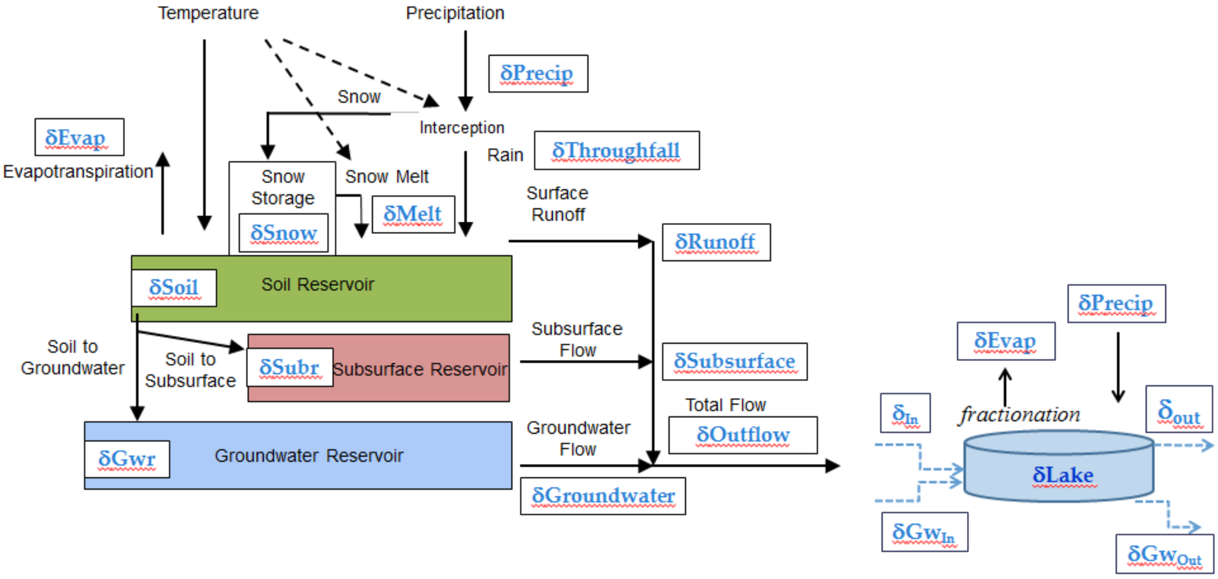
IWBMIso – An Isotope-enabled, Coupled Catchment-Lake Water Balance Model

Model Description, Users Manual, and Supporting Materials



George H. Leavesley

Dagnachew Belachew

Olaf David

David Patterson

Jack R. Carlson

Rumpal Sidhu

Mazdak Arabi

November, 2014

Table of Contents

Model Description

1.0 INTRODUCTION

Water balance models have been developed to varying degrees of complexity since the 1940s (Xu and Singh, 1998). The use of monthly water balance models has ranged from estimating runoff (Alley, 1985; Yates, 1996), to estimating the global water balance (Legates and Mather, 1992; Legates and McCabe, 2005), to evaluating the hydrologic effects of climate change (Strzepek and Yates, 1997; Wolock and McCabe, 1999).

Continued advances in hydrological research and understanding are providing new knowledge to improve the application and interpretation of water balance models. The use of stable isotopes to identify the flow paths and residence times of water in a catchment is one area of research that can be used to constrain the simulated components of the water balance (Kendall and McDonnell, 1998, Fekete et al. 2006, Stadnyk et al. 2013). Isotopes have also proven to be very useful for the quantification of surface water and groundwater flows to open and closed-basin lakes (Gonfiantini, 1986, Sets et al., 2010) and in estimating regional water balance trends (Gibson and Edwards, 2002). The increasing availability of global datasets for a variety of meteorological, hydrological, and physical measures can also assist in improving the application of water balance models. These datasets can be used for improved model parameter estimation and to provide data resources where local data measurements are limited.

