

```
# ---------------------------------------------------------------------------
# Mosaic_ClipDOI_DenGrid_by_GA.py
# Created on: 2015-02-06 11:41:19.00000
# Usage: Mosaic_ClipDOI_DenGrid_by_GA
<Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_>
<Input_Developed_Areas_DOI_Lands_Clipped_Grids__ex__caca1dvayrd__caca2dvayrd__caca3dvayrd_
___> <Output_Developed_Areas_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagadvayrd_>
<Input_Infrastructure_DOI_Lands_Clipped_Grids__ex__caca1nfayrd__caca2nfayrd__caca3nfayrd____>
<Output_Infrastructure_Mosaic_DOI_Lands_Grids_by_GA_s__ex__caganfayrd_>
<Input_Recreation_Sites_DOI_Lands_Clipped_Grids__ex__caca1rcayrd__caca2rcayrd__caca3rcayrd___
_> <Output_Recreation_Sites_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagarcayrd_>
<Input_Riparian_Areas_DOI_Lands_Clipped_Grids__ex__caca1rpayrd__caca2rpayrd__caca3rpayrd____
> <Output_Riparian_Areas_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagarpayrd_>
<Input_Forest_Areas_DOI_Lands_Clipped_Grids__ex__caca1ftayrd__caca2ftayrd__caca3ftayrd____>
<Output_Forest_Areas_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagaftayrd_>
<Input_Threatened___Endangered_DOI_Lands_Clipped_Grids__ex__caca1teayrd__caca2teayrd__caca3
teayrd____>
<Output_Threatened___Endangered_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagateayrd_>
<Input_Drinking_Water_DOI_Lands_Clipped_Grids__ex__caca1dwayrd__caca2dwayrd__caca3dwayrd_
___> <Output_Drinking_Water_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagadwayrd_> <Input_Sage-
Grouse_DOI_Lands_Clipped_Grids__ex__caca1sgayrd__caca2sgayrd__caca3sgayrd____>
<Output_Sage-Grouse_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagasgayrd_>
# Description: Mosaic DOI wildfire-fighting Bureau boundary clipped Density Grids to form new raster
#                    grid feature extending to Geographic Area (GA) borders
#---------------------------------------------------------------------------

# Import arcpy module
import arcpy

# Script arguments
Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_ =
arcpy.GetParameterAsText(0)

Input_Developed_Areas_DOI_Lands_Clipped_Grids__ex__caca1dvayrd__caca2dvayrd__caca3dvayrd__
__ = arcpy.GetParameterAsText(1)

Output_Developed_Areas_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagadvayrd_ =
arcpy.GetParameterAsText(2)

Input_Infrastructure_DOI_Lands_Clipped_Grids__ex__caca1nfayrd__caca2nfayrd__caca3nfayrd____ =
arcpy.GetParameterAsText(3)

Output_Infrastructure_Mosaic_DOI_Lands_Grids_by_GA_s__ex__caganfayrd_ =
arcpy.GetParameterAsText(4)

Input_Recreation_Sites_DOI_Lands_Clipped_Grids__ex__caca1rcayrd__caca2rcayrd__caca3rcayrd____
= arcpy.GetParameterAsText(5)
```

```
Output_Recreation_Sites_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagarcayrd_ =
arcpy.GetParameterAsText(6)

Input_Riparian_Areas_DOI_Lands_Clipped_Grids__ex__caca1rpayrd__caca2rpayrd__caca3rpayrd____ =
arcpy.GetParameterAsText(7)

Output_Riparian_Areas_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagarpayrd_ =
arcpy.GetParameterAsText(8)

Input_Forest_Areas_DOI_Lands_Clipped_Grids__ex__caca1ftayrd__caca2ftayrd__caca3ftayrd____ =
