Description of New Climate Library and Associated Changes in Century, BDA and Multi-Regime Fire Extensions

Melissa Lucash, Robert Scheller and Alec Kretchun

Document completed on 4/15/2014

This document served to guide development of the climate library, but is no longer up-to-date. Please refer to the Climate Library User's Guide for the most up-to-date information about the climate library and associated changes in the Century, BDA and Multi-Regime Fire Extensions.

For questions about the climate library, contact Dr. Robert Scheller (rmschell@pdx.edu) or Dr. Melissa Lucash (lucash@pdx.edu).

This work was funded by the USDA AFRI grant #OREW-2012-00978.

Table of Contents

Overall description

A significant recently emerging issue for projecting landscape-scale climate effects using LANDIS-II is the need to seamlessly integrate climate inputs across multiple, interacting ecological processes, such as forest growth and decomposition, drought, wildfire, insect outbreaks, and storm intensity. The aim of this project is to tightly integrate a suite of LANDIS-II model extensions that all ‘feed’ off of the same stream of climate data.

First, we will develop an integrated climate library that will directly utilize outputs available from PRISM (baseline or historic climate data) and the USGS Geo Data Portal (climate change data). These data will be aggregated to the ecoregion level, the fundamental climate unit of LANDIS-II and are delivered as monthly or daily data (minimum and maximum temperature, precipitation) for the requested time period in a common format (comma delimited with a header). These data will be read into a new climate library (a sharable body of code) that will perform all necessary pre-processing for all climate-dependent LANDIS-II extensions. For example, the Dynamic Fire extension (Sturtevant et al. 2009b) requires daily Fire Weather Indices (FWI), that are calculated from seasonal temperature and precipitation trends. The climate library will perform these calculations, eliminating the need for external pre-processing, and will make these data visible to any other extension.

The relevant extensions (listed below) will be revised so that they are all using data provided by the climate library. Each extension requires slightly different climate data inputs; the succession extension will serve as the nominal controller of the climate library (activating it with necessary input file(s)). Such deep integration across ecological processes (extensions) will allow LANDIS-II to respond to climate in a coordinated fashion at each model time step and will allow climate variability to produce realistic emergent properties of species composition, disturbance regimes, and ecosystem dynamics (e.g., carbon cycling). This integration will also facilitate rapid deployment and will minimize the pre-processing overhead typical of many landscape models.

Task #1: Modify BDA to use the new climate library: PDSI

The first priority is to revise the climate library such that it will calculate PDSI (Palmer Drought Severity Index, index of soil drought). PDSI is a soil water supply-demand model developed in the 1960’s that is designed to estimate surplus (wet spell) or deficiency (drought) of soil moisture. The values range from -6 to 6, with lower values representing more severe drought conditions.

Calculated PDSI will feed into the BDA (biological disturbance agent, i.e. insects) extension. Within the current project, the BDA extension will be used to simulate bark beetles, an endemic but occasionally harmful group of insect throughout the Western United States. There is an intrinsic and observed ecological link between drought periods and beetle outbreaks - as trees are weakened by drought stress they become more vulnerable to bark beetle attacks. The goal of the PDSI-BDA link is to represent this ecological link by coordinating drought (as represented by PDSI) with the initiation of bark beetle outbreaks (as modeled with the BDA extension). This update will be responsible for the temporal dynamics of BDA outbreaks, as an optional replacement for the current 'Regional Outbreak Inputs'. Use of PDSI within the BDA will be an optional step, such that the user will decide whether to use the current design (ROS) or this new approach (PDSI-initiated outbreaks). This is because BDA works with some extensions (like Age-Only Succession) that lack the information needed to use this new interface.

The climate library for the Century extension currently contains monthly values for temperature and precipitation for each of the ecoregions in the landscape (Figure 1). An ecoregion is an area of the landscape which has similar temperatures and precipitation. There can be one ecoregion per landscape or more like 30 ecoregions, depending on the project. As an example, there are 7 ecoregions (e.g. uplow, upmed) in the NJ Pine Barrens project.

```
LandisData "Climate Data"

ClimateTable
>>Eco    Time    Month    AvgMinT  AvgMaxT  StdDevT  AvgPpt  StdDev  PAR
>>Index  Step             (C)      (C)      (K)      (cm)    Ppt     µmol m-2 s-1
uplow    0       1        -10.9    10.2     3.5      8.7     4.0     0.0
uplow    0       2        -8.8     10.2     3.4      7.7     3.3     0.0
uplow    0       3        -3.0     14.7     3.2      9.2     3.9     0.0
uplow    0       4        2.9      19.0     2.4      9.0     4.6     0.0
uplow    0       5        8.9      23.7     2.1      8.4     4.2     0.0
uplow    0       6        14.9     26.8     1.6      6.8     3.6     0.0
uplow    0       7        18.6     28.5     1.4      9.8     5.9     0.0
uplow    0       8        17.2     27.4     1.6      10.6    6.8     0.0
uplow    0       9        11.5     25.2     1.8      7.4     4.0     0.0
uplow    0       10       4.6      20.7     2.2      7.2     3.6     0.0
uplow    0       11       -0.9     16.6     2.3      8.9     5.2     0.0
uplow    0       12       -7.5     12.4     3.3      8.3     4.2     0.0
upmed    0       1        -10.9    10.2     3.5      8.7     4.0     0.0
upmed    0       2        -8.8     10.2     3.4      7.7     3.3     0.0
upmed    0       3        -3.0     14.7     3.2      9.2     3.9     0.0
upmed    0       4        2.9      19.0     2.4      9.0     4.6     0.0
upmed    0       5        8.9      23.7     2.1      8.4     4.2     0.0
upmed    0       6        14.9     26.8     1.6      6.8     3.6     0.0
upmed    0       7        18.6     28.5     1.4      9.8     5.9     0.0
upmed    0       8        17.2     27.4     1.6      10.6    6.8     0.0
upmed    0       9        11.5     25.2     1.8      7.4     4.0     0.0
upmed    0       10       4.6      20.7     2.2      7.2     3.6     0.0
upmed    0       11       -0.9     16.6     2.3      8.9     5.2     0.0
upmed    0       12       -7.5     12.4     3.3      8.3     4.2     0.0
uphigh   0       1        -10.9    10.2     3.5      8.7     4.0     0.0
```