The Isotope Hydrology Section of the International Atomic Energy Agency (IAEA) is a world leader in research and development of methods, databases, and tools to support the application of knowledge and data on isotopes to hydrological studies. It supports the use of isotope hydrology to improve knowledge of water resources and to improve water management. The Global Network of Isotopes in Precipitation (GNIP) and the Global Network of Isotopes in Rivers (GNIR) are two large databases initiated and maintained by IAEA to provide data on isotope content of precipitation and rivers across global scales.

The IAEA has been exploring the use of GNIP and GNIR data in hydrological models for application to hydrological studies and water resources management issues in basins around the globe. The principal premise of this effort is to contribute for the improvement of water balance models by providing an additional means of constraining model calibration and validation processes. Although this is not the first of its kind (see Stadnyk et al., 2013), our approach tries to be unique by providing a tightly coupled catchment-lake water balance model with integrated isotope mass balance for each component of the water balance at a monthly time step. A monthly water balance model was selected as the initial approach based on the typical monthly or longer frequency of isotope data availability and on an assessment of the limited spatial and temporal distribution of supporting physical and meteorological data in many of the basins. The resulting model was the IAEA Water Balance Model with Isotopes (IWBMIso).

While data limitations were a major consideration in IWBMIso development, other factors considered included user knowledge and skills, internet accessibility, and limited resources to purchase commercial software. To address these concerns and to facilitate the use of IWBMIso to a broad range of basins and users, IWBMIso is integrated with a set of associated GIS pre-processing, model application, and analysis tools using the Colorado State University Environmental Risk Assessment and Management System (eRAMS) framework. This eRAMS-based toolbox has been enabled as a Virtual Machine (VM) application that can be installed on a user's computer for application to a user's basins of interest. A detailed description of IWBMIso, the installation and application of the eRAMS VM platform, and steps required to set up and apply IWBMIso are provided in this document.

2.0 IWBMIso Design

2.1 Hydrological Design Concepts

The IWBMIso model is a monthly time step catchment water balance model tightly coupled with a monthly lake water balance model. It also simulates the oxygen-18 (O^{18}) and deuterium (H^2) stable isotope composition of the model storage and flux components. IWBMIso is a combination of concepts developed from a simple monthly water balance model (Wolock and McCabe, 1999), a monthly isotope-enabled water balance model (Fekete, et al., 2006), a daily hydrologic model PRMS (Leavesley et al. 2005), and an isotope-enabled lake model (Hostetler and Benson, 1994). The water and isotope storages and fluxes of IWBMIso are schematically shown in Figure 1.