arcpy.GetParameterAsText(9)

Output_Forest_Areas_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagaftayrd_ =
arcpy.GetParameterAsText(10)

Input_Threatened___Endangered_DOI_Lands_Clipped_Grids__ex__caca1teayrd__caca2teayrd__caca3t
eayrd____ = arcpy.GetParameterAsText(11)

Output_Threatened___Endangered_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagateayrd_ =
arcpy.GetParameterAsText(12)

Input_Drinking_Water_DOI_Lands_Clipped_Grids__ex__caca1dwayrd__caca2dwayrd__caca3dwayrd__
__ = arcpy.GetParameterAsText(13)

Output_Drinking_Water_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagadwayrd_ =
arcpy.GetParameterAsText(14)

Input_Sage-Grouse_DOI_Lands_Clipped_Grids__ex__caca1sgayrd__caca2sgayrd__caca3sgayrd____ =
arcpy.GetParameterAsText(15)

Output_Sage-Grouse_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagasgayrd_ =
arcpy.GetParameterAsText(16)

# Local variables:
Output_Raster =
Input_Riparian_Areas_DOI_Lands_Clipped_Grids__ex__caca1rpayrd__caca2rpayrd__caca3rpayrd____
Output_Raster__2_ = Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_
Output_Raster__3_ = Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_
Output_Raster__4_ = Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_
Output_Raster__5_ = Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_
Output_Raster__6_ = Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_
Output_Raster__7_ = Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_
Output_Raster__8_ = Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_

# Process: Mosaic To New Raster
arcpy.MosaicToNewRaster_management(Input_Riparian_Areas_DOI_Lands_Clipped_Grids__ex__caca1r
payrd__caca2rpayrd__caca3rpayrd____,
Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_,
```

```
Output_Riparian_Areas_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagarpayrd_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")

# Process: Mosaic To New Raster (2)
arcpy.MosaicToNewRaster_management(Input_Recreation_Sites_DOI_Lands_Clipped_Grids__ex__caca
1rcayrd__caca2rcayrd__caca3rcayrd____,
Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_,
Output_Recreation_Sites_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagarcayrd_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")

# Process: Mosaic To New Raster (3)
arcpy.MosaicToNewRaster_management(Input_Infrastructure_DOI_Lands_Clipped_Grids__ex__caca1nf
ayrd__caca2nfayrd__caca3nfayrd____,
Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_,
Output_Infrastructure_Mosaic_DOI_Lands_Grids_by_GA_s__ex__caganfayrd_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")

# Process: Mosaic To New Raster (4)
arcpy.MosaicToNewRaster_management(Input_Developed_Areas_DOI_Lands_Clipped_Grids__ex__cac
a1dvayrd__caca2dvayrd__caca3dvayrd____,
Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_,
Output_Developed_Areas_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagadvayrd_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")

# Process: Mosaic To New Raster (5)
arcpy.MosaicToNewRaster_management(Input_Sage-