Figure 1. Example of the climate input file for Century Extension of LANDIS-II. Input file contains Ecoregion (e.g. uplow, upmed), TimeStep (year), Month, AvgMinT (average minimum temperature), AvgMaxT (average maximum temperature), StdDevT (standard deviation of temperature), AvgPpt

(average precipitation), StdDev Ppt (standard deviation of precipitation), and PAR (which are the light levels).

For this project we will be using PDSI values calculated at a monthly timestep. From that we will get average annual PDSI, the value that will ultimately be used by the BDA extension.

In order to calculate PDSI based upon these values, the following information is necessary: monthly temperature (T) , monthly precipitation (Ppt) , average monthly temperature, available water holding capacity, and latitude. The monthly T and monthly Ppt are the current month's values - the values produced by the Climate Library within the current simulation year.

The average monthly temperature represents the base climatology for a region, the 'historic' data from which PDSI is calculated. NOAA uses a calibration period of 1931-1990 in their monthly calculation of PDSI for the US. In projecting PDSI, Wehner et al (2011) used a period of 1950-1999 to get these average values. We will use the Climate LIbrary to produce these monthly average values. Basically, this approach uses the model climate inputs from time step=0 (Figure 1) to create a period (e.g. 30 simulated years) of monthly temperature using the Climate Generator already within LANDIS-II. The annual averages for each month with be used for these average monthly T values.

AWC (available water holding capacity) can be found by retrieving 2 values from the CENTURY succession input file (typically labeled something like "century-succession.txt"). An example is in Figure 2 below. AWC is calculated as Field capacity - Wilting point, the values for which are under the 'EcoregionParameters' section of the CENTURY input file. The definitions of each of these parameters within the CENTURY Users Guide are as such.

2.19.4 Field Capacity, Wilting Point

Fraction of the soil depth. Field capacity and wilting point are

calculated as this fraction multiplied by soil depth.

The final value is the latitude for the study landscape. This can be found relatively easily within the same CENTURY input file. The latitude for the Lake Tahoe Basin is 39.02N

```
EcoregionParameters
>>       Soil    Percent Percent Field   Wilt
>>       Depth   Clay    Sand    Cap     Point
>>       cm      frac    frac
uplow    100     0.045   0.894   0.104   0.035
upmed    100     0.102   0.76    0.143   0.066
uphigh   100     0.082   0.784   0.14    0.059
wetlow   100     0.048   0.889   0.118   0.043
wetmed   100     0.109   0.727   0.167   0.081
wethigh  100     0.042   0.895   0.128   0.053
plains   100     0.095   0.768   0.136   0.061
```

Figure 2. Excerpt from the Century input file.

This approach was modified from a PDSI calculation tool called the Green Leaf Project (http://greenleaf.unl.edu/) developed by the University of Nebraska-Lincoln. The UN-L tool is written in C++ and the code was converted to C#.

A full set of equations for calculating PDSI can be found in Palmer (1965) Meteorological Drought. Research Paper No 45:

Four potential values are computed:
1.Potential evapotranspiration (PE, e.g. by Hargreaves equation or other),
2.Potential recharge (PR) - the amount of moisture required to bring the soil to field capacity. 3.Potential loss (PL) - the amount of moisture that could be lost from the soil to evapotranspiration provided precipitation during the period was zero.
4.Potential runoff (PRO) - the difference between the potential precipitation and the PR

The climate coefficients are computed as a proportion between averages of actual versus potential values for each of 12 months. These climate coefficients are used to compute the amount of precipitation required for the Climatically Appropriate for Existing Conditions (CAFEC). The difference, *d*, between the actual (*P*) and CAFEC precipitation ($\hat{p}$) is an indicator of water deficiency for each month.

$$d = P - \hat{P} = P - (\alpha PE + \beta PR + yPRO + \delta PL)$$
(*1*)

$$a = \overline{ET} / \overline{PE}$$
$$\beta = \overline{R} / \overline{PR}$$
$$\gamma = \overline{RO} / \overline{PRO}$$
$$\delta = L / \overline{PL}$$

for 12 months. The value of *d* is regarded as a moisture departure from normal because the CAFEC precipitation is an adjusted normal precipitation.
A Palmer Moisture Anomaly Index (PMAI), Z, is then defined as