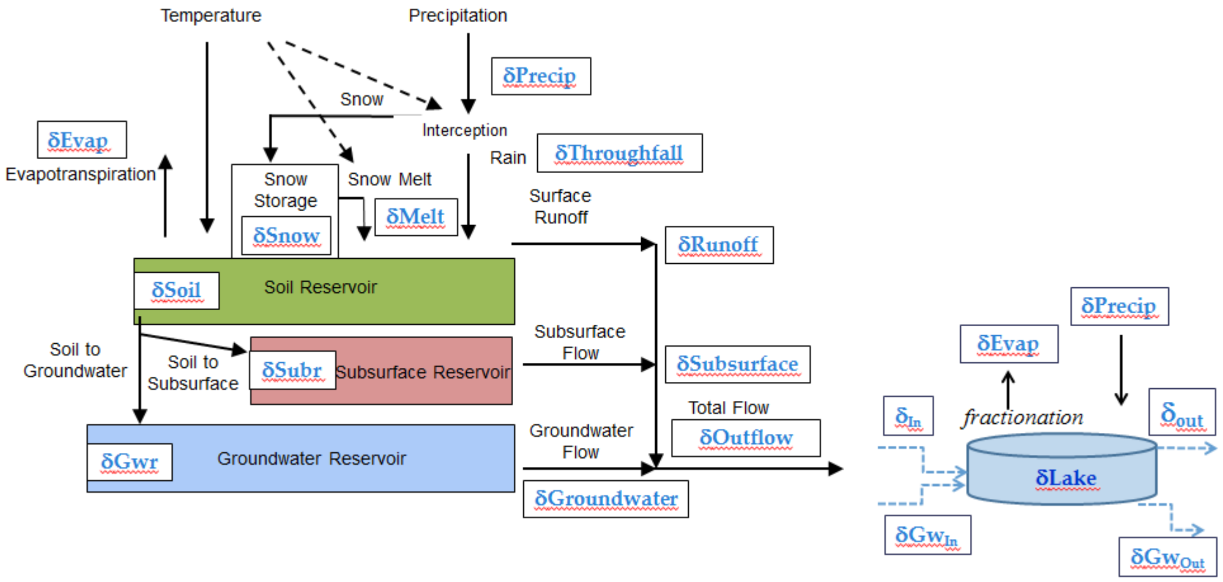


Figure 1. Water and isotope storages and fluxes

IWBMIso is a spatially distributed model that can account for some of the spatial and temporal variability in watershed characteristics and the magnitude and timing of model inputs and outputs. Delineation and characterization of a watershed for use in the water balance model is accomplished at two levels of detail.

The basic computational unit for the water balance components shown in Figure 1 is the hydrologic response unit (HRU). A HRU is a watershed subarea that is typically delineated by partitioning a watershed into units, using one or more characteristics such as slope, aspect, elevation, vegetation type, soil type, and precipitation distribution. Each unit is assumed to be homogeneous with respect to its hydrologic response and to the characteristics listed above. Precipitation and temperature inputs to each HRU can differ as a function of the HRU’s elevation and spatial location, and the number and location of climate stations available to provide precipitation and air temperature data. Water and isotope balances are computed monthly for each HRU. The sum of the responses of all HRUs, weighted on a unit-area basis, produces the monthly watershed response.

HRUs can be grouped to form subbasins within a watershed. Subbasins are typically based on defining specific streamflow nodes within a channel network where either streamflow data are collected or the nodes are points of interest for computing water balance components. Water bodies such as lakes are considered to be both individual hrus and subbasins.

The water balance components computed on each HRU within a subbasin are combined on an area weighted basis to provide the subbasin equivalent set of values. The total flow component from each subbasin is routed to its downstream subbasin and combined with the total flow of that subbasin. The routed total flow from the last downstream subbasin is the total flow for the entire watershed. Water bodies on the channel network are treated as subbasins and so are parts of the routing sequence.

The delineation of HRUs and subbasins can be accomplished using a variety of standard GIS tools. A set of GIS tools has been made available in the eRAMS platform for the HRU and subbasin delineation process. A detailed description of the tools and their use is provided in later sections of this document.

Figure 2 shows the HRU and subbasin delineation of the LakeTana basin, located in the Upper Blue Nile River basin. The outlet of the basin is the outflow of Lake Tana at Bahir Dar, Ethiopia. There are 33 HRUs and 5 subbasins in this delineation. Subbasins 1, 2, and 3 flow into subbasin 4 and subbasin 4 flows into subbasin 5, the last subbasin in the routing sequence. As a lake, the outflow from subbasin 5 is computed as a function of the elevation of the lake surface above the lake outflow point. The details of the lake water balance and outflow computations are presented in sections below.