Grouse_DOI_Lands_Clipped_Grids__ex__caca1sgayrd__caca2sgayrd__caca3sgayrd____,
Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_, Output_Sage-
```

```
Grouse_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagasgayrd_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")

# Process: Mosaic To New Raster (6)
arcpy.MosaicToNewRaster_management(Input_Drinking_Water_DOI_Lands_Clipped_Grids__ex__caca1
dwayrd__caca2dwayrd__caca3dwayrd____,
Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_,
Output_Drinking_Water_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagadwayrd_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")

# Process: Mosaic To New Raster (7)
arcpy.MosaicToNewRaster_management(Input_Threatened___Endangered_DOI_Lands_Clipped_Grids_
_ex__caca1teayrd__caca2teayrd__caca3teayrd____,
Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_,
Output_Threatened___Endangered_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagateayrd_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")

# Process: Mosaic To New Raster (8)
arcpy.MosaicToNewRaster_management(Input_Forest_Areas_DOI_Lands_Clipped_Grids__ex__caca1fta
yrd__caca2ftayrd__caca3ftayrd____,
Output_Location__ex_____Data_Generated_Grids_ver5_California_GA_,
Output_Forest_Areas_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagaftayrd_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")
```

Appendix O

Python Script for Step 13:

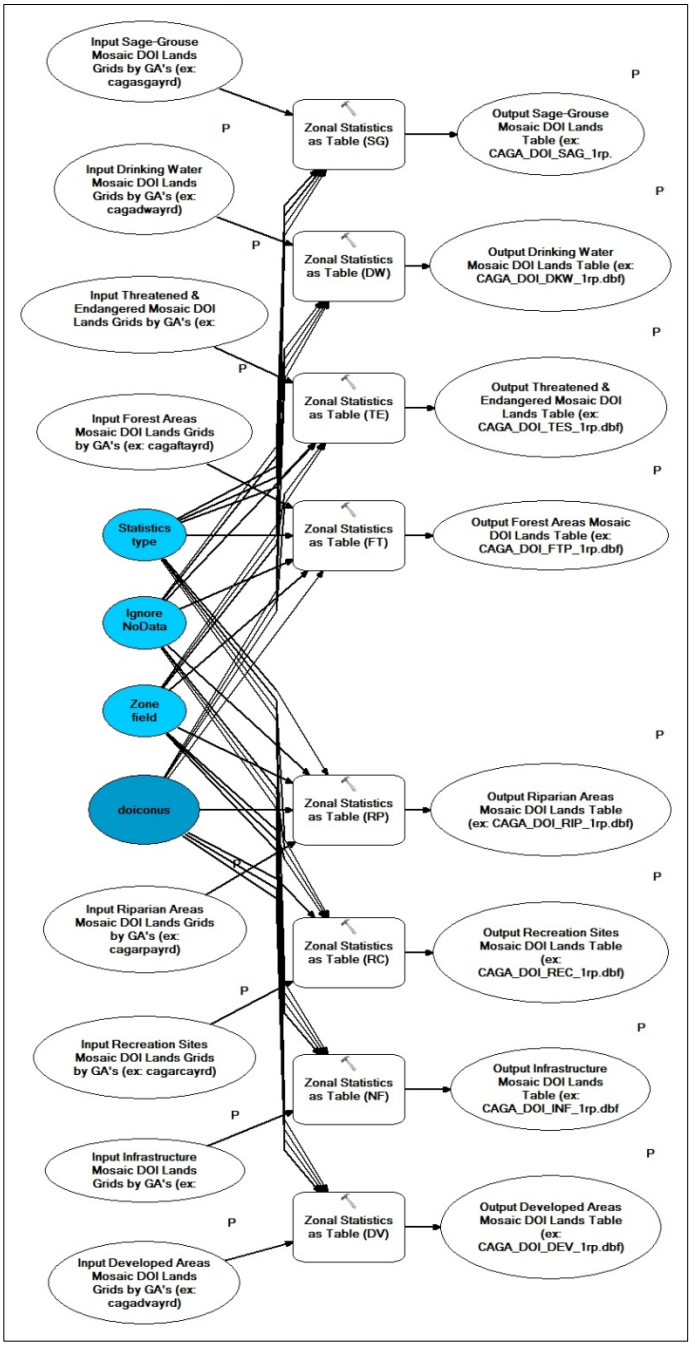


Figure A- 13: Step 13 process flow diagram - calculates zonal statistics for each NLPR raster and writes vales to a database table

```
# ----------------------------------------------------------------------------

# ZonStats_to_Table.py
```

```
# Created on: 2015-02-06 11:41:55.00000
# Usage: ZonStats_to_Table
<Input_Developed_Areas_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagadvayrd_>
<Output_Developed_Areas_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_DEV_1rp_dbf_>
<Input_Infrastructure_Mosaic_DOI_Lands_Grids_by_GA_s__ex__caganfayrd_>
<Output_Infrastructure_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_INF_1rp_dbf_>
<Input_Recreation_Sites_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagarcayrd_>
<Output_Recreation_Sites_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_REC_1rp_dbf_>
<Input_Riparian_Areas_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagarpayrd_>
<Output_Riparian_Areas_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_RIP_1rp_dbf_>
<Input_Forest_Areas_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagaftayrd_>
<Output_Forest_Areas_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_FTP_1rp_dbf_>
<Input_Threatened___Endangered_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagateayrd_>
<Output_Threatened___Endangered_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_TES_1rp_dbf_>
<Input_Drinking_Water_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagadwayrd_>
<Output_Drinking_Water_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_DKW_1rp_dbf_> <Input_Sage-
Grouse_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagasgayrd_> <Output_Sage-