$Z = Kd$ (*2*)

where *K* is a weighting factor. The value of *K* is determined from the climate record before the actual model calculation. Palmer suggested empirical relationships for K such that

$$K_i = \left(\frac{17.6}{\sum_{i=1}^{12} \overline{D}_i K_i^{'}} \right) K_i^{'} \qquad (3)$$

where $\overline{D}_i$ is the average of the absolute values of d, and $K^{'}_i$ is dependent on the average water supply and demand, given by

$$K^{'}_i = 1.5 \log_{10} \left[\left(\frac{\overline{P}\overline{E} + \overline{R} + \overline{R}\overline{O}}{\overline{P} + \overline{L}} + 2.8 \right) \overline{D}^{-1} \right] + 0.5 \qquad (4)$$

where *PE* is the potential evapotranspiration, *R* is the recharge, *RO* is the runoff, *P* is the precipitation, and *L* is the loss. The PDSI is now given by

$$PDSI_i = 0.897 PDSI_{i-1} + \frac{1}{3} Z_i \qquad (5)$$

where the PDSI of the initial month in a dry or wet spell is equal to $\frac{1}{3} z_i$.

Task #2: Incorporate PDSI output from climate library into BDA

Overview

After the PDSI calculation is performed, we will incorporate it into the BDA extension, as a temporal trigger for initiating outbreaks. Previous research has shown a demonstrable link between drought and insect outbreaks (Guarin and Taylor 2005, Egan unpublished). Although a connection between beetle outbreaks and drought exists, the relationship between the severity of both disturbances is unclear, i.e. droughts of increasing severity do not always mean beetle outbreaks of increasing severity. Because of this uncertainty, the method for controlling severity within the BDA extension will not be changed. This corresponds to the MinROS and MaxROS parameters in the individual DBA agent input files.

Although PDSI will be calculated at a monthly timestep, annual average PDSI will be used within the BDA extension. This is primarily due to available information on drought and beetle outbreaks, most of which uses annual PDSI in creating statistical linkages between the two. This means replacing the current method of dictating when outbreaks occur, referred to as the ‘Regional Outbreak Status’ or ROS, with a PDSI-based threshold.. Below are the proposed changes to the BDA agent input file. These reflect updates to the currently unreleased BDA v3.0.

Proposed input format for climate-induced temporal outbreak patterns: Input parameters vary depending on selection of OutbreakPattern. There are 3 options for OutbreakPattern (CyclicNormal, CyclicUniform, and Climate), each of which have unique parameters that must be used with each particular OutbreakPattern choice. In the example sets of parameters, these unique parameters are marked with a *. For instance, in the “OutbreakPattern Climate” example below, the unique parameters for this choice are VariableName, LogicalTest, and OutbreakLag. Examples using the Cyclic functions are provided at the end of the document.
If ‘OutbreakPattern = Climate’, the extension will use climate data provided by whichever succession extension is being run to determine the temporal aspects of BDA agent outbreaks.

```
>>--------  Regional Outbreak Inputs --------
OutbreakPattern      Climate                 << CyclicNormal or CyclicUniform or Climate
>> If OutbreakPattern = CyclicNormal or CyclicUniform, then following 4 inputs should be commented out
*VariableName        AnnualPDSI              << Potentially any variable from the climate library
*LogicalTest  “< -2.0”                 << outbreak threshold
*OutbreakLag         3                       << years
TimeSinceLastClimate       1                 << years - optional

TemporalType         pulse                   << pulse or variablepulse
MinROS               0                       << background level (0-3)
MaxROS               3                       << maximum level (0-3)
```

This approach is designed to achieve maximum flexibility when applied to the climate library being developed.   The VariableName input above gives the future developers the flexibility to add new functions that summarize climate data in unique ways. For example, one could add a function “JanuaryMinimumTemperature” that would be added to the climate library and would return those specific data. The example inputs above would use the annual mean landscape PDSI value, and if that value is < -2.0, then an outbreak would be triggered 3 years from the current time step. The OutbreakLag is used to reset the BDA’s internal counter that tracks time to next outbreak.

An additional consideration needs to be made to allow multiple outbreaks to occur within a timespan less than OutbreakLag. For example, 2 consecutive drought years should trigger outbreaks in consecutive years. If the OutbreakLag is used to simply overwrite TimeToNextDisturbance, then it will not handle consecutive outbreaks correctly (only the second would occur). A possible solution to this is to construct a continuously updated list of time steps that meet the LogicalTest (this would be an agent parameter). Then that list is consulted each time step when the TimeToNext has the opportunity to be updated. So, when a new TimeToNext is assigned, it does not cancel out any pending outbreaks.

TimeSinceLastClimate
The time since the last Climate event (TimeSinceLastClimate) is an optional input to allow outbreaks to occur in the first years of simulation before the outbreak lag has passed. If TimeSinceLastClimate > OutbreakLag or TimeSinceLastClimate is excluded, then no outbreak will occur until OutbreakLag has passed.