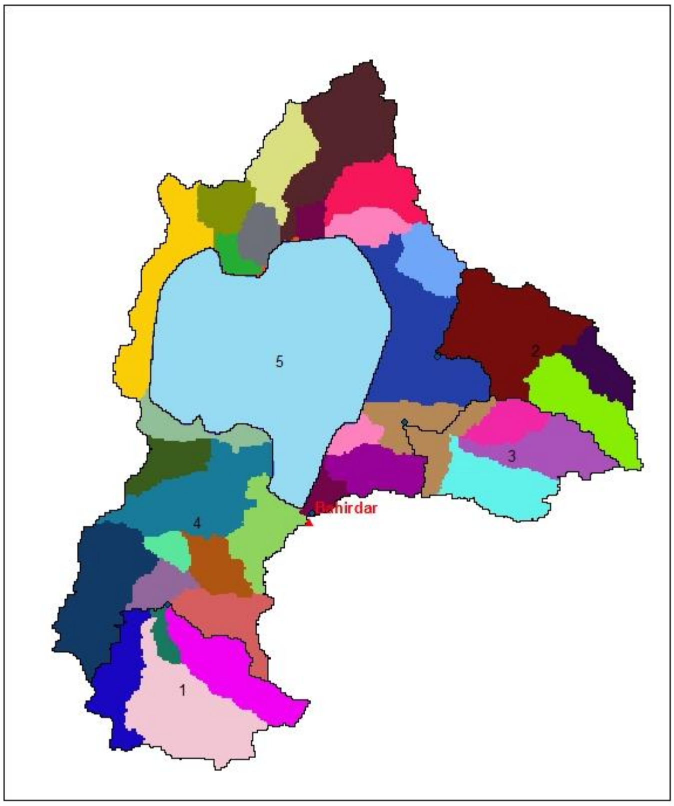


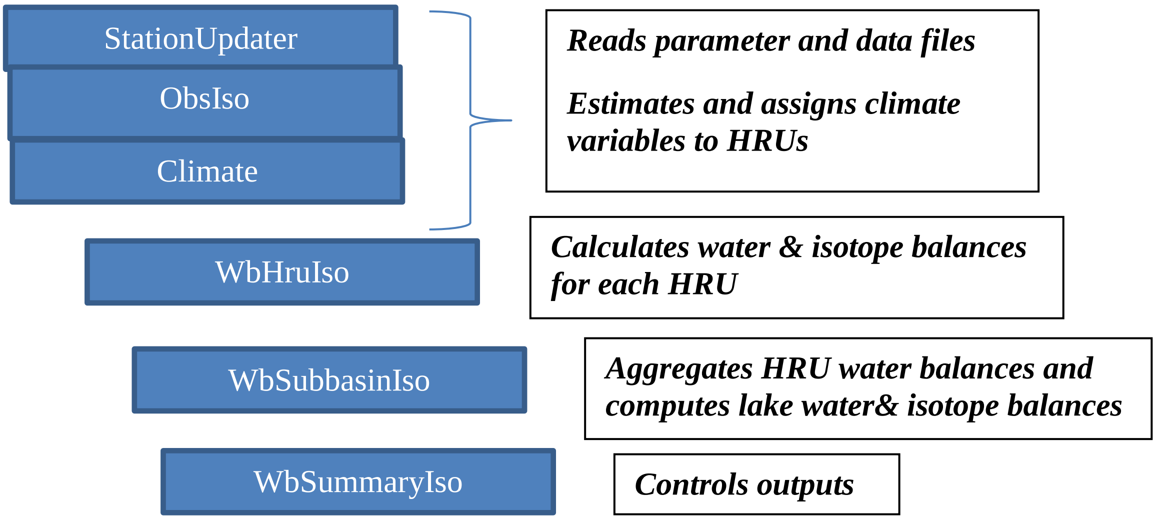
Figure 2. The HRU and subbasin delineation of the Lake Tana basin.

2.2 Model Design Concepts

The IWBMIso model has been developed using the United States Department of Agriculture (USDA) Object Modeling System (OMS) (David et al., 2010). OMS is a Java-based modular modeling framework for model development, testing, and deployment. It consists of a library of stand-alone science, control, and database components (modules) and a means to assemble selected components into a modeling package that is customized to the problem, data constraints, and scale of application. The framework is supported by utility modules that provide a variety of data management, land unit delineation and parameterization, sensitivity analysis, calibration, statistical analysis, and visualization capabilities. OMS uses an open source software approach to enable all members of the scientific community to collaboratively work on addressing the many complex issues associated with the design, development, and application of distributed hydrological and environmental models. A detailed description of the OMS framework is given in the ‘Object Modeling System v3.0 Developer and User Handbook’ (David et al., 2010) that can be accessed from http://javaforge.com/project/oms.