Grouse_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_SAG_1rp_dbf_>
# Description: Compares the values of each NLPR raster grid within the zones of the DOI grid and writes
#              summarized values to a database table
# ---------------------------------------------------------------------------

# Import arcpy module
import arcpy

# Check out any necessary licenses
arcpy.CheckOutExtension("spatial")

# Script arguments
Input_Developed_Areas_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagadvayrd_ =
arcpy.GetParameterAsText(0)

Output_Developed_Areas_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_DEV_1rp_dbf_ =
arcpy.GetParameterAsText(1)
if Output_Developed_Areas_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_DEV_1rp_dbf_ == '#' or not
Output_Developed_Areas_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_DEV_1rp_dbf_:
    Output_Developed_Areas_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_DEV_1rp_dbf_ =
"T:\\2_GISProjects\\Risk-Based_Wildland_Fire_Mgmt\\Data\\Grids\\ZonalSt_doiconu1" # provide a
default value if unspecified

Input_Infrastructure_Mosaic_DOI_Lands_Grids_by_GA_s__ex__caganfayrd_ =
arcpy.GetParameterAsText(2)

Output_Infrastructure_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_INF_1rp_dbf_ =
arcpy.GetParameterAsText(3)
if Output_Infrastructure_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_INF_1rp_dbf_ == '#' or not
Output_Infrastructure_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_INF_1rp_dbf_:
```

```
    Output_Infrastructure_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_INF_1rp_dbf_ =
"T:\\2_GISProjects\\Risk-Based_Wildland_Fire_Mgmt\\Data\\Grids\\ZonalSt_doiconu2" # provide a
default value if unspecified

Input_Recreation_Sites_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagarcayrd_ =
arcpy.GetParameterAsText(4)

Output_Recreation_Sites_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_REC_1rp_dbf_ =
arcpy.GetParameterAsText(5)
if Output_Recreation_Sites_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_REC_1rp_dbf_ == '#' or not
Output_Recreation_Sites_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_REC_1rp_dbf_:
    Output_Recreation_Sites_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_REC_1rp_dbf_ =
"T:\\2_GISProjects\\Risk-Based_Wildland_Fire_Mgmt\\Data\\Grids\\ZonalSt_doiconu3" # provide a
default value if unspecified

Input_Riparian_Areas_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagarpayrd_ =
arcpy.GetParameterAsText(6)

Output_Riparian_Areas_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_RIP_1rp_dbf_ =
arcpy.GetParameterAsText(7)
if Output_Riparian_Areas_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_RIP_1rp_dbf_ == '#' or not
Output_Riparian_Areas_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_RIP_1rp_dbf_:
    Output_Riparian_Areas_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_RIP_1rp_dbf_ =
"T:\\2_GISProjects\\Risk-Based_Wildland_Fire_Mgmt\\Data\\Grids\\ZonalSt_doiconu4" # provide a
default value if unspecified

Input_Forest_Areas_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagaftayrd_ =