Future development

Ecoregion-scale considerations
Due to spatial differences among ecoregions (see below), the time to next outbreak may need to be tracked as a site variable or potentially an ecoregion variable. The spatial differences among ecoregions create some challenges to applying the BDA. The RestrictionMethod input above attempts to get at how the spatial differences in climate at the ecoregion level would impact an outbreak. This input is only used when FunctionScale = ecoregion. When using the landscape FunctionScale there are no ecoregion differences. Assuming that climate is calculated at an ecoregion level, then some portions of the landscape may cross the climate threshold while others do not. This would then create a spatial mask for areas that do meet the threshold by altering the ecoregion modifiers. Using the first RestrictionMethod ('Initiation'), those modified values could be applied to limit only the initiation of new outbreak epicenters (if using dispersal), which would restrict epicenters to ecoregions meeting the threshold, but allow an outbreak to spread into areas not meeting the climate threshold. The second RestrictionMethod option ('Disturbance') is designed to also restrict the spread of disturbance to areas meeting the climate threshold, so that the outbreak is completely limited to areas meeting the threshold (all ecoregion modifiers not meeting threshold set to -1.0). If not using the dispersal functions, then the Disturbance option is the only logical choice.

Example input formats for the Cyclic OutbreakPatterns:
```
>>--------  Regional Outbreak Inputs --------
OutbreakPattern      CyclicNormal << CyclicNormal or CyclicUniform or Climate
*Mean                20                 << mean years between outbreaks
```

```
*StDev                          15                  << standard deviation (years)
*TimeSinceLastEpidemic          20        << years
TemporalType        pulse                 << pulse or variablepulse
MinROS              0                     << background level (0-3)
MaxROS              3                     << maximum level (0-3)

>>--------  Regional Outbreak Inputs --------
OutbreakPattern     CyclicUniform         << CyclicNormal or CyclicUniform or Climate
*MaxInterval  20                << maximum years between outbreaks
*MinInterval  15                << minimum years between outbreaks
*TimeSinceLastEpidemic          20        << years
TemporalType        pulse                 << pulse or variablepulse
MinROS              0                     << background level (0-3)
MaxROS              3                     << maximum level (0-3)
```

Here is a reference chart for knowing which parameters are necessary with the different Outbreak pattern options. The three options are Climate, CyclicNormal, and CyclicUniform.

Parameters that will always be used, regardless of OutbreakPattern choice	
OutbreakPattern	<< CyclicNormal or CyclicUniform or Climate
TemporalType	<< pulse or variablepulse
MinROS	<< background level (0-3)
MaxROS	<< maximum level (0-3)
Parameters that will only be used if OutbreakPattern = Climate	
VariableName	<< Potentially any variable from the climate library
LogicalTest	<< outbreak threshold
OutbreakLag	<< years
Parameters that will only be used if OutbreakPattern = CyclicNormal	
Mean	<< mean years between outbreaks
StDev	<< standard deviation (years)
TimeSinceLastEpidemic	<< years
Parameters that will only be used if OutbreakPattern = CyclicUniform	
MaxInterval	<< maximum years between outbreaks
MinInterval	<< minimum years between outbreaks
TimeSinceLastEpidemic	<< years

Task #3: Improve user interface between the Century Extension and the climate library

In previous versions of Century (v3.1 and earlier), the user provided a climate input file with monthly data. This option will no longer be an option in the new climate library.

Instead, the user will use climate change data directly from the USGS data portal and have the climate library process the data for the "future" climate, i.e. climate for time steps ≥0 (Figure 4). This data will need to be at a **daily or a monthly time step**. In the Century input file, the user will need to tell LANDIS the name of the climate configuration file (Figure 4).

```
LandisData  "Century Succession"

Timestep  10

SeedingAlgorithm  WardSeedDispersal

InitialCommunities       "ic_MNstate7000_8.15.13.txt"

InitialCommunitiesMap    "ICMAP_L6M1.img"

ClimateConfigFile        "climate-generator-CC.txt"
```

Figure 3. The user will need to have the keyword ClimateConfigFile with the name of climate file after it. In this example "climate-generator-CC.txt" is the file that describes the types of climate data, formatting and the path/file names.

The climate configuration file specifies the formats, file names and formatting of all the climate data. The user can either choose to use <u>climate change data</u>

```
**********************************************************************
#Use daily data from the USGS data portal to simulate climate change.

LandisData "Climate Config"


ClimateTimeSeries                       DailyGCM
ClimateFile                             "I:/directory/GCM_data.txt"
ClimateFileFormat                       GriddedObserv

SpinUpClimateTimeSeries                 MonthlyAverage <MonthlyRandom>
SpinUpClimateFile                       "I:/directory/PRISMclimate.csv"
SpinUpClimateFileFormat                 PRISM
```

Figure 5. Climate configuration file. This is the format for use with climate change scenarios.

or the user could also decide to run a “baseline” scenario, where we assume that there will be no climate change occurring in his/her landscape. The user could use the average climate data to simulate climate (DailyHistAverage) or randomly select climate from the climate data (DailyHistRandom). Either way, this data will need to be at a **daily time step**.

```
#Use data from the USGS data portal to simulate baseline climate.
#Uses average historic data

LandisData "Climate Config"

ClimateTimeSeries                      DailyHistAverage
ClimateFile                            "I:/directory/dailyclimate.csv"
ClimateFileFormat                      GriddedObserv

SpinUpClimateFileStream                MonthlyAverage  <MonthlyRandom>
SpinUpClimateFile                      "I:/directory/PRISMclimate.csv"
SpinUpClimateFileFormat                PRISM


*************************************************************************
#Use data from the USGS data portal to simulate baseline climate.
#Randomly selects climate each year.

LandisData "Climate Config"

ClimateFileStream                      DailyHistRandom
ClimateFile                            "I:/directory/dailyclimate.csv"
ClimateFileFormat                      GriddedObserv

SpinUpClimateFileStream                MonthlyAverage <MonthlyRandom>
SpinUpClimateFile                      "I:/directory/PRISMclimate.csv"
SpinUpClimateFileFormat                PRISM
```

Figure 6. Climate configuration file. This is the format for use with baseline scenarios (NOT climate change scenarios).