The IWBMIso model is composed of six components. In OMS, a component is a modeling entity that implements one modeling concept and is implemented as a plain (Java) object that comes along with annotations. A component can be hierarchical; it may contain other, finer grained components contributing to the larger goal.

The six components of the IWBMIso are:



Reads parameter and data files

Estimates and assigns climate variables to HRUs

Calculates water & isotope balances for each HRU

Aggregates HRU water balances and computes lake water& isotope balances

Controls outputs

***StationUpdater* - ** The StationUpdater component controls the data retrieval process by reading the names and locations of the various input files whose details are provided by the user in the OMS simulation (.sim) file. The input variables for this model are assumed to come largely from regional and global gridded climate and isotope datasets. However the use of point data measurements is also supported. The gridded and point data sets can be used in a set of data pre-processing tools to estimate the value of data variables for each HRU.

Monthly mean values of precipitation, maximum and minimum air temperature, potential evapotranspiration, and relative humidity are computed for each HRU for each time step of the model simulation period. However, the availability of measured values for δH^{2}, δO^{18}, and Normalized Difference Vegetation Index (NDVI) are typically for much shorter periods of time. To enable the use of these variables for longer time periods, a mean monthly value is computed for each HRU using the available period of data, and these mean monthly values are then used for the entire period of a model simulation.

The model input variable file names are provided in the OMS .sim file. An annotated example of the .sim input files names is

pptFileName "pre.csv" (monthly precipitation, mm)

tminFileName "tmn.csv" (monthly mean maximum air temperature, deg C)

tmaxFileName "tmx.csv" (monthly mean minimum air temperature, deg C)

petFileName "pet.csv" (monthly potential evapotranspiration, mm)

ndviFileName "ndvi.csv" (monthly Normalized Difference Vegetation Index (NDVI))

obsFileName "obs.csv" (measured variables used for model performance evaluation)

isoH2FileName "dh2p.csv" (monthly precipitation, δH^{2})

isoO18FileName "do18p.csv" (monthly precipitation, δO^{18})

rhnFileName "rhn.csv"   (monthly mean normalized relative humidity)

ObsIso - The ObsIso component reads the input data files using the file names and directory locations provided by the StationUpdater component.

Climate - The Climate component distributes all the input variables read by ObsIso to their associated computational variable names and array locations. The contents of an input data file can be the field measurements for each measurement station associated with a basin, or they can be the results of a user-selected spatial distribution procedure that calculates the distribution of the measured values to each HRU. Selected model parameters identify the number of stations of data in each data file. A set of data pre-processing tools that handle the spatial and temporal distribution of selected input variables for each HRU and time step are provided in the modeling framework.

WbHrusIso - The WbHruIso component computes the water balance and isotope balance elements on each HRU using the inputs from the StationUpdater, ObsIso and Climate components, and the initial conditions defined for each HRU in the parameter and simulation files. The water and isotope balance elements are presented schematically in Figure 1.

WbSubbasinIso - The WbSubbasinIso component aggregates all the HRU components to the subbasin level, computes lake water and isotope mass balance components, and routes water and isotopes through the network of subbasins to the last subbasin outflow point. If there is a lake, its bathymetric relationships are taken care of here based on polynomial coefficients that are provided in the parameter file.

WbSummaryIso – The WbSummaryIso component computes summary water balance values for the complete watershed, watershed subbasins and lakes and provides these values in tabular output form.

3.0 IWBMIso

The conceptualized watershed system and its inputs, outputs and component fluxes are shown schematically in Figure 1. Descriptions of the components that comprise the hydrologic and isotopic processes in IWBMIso are described below.

3.1 Land Surface Water Balance Processes

Monthly precipitation and monthly mean maximum and minimum air temperature are the input climate variables. The monthly mean of the maximum and minimum air temperature values is used to determine the form of precipitation. Precipitation occurring as snow will form a snowpack that is then melted as a function of monthly mean air temperature and a melt coefficient.