arcpy.GetParameterAsText(8)

Output_Forest_Areas_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_FTP_1rp_dbf_ =
arcpy.GetParameterAsText(9)
if Output_Forest_Areas_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_FTP_1rp_dbf_ == '#' or not
Output_Forest_Areas_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_FTP_1rp_dbf_:
    Output_Forest_Areas_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_FTP_1rp_dbf_ =
"T:\\2_GISProjects\\Risk-Based_Wildland_Fire_Mgmt\\Data\\Grids\\ZonalSt_doiconu5" # provide a
default value if unspecified

Input_Threatened___Endangered_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagateayrd_ =
arcpy.GetParameterAsText(10)

Output_Threatened___Endangered_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_TES_1rp_dbf_ =
arcpy.GetParameterAsText(11)
if Output_Threatened___Endangered_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_TES_1rp_dbf_ == '#'
or not Output_Threatened___Endangered_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_TES_1rp_dbf_:
    Output_Threatened___Endangered_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_TES_1rp_dbf_ =
"T:\\2_GISProjects\\Risk-Based_Wildland_Fire_Mgmt\\Data\\Grids\\ZonalSt_doiconu6" # provide a
default value if unspecified
```

```
Input_Drinking_Water_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagadwayrd_ =
arcpy.GetParameterAsText(12)

Output_Drinking_Water_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_DKW_1rp_dbf_ =
arcpy.GetParameterAsText(13)
if Output_Drinking_Water_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_DKW_1rp_dbf_ == '#' or not
Output_Drinking_Water_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_DKW_1rp_dbf_:
    Output_Drinking_Water_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_DKW_1rp_dbf_ =
"T:\\2_GISProjects\\Risk-Based_Wildland_Fire_Mgmt\\Data\\Grids\\ZonalSt_doiconu7" # provide a
default value if unspecified

Input_Sage-Grouse_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagasgayrd_ =
arcpy.GetParameterAsText(14)

Output_Sage-Grouse_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_SAG_1rp_dbf_ =
arcpy.GetParameterAsText(15)
if Output_Sage-Grouse_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_SAG_1rp_dbf_ == '#' or not
Output_Sage-Grouse_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_SAG_1rp_dbf_:
    Output_Sage-Grouse_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_SAG_1rp_dbf_ =
"T:\\2_GISProjects\\Risk-Based_Wildland_Fire_Mgmt\\Data\\Grids\\ZonalSt_doiconu8" # provide a
default value if unspecified

# Local variables:
doiconus = "T:\\2_GISProjects\\Risk-Based_Wildland_Fire_Mgmt\\Data\\Grids\\doiconus"
Zone_field = "BUREAU"
Ignore_NoData_in_calculations = "true"
Statistics_type = "ALL"

# Process: Zonal Statistics as Table (DV)
arcpy.gp.ZonalStatisticsAsTable_sa(doiconus, Zone_field,
Input_Developed_Areas_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagadvayrd_,
Output_Developed_Areas_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_DEV_1rp_dbf_,
Ignore_NoData_in_calculations, Statistics_type)

# Process: Zonal Statistics as Table (NF)
arcpy.gp.ZonalStatisticsAsTable_sa(doiconus, Zone_field,