These are the following climate file formats that are commonly used right now, but we’ll add more as needs change.

Table 1. Data sources, file formatting and the time step

Type of data	**ClimateFileFormat**	**Time step**
Climate change Projections	GriddedObserv	Daily
PRISM	PRISM	Monthly
Historical (Mauer dataset)	GriddedObserv	Daily

This climate data will be used to “simulate” the climate for each year of the simulation using the current climate generator in LANDIS.

Task #4: Process daily climate change data from USGS Data Portal

The USGS-GDP serves downscaled (12 km resolution) data **projected** from multiple global circulation models and multiple emissions scenarios. The user can upload a shape file to their web site that enables their web server to parse the landscape by ecoregion. The data is then downloaded by the user as **daily** means, variances and standard errors for minimum temperature (tmin), maximum temperature (tmin) and mean precipitation (pr) for each climate region for the requested time period in a common format (comma delimited with a header, Figure 3). At this time, we are not using the variances and standard errors from the USGS data portal. These represent variation in the climate between grid cells; this is a small source of variation so we are currently omitting this variation. We are only using the averages at this point.

The user would need to parse the data by GCM and emissions scenario so that each input file contains one climate change scenario (eg. Bcm2_a1b).

	A	B	C	D	E	F
1	# bcm2_a1b_tmax					
2		1	2	3	4	5
3	TIMESTEP	MINIMUM(degreesC)	MINIMUM(degreesC)	MINIMUM(degreesC)	MINIMUM(degreesC)	MINIMUM(degreesC)
4	2000-01-01T00:00:00Z	-8.16	-8.19	-7.8	-8.9	-8.09
5	2000-01-02T00:00:00Z	-7.07	-6.57	-6.1	-7.01	-6.25
6	2000-01-03T00:00:00Z	-0.73	-0.38	-0.24	-2.54	-0.4
7	2000-01-04T00:00:00Z	-4.55	-4.42	-4.18	-5.96	-4.58
8	2000-01-05T00:00:00Z	-8.03	-7.73	-8.51	-8.36	-9
9	2000-01-06T00:00:00Z	-7.08	-7.03	-8.27	-8.29	-9.13
10	2000-01-07T00:00:00Z	-2.48	-2.62	-3.31	-4.33	-4.12
11	2000-01-08T00:00:00Z	-1.67	-1.51	-2.3	-4.35	-3.61
12	2000-01-09T00:00:00Z	-1.65	-1.66	-2.07	-4.79	-3.71
13	2000-01-10T00:00:00Z	-1.76	-1.78	-1.64	-4.28	-2.46
14	2000-01-11T00:00:00Z	-4.06	-5.3	-5.02	-7.83	-6.56
15	2000-01-12T00:00:00Z	-15.08	-14.57	-14.13	-15.8	-15.31
16	2000-01-13T00:00:00Z	-20.58	-18.08	-17.42	-17.43	-16.75
17	2000-01-14T00:00:00Z	-18.37	-15.91	-15.14	-15.11	-14.5
18	2000-01-15T00:00:00Z	-14.61	-12.68	-11.98	-11.97	-11.33

Figure 7. Example of USGS data portal output file from one scenario (bcm2_a1b_tmax). In this excerpt of the output file, there are 5 ecoregions and this is showing the minimum temperature (tmin) as an example.

If there are multiple soil soil regions within each climate region, the user will need to copy the climate regions so that each ecoregion has a climate. For example, in the CNF+ landscape, we have five climatic regions (i.e. five polygons) so we download data from the USGS data portal for the five regions. Then we copy the data from each climate region for each of the soil regions for a total of 25 ecoregions (5 climate regions * 5 soil regions = 25 ecoregions).

The user will need to adjust the headers, with the **climate regions starting at 1** if the first ecoregion is inactive in the ecoregion.txt file (e.g. Figure 7). If there are no inactive ecoregions, the user will need to start numbering the **climate regions at zero**. The user will also need to have the correct key words (though the words are not case sensitive) to identify the data (i.e. if it's max or minimum temperature). The relative humidity and wind speed data are optional when you run LANDIS, Century Extension. They are only needed if you run the fire extension.

Table 1. Key words needed in climate input file

Maximum temperature	Minimum temperature	Precipitation	Relative humidity	Wind speed
# Tmax	# Tmin	# Prcp	#rh	#windspeed
# maxtemp	# mintemp	# ppt	#RH	#windSpeed

These data will then be read into the new climate library which will store the data by ecoregion. This library will then provide all the necessary climate data for the other extensions, keeping in mind that the some extensions require monthly climate data (e.g. Century succession, drought extension, insect leaf biomass extension), while others require daily climate data (e.g. the Dynamic Fire extension). The fire extension will use the daily data from the climate library (see task 7) to make all the calculations needed to simulate fire.