Monthly mean air temperature for each HRU (hru_temp_i) is computed by

$$\text{hru\_temp}_i = (\text{tmax}_{tsta} + \text{tmin}_{tsta}) * .5 + \text{temp\_adjust}_{month,k} \quad (1)$$

where tmax_{tsta} and tmin_{tsta} are the monthly mean maximum and minimum air temperature values from ObsIso for each temperature station *tsta* and *temp_adjust* is an adjustment parameter with values defined for each month and subbasin *k*. The value for *tsta* is obtained from the model parameter *hru_tsta* which defines the *tsta* associated with an HRU. The value of *k* is obtained from the model parameter *hru_subbasin* which defines the subbasin in which an HRU resides.

The model parameter *ntemp* defines the number of temperature stations, or columns, in the temperature input files read by ObsIso. When temperature input for each HRU has been estimated using pre-processing tools, then *ntemp* will equal *nhru*, the parameter that defines number of HRUs. In this case, *tsta* then has the same value as i. If *ntemp* is not equal to *nhru*, then the user has the option to assign *hru_tsta* to an HRU based on local or regional information and adjust these station values to each HRU in the model using *temp_adjust*.

Monthly mean precipitation for each HRU i (hru_ppt_i) is computed by

$$\text{hru\_ppt}_i = \text{precip}_{psta} * \textit{ppt\_adjust}_{\text{month,k}} \qquad (2)$$

where precip_{psta} is the input precipitation value from ObsIso for the precipitation station *psta* associated with HRU i and $\textit{ppt\_adjust}_{\text{month,k}}$ is an adjustment parameter with values defined for each month *i* and each subbasin *k*. The value for *psta* is obtained from the model parameter *hru_psta* which defines the *psta* associated with an HRU. The value of *k* is obtained from the model parameter *hru_subbasin* which defines the subbasin in which an HRU resides.

The model parameter *nrain* defines the number of precipitation stations, or columns, in the precipitation input file read by ObsIso. When precipitation input for each HRU has been estimated using gridded data fields, then *nrain* will equal *nhru*, the parameter that defines the number of HRUs. In this case, psta then has the same value as i. If *nrain* is not equal to *nhru*, then the user has the option to assign *hru_psta* to an HRU based on local or regional information and adjust these station values to an HRU using *ppt_adjust*.

The form of input precipitation is determined using temperature threshold parameters. The parameter *train* is a threshold temperature above which all precipitation is assumed to be rain. The parameter *tsnow* is a threshold temperature below which all precipitation is assumed to be all snow. Given a monthly mean air temperature hru_temp and precipitation depth hru_ppt, rain and snow amounts are computed by

hru_temp >= *train* Precipitation is all rain (3)

hru_temp <= *tsnow* Precipitation is all snow (4)

tsnow < hru_temp < *train* then

$$\text{psnow}_i = \text{hru\_ppt}_i * ((\textit{train} - \text{hru\_temp}_i) / (\textit{train} - \textit{tsnow})) \qquad (5)$$

where psnow_i is monthly snow fall and prain_i is then $\text{hru\_ppt}_i - \text{psnow}_i$.

The value of psnow is used to initiate a snowpack (snstor) or is added to an existing snowpack.

Precipitation that falls on a vegetated cover is intercepted by the canopy. The amount of interception is limited by the maximum interception capacity of the canopy which is defined by the parameter *intercept_pct*. The remaining amount forms the net rainfall, or throughfall, that is available for surface runoff and infiltration. The consideration of the interception process at a monthly time step may not represent a significant process for some basins. The user has the option of turning this process on or off using the parameter *interceptionOn.*

Surface runoff (rodirect) is computed as a percentage of the monthly net rainfall, independent of the storage in the soil reservoir by

$$\text{rodirect}_i = \text{prain}_i * \textit{directfac}_i \qquad (6)$$

where $\textit{directfac}_i$ is the HRU parameter that defines the percentage of prain_i that becomes rodirect on HRU i. The variable prain_i is then updated to remove rodirect_i and the remainder is available for infiltration.