Input_Infrastructure_Mosaic_DOI_Lands_Grids_by_GA_s__ex__caganfayrd_,
Output_Infrastructure_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_INF_1rp_dbf_,
Ignore_NoData_in_calculations, Statistics_type)

# Process: Zonal Statistics as Table (RC)
arcpy.gp.ZonalStatisticsAsTable_sa(doiconus, Zone_field,
Input_Recreation_Sites_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagarcayrd_,
Output_Recreation_Sites_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_REC_1rp_dbf_,
Ignore_NoData_in_calculations, Statistics_type)

# Process: Zonal Statistics as Table (RP)
```

```
arcpy.gp.ZonalStatisticsAsTable_sa(doiconus, Zone_field,
Input_Riparian_Areas_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagarpayrd_,
Output_Riparian_Areas_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_RIP_1rp_dbf_,
Ignore_NoData_in_calculations, Statistics_type)

# Process: Zonal Statistics as Table (FT)
arcpy.gp.ZonalStatisticsAsTable_sa(doiconus, Zone_field,
Input_Forest_Areas_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagaftayrd_,
Output_Forest_Areas_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_FTP_1rp_dbf_,
Ignore_NoData_in_calculations, Statistics_type)

# Process: Zonal Statistics as Table (TE)
arcpy.gp.ZonalStatisticsAsTable_sa(doiconus, Zone_field,
Input_Threatened___Endangered_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagateayrd_,
Output_Threatened___Endangered_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_TES_1rp_dbf_,
Ignore_NoData_in_calculations, Statistics_type)

# Process: Zonal Statistics as Table (DW)
arcpy.gp.ZonalStatisticsAsTable_sa(doiconus, Zone_field,
Input_Drinking_Water_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagadwayrd_,
Output_Drinking_Water_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_DKW_1rp_dbf_,
Ignore_NoData_in_calculations, Statistics_type)

# Process: Zonal Statistics as Table (SG)
arcpy.gp.ZonalStatisticsAsTable_sa(doiconus, Zone_field, Input_Sage-
Grouse_Mosaic_DOI_Lands_Grids_by_GA_s__ex__cagasgayrd_, Output_Sage-
Grouse_Mosaic_DOI_Lands_Table__ex__CAGA_DOI_SAG_1rp_dbf_, Ignore_NoData_in_calculations,
Statistics_type)
```

Appendix P

Python Script for Steps 14-16:

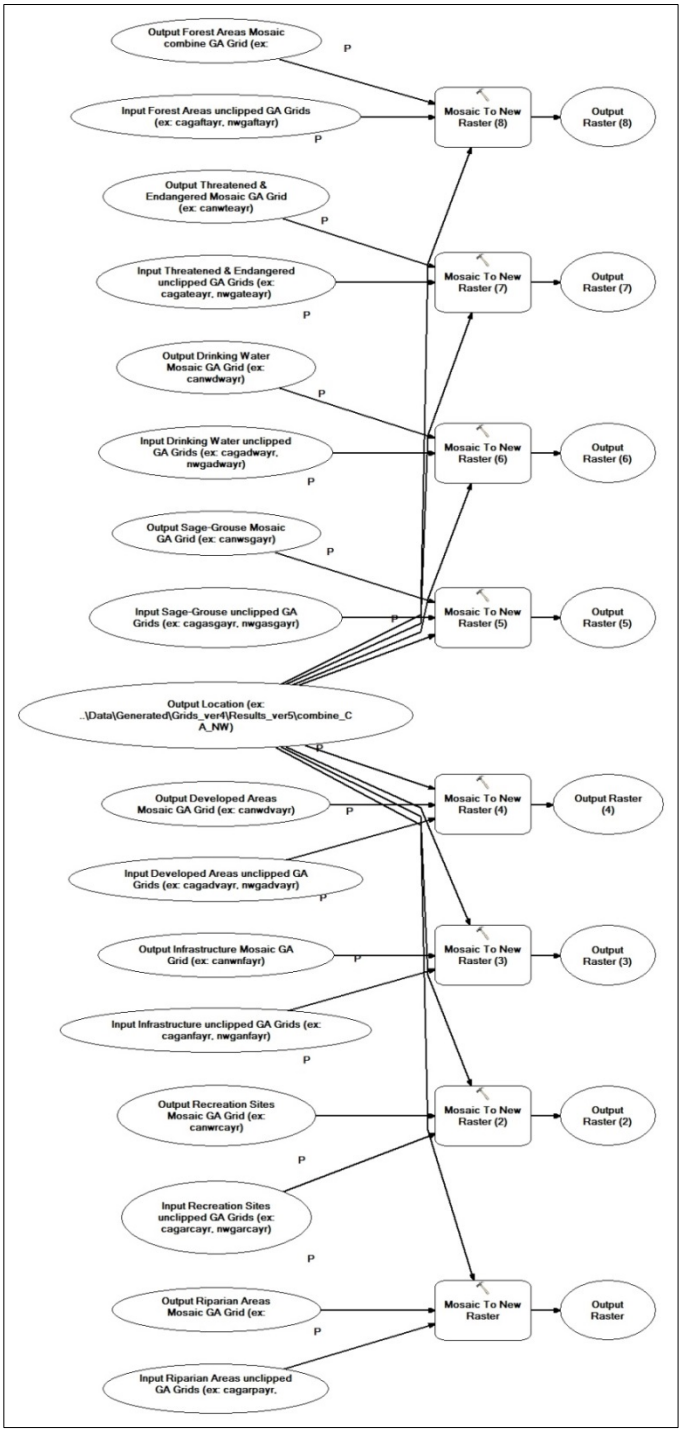


Figure A- 14: Step 14 through 16 process flow diagram - merge GA extent grid features to CONUS level extent

```
# ----------------------------------------------------------------------------
# Mosaic_GA_DenGrid_1stLev.py
```

```
# Created on: 2015-02-06 11:42:33.00000
# Usage: Mosaic_GA_DenGrid_1stLev
<Output_Location__ex_____Data_Generated_Grids_ver4_Results_ver5_combine_CA_NW_>
<Input_Developed_Areas_unclipped_GA_Grids__ex__cagadvayr__nwgadvayr_>
<Output_Developed_Areas_Mosaic_GA_Grid__ex__canwdvayr_>
<Input_Infrastructure_unclipped_GA_Grids__ex__caganfayr__nwganfayr_>
<Output_Infrastructure_Mosaic_GA_Grid__ex__canwnfayr_>
<Input_Recreation_Sites_unclipped_GA_Grids__ex__cagarcayr__nwgarcayr_>
<Output_Recreation_Sites_Mosaic_GA_Grid__ex__canwrcayr_>
<Input_Riparian_Areas_unclipped_GA_Grids__ex__cagarpayr__nwgarpayr_>
<Output_Riparian_Areas_Mosaic_GA_Grid__ex__canwrpayr_>