For the Century succession extension, the climate library will convert daily data to monthly data for the following parameters: maximum temperature, minimum temperature, and precipitation . To convert from daily to monthly data, the climate library will treat temperatures differently than precipitation, since the temperatures reflect monthly averages and the precipitation is a monthly sum (cumulative). Also, the original precipitation data from the USGS data portal is reported in mm/day, but LANDIS uses cm/day or cm per month. So the precipitation values need to be divided by 10. For min and max temperature, it will simply take the average and standard deviaton data for each month. For precipitation, the climate library will sum all the daily precipitation and calculate the standard deviation for that month.

When the user runs Century, the climate library will produce a file that shows all the summarized data in the "old" format (see Figure 3).
When the user runs LANDIS, the model produces a century-succession-monthly-log.csv file which summarizes the temperature and precipitaiton data for the model run. When the model is run, it selects the temperature or precipitation for that month, but then randomly selects the actual value by sampling within the distribution of values which is determined by the standard deviation for that month.

Task #5: Calculating climate during spin-up

The climate for the spin-up phase of the model and the baseline scenarios is currently supplied using a climate input file. This data is used to calculate an 'actual' climate, which is generated from the means and standard deviations (SD) of temperature and precipitation.

Alternatively, the user can now download daily data from the USGS-GDP, which is **actual daily** historical temperature and precipitation data (1950 to 1999) and then allow the climate library to generate the climate inputs internally. Using the same procedures as outline in Task #4, the user can upload a shape file and download the monthly max temperature, minimum temperature and precipitation for each ecoregion for the requested time period in a common format (comma delimited with a header). These data will be read into the new climate library that will parse the data by ecoregion and provide all the necessary climate data for the other extensions.

If the user chooses "SpinUpClimateFile Format=AverageMonthly" to generate the climate during spin-up, the SD is calculated by the climate library and the data are treated the same as data coming from the current method (ClimateFileFormat=Monthly Standard).

Alternatively, the user could decide to allow the climate library to randomly choose years from the total set and use these as the 'actual' data (SpinUpClimateFileFormat=Random).

Task #6: Restructuring the new climate library

Century is currently selecting the climate data and storing it, but this will be moved into the climate library. The climate library will generate all the climate, store it as daily data for fire and then store it as monthly data for Century. The climate library will keep the spin-up and "future" (historic or GCM data) climate separate and will need to be initialized by Century. The climate library will use the ClimateFile to determine the the first year of the simulation. For example, if the user downloads climate data from a GCM starting at year 2010, then simulation year=1 will correspond to year 2010.

Task #7 Calculating fire parameters under climate change in the climate library

Overview

The Dynamic Fire extension (Sturtevant et al. 2009b) requires daily climate data to calculate the fine fuel moisture code (FFMC), burn index (BUI), wind speed velocity (WSV), wind direction or azimuth (WindDir), fire weather index (FWIBin) and season of year (Season) to initialize the extension. These calculations are currently calculated by the user and supplied to the extension in the input file in the InitialWeatherDatabase (baseline scenario) and DynamicWeatherTable (climate change scenarios, see Figure 4). This will continue to be an option for the user.

```
InitialWeatherDatabase  "I:/research/samba/scheller_lab/Lucash/Century-version-3/FHEM_input_files/ClimateChange_Fire/hadcm3_a2_2000-2004.csv"

DynamicWeatherTable
5       "I:/research/samba/scheller_lab/Lucash/Century-version-3/FHEM_input_files/ClimateChange_Fire/hadcm3_a2_2005-2009.csv"
10      "I:/research/samba/scheller_lab/Lucash/Century-version-3/FHEM_input_files/ClimateChange_Fire/hadcm3_a2_2010-2014.csv"
15      "I:/research/samba/scheller_lab/Lucash/Century-version-3/FHEM_input_files/ClimateChange_Fire/hadcm3_a2_2015-2019.csv"
20      "I:/research/samba/scheller_lab/Lucash/Century-version-3/FHEM_input_files/ClimateChange_Fire/hadcm3_a2_2020-2024.csv"
25      "I:/research/samba/scheller_lab/Lucash/Century-version-3/FHEM_input_files/ClimateChange_Fire/hadcm3_a2_2025-2029.csv"
30      "I:/research/samba/scheller_lab/Lucash/Century-version-3/FHEM_input_files/ClimateChange_Fire/hadcm3_a2_2030-2034.csv"
35      "I:/research/samba/scheller_lab/Lucash/Century-version-3/FHEM_input_files/ClimateChange_Fire/hadcm3_a2_2035-2039.csv"
40      "I:/research/samba/scheller_lab/Lucash/Century-version-3/FHEM_input_files/ClimateChange_Fire/hadcm3_a2_2040-2044.csv"
45      "I:/research/samba/scheller_lab/Lucash/Century-version-3/FHEM_input_files/ClimateChange_Fire/hadcm3_a2_2045-2049.csv"
50      "I:/research/samba/scheller_lab/Lucash/Century-version-3/FHEM_input_files/ClimateChange_Fire/hadcm3_a2_2050-2054.csv"
55      "I:/research/samba/scheller_lab/Lucash/Century-version-3/FHEM_input_files/ClimateChange_Fire/hadcm3_a2_2055-2059.csv"
60      "I:/research/samba/scheller_lab/Lucash/Century-version-3/FHEM_input_files/ClimateChange_Fire/hadcm3_a2_2060-2064.csv"
65      "I:/research/samba/scheller_lab/Lucash/Century-version-3/FHEM_input_files/ClimateChange_Fire/hadcm3_a2_2065-2069.csv"
70      "I:/research/samba/scheller_lab/Lucash/Century-version-3/FHEM_input_files/ClimateChange_Fire/hadcm3_a2_2070-2074.csv"
75      "I:/research/samba/scheller_lab/Lucash/Century-version-3/FHEM_input_files/ClimateChange_Fire/hadcm3_a2_2075-2079.csv"
80      "I:/research/samba/scheller_lab/Lucash/Century-version-3/FHEM_input_files/ClimateChange_Fire/hadcm3_a2_2080-2084.csv"
85      "I:/research/samba/scheller_lab/Lucash/Century-version-3/FHEM_input_files/ClimateChange_Fire/hadcm3_a2_2085-2089.csv"
90      "I:/research/samba/scheller_lab/Lucash/Century-version-3/FHEM_input_files/ClimateChange_Fire/hadcm3_a2_2090-2094.csv"
95      "I:/research/samba/scheller_lab/Lucash/Century-version-3/FHEM_input_files/ClimateChange_Fire/hadcm3_a2_2095-2099.csv"
```