Another source of water for infiltration is snowmelt. If a snowpack exists, then hru_temp is used to compute snowmelt. Snowmelt (snroff) is computed when hru_temp is greater than *tsnow* by

$$\text{snroff}_i = \textit{xmeltcoeff}_i * (\text{hru\_temp}_i - \textit{tsnow})\text{*DAYS[month]} \qquad (7)$$

where $\textit{xmeltcoeff}_i$ is a parameter that expresses a daily snowmelt rate and DAYS[month] is the number of days in the month. If snroff_i is greater than the existing snowpack snstor_i, then snroff_i is set equal to the snstor_i. It is possible to have snowmelt and rain occur in the same monthly time step. In this case snroff is added to prain to compute the water available for infiltration.

Infiltrated water is used to fill the water holding capacity of the soil reservoir. The maximum storage capacity of the soil reservoir is defined by the parameter *whc*. Water in storage in the soil reservoir is available for loss by evapotranspiration. Computation of the amount of actual evapotranspiration (aet) requires as the estimation of the potential evapotranspiration (pe). There are two methods available for the estimation of pe. Method selection is accomplished using the parameter *pet_method*.

Setting *pet_method* to 0 selects the option for the user to compute these values using a user-selected pre-processing method that computes pe for each HRU and each time step. The resulting data file is then input to the model using the StationUpdater and ObsIso components.

Setting pet_method to 1 selects the method that computes pe using the Hamon equation (Hamon, 1961). The value of pe_i is computed by

$$\text{pe}_i = \text{hamon\_c}_{\text{mo},j} * (\text{DL} / 12)^2 * \text{PT} * \text{DAYS}_{\text{mo}} \qquad (8)$$

$$\text{PT} = (4.95 * \exp(0.062 * \text{hru\_temp}_i)) / 100. \qquad (9)$$

where $\text{hamon\_c}_{\text{mo,j}}$ is the hamon coefficient, varied by month *mo* and by subbasin *j*, DL is the mean monthly hours of daylight per day, PT is a saturated water vapor density term computed using Equation 9, DAYS_{mo} is the number of days in the month mo, and hru_temp_i is monthly mean air temperature on HRU i. Computed values of pe_i less than zero are set to zero.

The actual amount of evapotranspiration aet can be computed by one of two methods. Method selection is accomplished using the parameter *aet_method*.

Setting *aet_method* to 0 selects a simple function that uses pe and the amount of water available in the soil reservoir to compute aet. The method first computes the variable $pmpe_i$ by

$$pmpe_i = (prain_i + snroff_i) – pe_i \qquad (10)$$

If $pmpe_i$ is positive, then aet_i is equal to pe_i and the $pmpe_i$ is the amount of water excess that moves into soil moisture storage. If $pmpe_i$ is less than zero, then the amount of moisture that can be removed from the soil is less than pe_i. The amount of water taken from the soil ($loss_i$) is computed as a linear function of the amount of water available in the soil (sm_stor_i) by

$$loss_i = abs (pmpe_i * (sm\_stor_i / whc_i)) \qquad (11)$$

The total aet_i is then computed as the sum of $prain_i$, $snroff_i$, and $loss_i$.

Setting *aet_method* to 1 selects the VegET method (Senay, 2008) which incorporates spatial and temporal variations in vegetation conditions. Changes in vegetation conditions are estimated using globally available MODIS composite Normalized Difference Vegetation Index (NDVI) data (MODIS, xxx).