<Input_Forest_Areas_unclipped_GA_Grids__ex__cagaftayr__nwgaftayr_>
<Output_Forest_Areas_Mosaic_combine_GA_Grid__ex__canwftayr_>
<Input_Threatened___Endangered_unclipped_GA_Grids__ex__cagateayr__nwgateayr_>
<Output_Threatened___Endangered_Mosaic_GA_Grid__ex__canwteayr_>
<Input_Drinking_Water_unclipped_GA_Grids__ex__cagadwayr__nwgadwayr_>
<Output_Drinking_Water_Mosaic_GA_Grid__ex__canwdwayr_> <Input_Sage-
Grouse_unclipped_GA_Grids__ex__cagasgayr__nwgasgayr_> <Output_Sage-
Grouse_Mosaic_GA_Grid__ex__canwsgayr_>
# Description: Merge grid features at the GA coverage to CONUS level extent
# ---------------------------------------------------------------------------

# Import arcpy module
import arcpy

# Script arguments
Output_Location__ex_____Data_Generated_Grids_ver4_Results_ver5_combine_CA_NW_ =
arcpy.GetParameterAsText(0)

Input_Developed_Areas_unclipped_GA_Grids__ex__cagadvayr__nwgadvayr_ =
arcpy.GetParameterAsText(1)

Output_Developed_Areas_Mosaic_GA_Grid__ex__canwdvayr_ = arcpy.GetParameterAsText(2)

Input_Infrastructure_unclipped_GA_Grids__ex__caganfayr__nwganfayr_ =
arcpy.GetParameterAsText(3)

Output_Infrastructure_Mosaic_GA_Grid__ex__canwnfayr_ = arcpy.GetParameterAsText(4)

Input_Recreation_Sites_unclipped_GA_Grids__ex__cagarcayr__nwgarcayr_ =
arcpy.GetParameterAsText(5)

Output_Recreation_Sites_Mosaic_GA_Grid__ex__canwrcayr_ = arcpy.GetParameterAsText(6)

Input_Riparian_Areas_unclipped_GA_Grids__ex__cagarpayr__nwgarpayr_ =
arcpy.GetParameterAsText(7)

Output_Riparian_Areas_Mosaic_GA_Grid__ex__canwrpayr_ = arcpy.GetParameterAsText(8)
```

```
Input_Forest_Areas_unclipped_GA_Grids__ex__cagaftayr__nwgaftayr_ = arcpy.GetParameterAsText(9)

Output_Forest_Areas_Mosaic_combine_GA_Grid__ex__canwftayr_ = arcpy.GetParameterAsText(10)

Input_Threatened___Endangered_unclipped_GA_Grids__ex__cagateayr__nwgateayr_ =
arcpy.GetParameterAsText(11)

Output_Threatened___Endangered_Mosaic_GA_Grid__ex__canwteayr_ =
arcpy.GetParameterAsText(12)

Input_Drinking_Water_unclipped_GA_Grids__ex__cagadwayr__nwgadwayr_ =
arcpy.GetParameterAsText(13)

Output_Drinking_Water_Mosaic_GA_Grid__ex__canwdwayr_ = arcpy.GetParameterAsText(14)

Input_Sage-Grouse_unclipped_GA_Grids__ex__cagasgayr__nwgasgayr_ =
arcpy.GetParameterAsText(15)

Output_Sage-Grouse_Mosaic_GA_Grid__ex__canwsgayr_ = arcpy.GetParameterAsText(16)

# Local variables:
Output_Raster = Input_Riparian_Areas_unclipped_GA_Grids__ex__cagarpayr__nwgarpayr_
Output_Raster__2_ =
Output_Location__ex_____Data_Generated_Grids_ver4_Results_ver5_combine_CA_NW_
Output_Raster__3_ =
Output_Location__ex_____Data_Generated_Grids_ver4_Results_ver5_combine_CA_NW_
Output_Raster__4_ =
Output_Location__ex_____Data_Generated_Grids_ver4_Results_ver5_combine_CA_NW_
Output_Raster__5_ =
Output_Location__ex_____Data_Generated_Grids_ver4_Results_ver5_combine_CA_NW_
Output_Raster__6_ =
Output_Location__ex_____Data_Generated_Grids_ver4_Results_ver5_combine_CA_NW_
Output_Raster__7_ =
Output_Location__ex_____Data_Generated_Grids_ver4_Results_ver5_combine_CA_NW_
Output_Raster__8_ =
Output_Location__ex_____Data_Generated_Grids_ver4_Results_ver5_combine_CA_NW_