Figure 8. Current input data for DFFS.

If instead, the user would like to let the new climate library perform the input calculations, then the new climate library will internally calculate FFMC, BUI, WSV, WD, FWIBin and Season. Since the new climate library will perform these calculations for the user, this will eliminate the need for external pre-processing, and will make these data visible to other extensions.   However, the user would also need to supply a wind input file and the FireCalcTable, which is list of 13 input parameters (FineFuelMoistureCode through PercLowFWI) in the DFFS input file (Figure 9). Files for the InitialWeatherDatabase and DynamicWeatherTable would not be necessary (Figure 8) would not be necessary.

```
WindDirectionFile       "I:/research/samba/scheller_lab/Lucash/Century-version-3/FHEM_input_files/ClimateChange_Fire/wind_azimuth.txt"

        FineFuelMoistureCode    85
        StartDMC                6
        StartDC                 15
        MinFWI                  10
        PercExtremeFWI          97
        PercVeryHighFWI         90
        PercHighFWI             75
        PercModFWI              50
        PercLowFWI              0
```

Figure 9. Example of the climate input file if the DFFS extension is in use and the user wants all the weather parameters to be processed internally in the new climate library.

Julian seasonal days (e.g. SpringJulianDate, SummerJulianDate) used to be input parameters in the input file but they will now be calculated internally within the DFFS extension. This will allow these four values to change with changing climate projections, rather than the fixed numbers in the input file.

The user would provide an input file (e.g. wind_azimut.txt) containing wind direction in txt or csv format (Figure 9) that would be downloaded from a nearby meteorological weather station (e.g. http://fam.nwcg.gov/fam-web/weatherfirecd/state_data.htm). The user could download as many years of data as desired. Ideally the wind direction would be downloaded from the USGS data portal and incorporated into the climate library itself, but it is not available as of 11.14.13. Therefore it needs to be an input file into DFFS.

To project wind direction into the future for the climate change scenarios, the climate library will assume that these parameters remain constant with climate change and vary only with the day of the year. Therefore, the climate library will randomly select wind direction from all years of available data for each julian day of the year.

Relative humidity and wind speed will both need to be included in the Climate Library input file below temperature and precipitation. This will be in the same format, with columns indicating ecoregions and demarcated sections for each individual climate variable. Rh and windspeed can be obtained through the USGS data portal but must be manually inserted into the climate input file below the other climate variables and the ecoregions renamed to match the temperature and precipitation inputs.

Then, daily relative humidity, wind speed, temperature and precipitation (all from USGS data portal) will be combined with wind direction (input file to DFFS) to calculate FFMC, BUI, WSV, WD, FWIBin and Season under the climate change scenarios. Currently, these parameters are calculated using PYTHON scripts prior to running LANDIS so the PYTHON scripts will need to be converted to C# and incorporated into the climate library.

Python script

Right now, the **inputs** for the **Python scripts** come in two files.

The **first** contains: year, Julian date, wind speed, wind azimuth, relative humidity, and precip. This file contains multiple years and days of data.

The **second** contains: Fine Fuel Moisture Code starting value, DMC (duff moisture code) starting value, DC (drought code) starting value, spring start julian date, summer start julian date, fall start julian date, winter start julian date, minimum fire weather index value, min percentile for extreme FWI class, min percentile for very high FWI class, min percentile for high FWI class, min percentile for moderate FWI class, min percentile for low FWI class. This second file only needs one value for each of the parameters, just a starting value on which it can base calculations. It then calculates all theses parameters on a daily basis forward using the climatic variables in the first file.

The **outputs** of the Python script are Fire weather index (FWI), fire weather index bin (I'm not really sure what this is yet), build up index (BUI), fine fuels moisture code (FFFMC), wind speed velocity (WSV), and season. It creates a **.csv file** with 7 columns: FFMC, BUI, WSV, WINDDir, FWIBin, Season, and Ecoregion. This .csv file is then used an input file to the dynamic fire extension (DFS). It's possible for the extension to use the same weather data (i.e. the same csv file) for all years of a simulation, as long as you specify the same file for both the "InitialWeatherDatabase" and "DynamicWeatherTable". This is usually used for 'baseline' climate scenarios. If you are running a climate change scenario, the "DynamicWeatherTable" needs to be updated every timestep with a new .csv (e.g. "DynamicWeather_a2_2010-2014.csv", "DynamicWeather_a2_2015-2020.csv", etc).