# Process: Mosaic To New Raster
arcpy.MosaicToNewRaster_management(Input_Riparian_Areas_unclipped_GA_Grids__ex__cagarpayr_
_nwgarpayr_,
Output_Location__ex_____Data_Generated_Grids_ver4_Results_ver5_combine_CA_NW_,
Output_Riparian_Areas_Mosaic_GA_Grid__ex__canwrpayr_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
```

```
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")

# Process: Mosaic To New Raster (2)
arcpy.MosaicToNewRaster_management(Input_Recreation_Sites_unclipped_GA_Grids__ex__cagarcayr
__nwgarcayr_,
Output_Location__ex_____Data_Generated_Grids_ver4_Results_ver5_combine_CA_NW_,
Output_Recreation_Sites_Mosaic_GA_Grid__ex__canwrcayr_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")

# Process: Mosaic To New Raster (3)
arcpy.MosaicToNewRaster_management(Input_Infrastructure_unclipped_GA_Grids__ex__caganfayr__
nwganfayr_, Output_Location__ex_____Data_Generated_Grids_ver4_Results_ver5_combine_CA_NW_,
Output_Infrastructure_Mosaic_GA_Grid__ex__canwnfayr_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")

# Process: Mosaic To New Raster (4)
arcpy.MosaicToNewRaster_management(Input_Developed_Areas_unclipped_GA_Grids__ex__cagadvay
r__nwgadvayr_,
Output_Location__ex_____Data_Generated_Grids_ver4_Results_ver5_combine_CA_NW_,
Output_Developed_Areas_Mosaic_GA_Grid__ex__canwdvayr_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")

# Process: Mosaic To New Raster (5)
arcpy.MosaicToNewRaster_management(Input_Sage-
Grouse_unclipped_GA_Grids__ex__cagasgayr__nwgasgayr_,
Output_Location__ex_____Data_Generated_Grids_ver4_Results_ver5_combine_CA_NW_,
Output_Sage-Grouse_Mosaic_GA_Grid__ex__canwsgayr_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
```

```
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")

# Process: Mosaic To New Raster (6)
arcpy.MosaicToNewRaster_management(Input_Drinking_Water_unclipped_GA_Grids__ex__cagadwayr
__nwgadwayr_,
Output_Location__ex_____Data_Generated_Grids_ver4_Results_ver5_combine_CA_NW_,
Output_Drinking_Water_Mosaic_GA_Grid__ex__canwdwayr_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")

# Process: Mosaic To New Raster (7)
arcpy.MosaicToNewRaster_management(Input_Threatened___Endangered_unclipped_GA_Grids__ex_
_cagateayr__nwgateayr_,
Output_Location__ex_____Data_Generated_Grids_ver4_Results_ver5_combine_CA_NW_,
Output_Threatened___Endangered_Mosaic_GA_Grid__ex__canwteayr_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")

# Process: Mosaic To New Raster (8)
arcpy.MosaicToNewRaster_management(Input_Forest_Areas_unclipped_GA_Grids__ex__cagaftayr__n
wgaftayr_, Output_Location__ex_____Data_Generated_Grids_ver4_Results_ver5_combine_CA_NW_,
Output_Forest_Areas_Mosaic_combine_GA_Grid__ex__canwftayr_,
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',
SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532
925199433]],PROJECTION['Albers'],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0],P
ARAMETER['Central_Meridian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.5],PARAMETER['Lat
itude_Of_Origin',23.0],UNIT['Meter',1.0]]", "8_BIT_UNSIGNED", "270", "1", "MAXIMUM", "FIRST")
```