Goal of Restructuring DFS

The overall goal is to eliminate the need for the Python scripts altogether. The functionality of them will be integrated into the DFS extension itself.

The updated **Climate Library** will directly interface with all extensions, including Dynamic Fire. Which means the climate library will provide the following **daily data** to the **DFS**: min temp, max temp, precipitation, relative humidity, and windspeed. Since this data will be fed into the climate library directly, this basically takes the place of the first Python input file (see paragraph above).

New inputs for the **existing DFS extension input file** will be required to incorporate the rest of the Python inputs. These will be incorporated into the existing DFS input file. These **9 parameters** are: Fine Fuel Moisture Code starting value, duff moisture code (DMC) starting value, drought code (DC) starting value, minimum fire weather index value, min percentile for extreme FWI class, min percentile for very high FWI class, min percentile for high FWI class, min percentile for moderate FWI class, min percentile for low FWI class. This basically takes the place of the second Python input file ('sample_startingvalues_py.csv'), though the Julian start dates from that Python input have been removed. This is because these calculations will now be done within the DFS extension, to allow them to change with changing climate.

All these inputs should retain the functionality of the Python scripts, but eliminating the need for all the “InitialWeatherDatabase” and “DynamicWeatherTable” .csv’s.

Adjustment of seasons

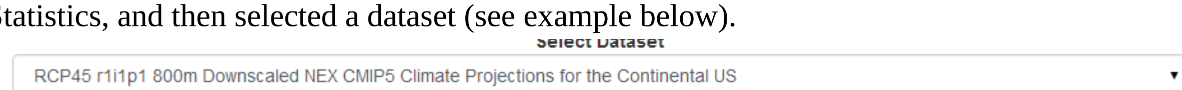
Fire occurence within the extension is adjusted based on season. The fire module selects the season for a fire event based on the proportion of fires that occur within each season, defined by the user. Fuels are also modified based on season, by determining, for example, leaf-off and –on. Seasons are determined by a Julian start date (e.g., spring starts on day 91), which is provided to the extension by the user. WE are currently looking into whether or not it would be appropriate for the determination of the season should be dynamic, allowing climate change to potentially lengthen or shorten seasons based on some criteria (for example temperature and precip).

Directions for using the new climate library

Downloading climate change and baseline data

The map of the climate regions (Note: climate regions NOT ecoregions) need to be converted to a shp file and zipped prior to downloading data.

For each climate scenario, I compiled maximum/minimum temperature, and precipitation from the downscaled Geo Data Portal for the 100 years of climate change data available. Specifically, I used the USGS data portal (http://cida.usgs.gov/gdp/) and clicked on Areal Statistics, and then selected a dataset (see example below).



Then I clicked on Process Data with the Geo Data Portal. Then I hit Upload Shapefile and uploaded the zipped climate region shp files, selected GRIDCODE as the available attribute, and Area Grid Statistics (weighted) as the algorithym. For the next menu, I left everything as the defaults, but select the Mean, Variance and SD as the statistics of interest. I selected the min temperature, max temperature and precipitation and the data range from 2000 to 2099. Hit Submit for Processing and they email you when the file is ready.

I repeated these steps to download MONTHLY PRISM data for each climate region and selecting years 1895 to 2012. These data are ACTUAL (historic) data and therefore will be used during the spin-up phase of the model and for the baseline scenario, where we assume that the climate will not change over time (i.e. no climate change).

Making the Climate Input File for Century

The climate data from the USGS data portal (see section called **Downloading climate change and PRISM data)** is downloaded for each CLIMATE region (n=5), but not ecoregion (n=25). Since LANDIS needs to have a climate for **each ecoregion**, I had to copy each climate column five times (5*5=25 ecoregions) for each of the parameters (e.g. max temp).

The columns for each ecoregion need to be numbered starting at zero (if there are no inactive ecoregions) or 1 if the first ecoregion is inactive. The climate library will only run if you have an inactive ecoregion that's first in the ecoregion.txt file.

Also the headings have to be identical to those below:
gfdl_2-1-a1fi-ppt-NAm-grid
gfdl_2-1-a1fi-maxtemp-NAm-grid
gfdl_2-1-a1fi-mintemp-NAm-grid

See clip below as an example of the input data:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	
1	# gfdl_2-1-a1fi-ppt-NAm-grid																
2		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
3	TIMESTEP	MEAN(mn	MEAN(mn	MEAN(mn	MEAN(mn	MEAN(mn	MEAN(mn	MEAN(mn	MEAN(mn	MEAN(mn	MEAN(mn	MEAN(mn	MEAN(mn	MEAN(mn	MEAN(mn	MEAN(mn	M
4	2000-01-0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	2000-01-0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6	2000-01-0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7	2000-01-0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8	2000-01-0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	2000-01-0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	2000-01-0	0.729012	0.956008	0.694332	0.729659	0.562057	0.729012	0.956008	0.694332	0.729659	0.562057	0.729012	0.956008	0.694332	0.729659	0.562057	C