The basic VegET model equation for estimating aet is:

$$aet_i = Ks * Kcp * pe_i \qquad (12)$$

$$Kcp = 1.25 * NDVI_i + 0.2 \qquad (13)$$

$$Ks = sm\_stor_i / MAD \qquad (\text{if } sm\_stor_i < MAD) \qquad (14)$$

$$Ks = 1.0 \qquad (\text{if } sm\_stor_i >= MAD)$$

where, Ks is the soil stress coefficient and Kcp is the land surface phenology (LSP) water-use coefficient that is estimated using the NDVI of the HRU.

Ks is determined using sm_stor_i and the Maximum Allowable Depletion (MAD) level of soil water in the soil zone. MAD is the level below which evapotranspiration is less than “potential” and is constrained by the availability of soil water according to Equations 14. Although MAD varies by crop/vegetation type, a nominal value of 50% of the soil whc_i can be used for most generalized crops such as cereals and natural vegetation. Thus, MAD is estimated as 50% of the whc_i. The dimensionless Ks coefficient varies from 0.0 to 1.0 depending sm_stor_i

The amount of aet is removed from sm_stor and the resulting sm_stor value is compared to *whc*. If sm_stor is greater than *whc*, sm_stor is set equal to *whc* and the amount in excess of *whc* is available for movement to the groundwater and subsurface reservoirs. The soil to ground water flux is assumed to have a maximum monthly value that is defined by the parameter *soil2gw_max*. The soil water excess is compared to the parameter *soil2gw_max*. If the excess is greater than *soil2gw_max*, then *soil2gw_max* is added to the groundwater storage reservoir and the remaining water is added to the subsurface reservoir. If the excess is less than *soil2gw_max*, then all the excess is added to the groundwater storage reservoir.

Subsurface flow (ssflow) is computed as a linear function of the volume of water stored in the subsurface storage reservoir (ssres_stor) by the equation

$$\text{ssflow}_i = \textit{ssflow\_coef}_i * \text{ssres\_stor}_i \qquad (15)$$

where $\textit{ssflow\_coef}_i$ is a parameter defining the depletion rate. The ssres_stor_i is then depleted by the amount of ssflow_i.

Groundwater flow (ssflow) is computed as a linear function of the volume of water stored in the groundwater storage reservoir (gwres_stor) by the equation

$$\text{gwflow}_i = \textit{gwflow\_coef}_i * \text{gwres\_stor}_i \qquad (16)$$

where $\textit{gwflow\_coef}_i$ is a parameter defining the depletion rate (baseflow recession coefficient). The gwres_stor_i is then depleted by the amount of gwflow_i.

In addition to gwflow from the groundwater reservoir, direct groundwater routing (gwflow_direct) between HRUs is possible. This is achieved by using a parameter known as *gwDirect_coef*, a linear coefficient similar to *gwflow_coef*.

$$\text{gwflow\_direct} = \text{gwDirect\_coef} * \text{gwres\_stor} \qquad (17)$$

The amount of gwflow_direct will further deplete gwres_stor. The use of direct groundwater routing also allows for lake-groundwater interaction to be considered because a lake is also considered to be an HRU. It also enables the movement of groundwater into or out of the basin. The parameter *gwDirect_coef* is used to estimate the potential amount of groundwater (Eq. 17) that could be lost from one HRU to one or more HRU(s) downstream given that it has a recipient HRU or an external basin.

The groundwater routing between HRUs is handled in the model through the following mechanisms:

- If an HRU, including a lake, loses groundwater to a downstream HRU, the HRU parameter *up_gwres* is set to the downstream HRU index with negative sign.

- If an HRU loses to more than one HRU, each receiving HRU gets the index of the upstream HRU with negative sign, and the percentage of the total to each HRU is defined with HRU parameter *up_gw_pct*.

- If an HRU, including a lake, gains groundwater from an upstream HRU, *up_gwres* is set to the upstream HRU index with a positive sign. If more than one HRU contributes to a single HRU such as a lake, each of them takes the index of the receiving HRU.

- If an HRU loses groundwater to adjacent basin (watershed), *up_gwres* is set to -9999.

- If HRU gains groundwater from adjacent basin (watershed), *up_gwres* is set to 9999 and the amount in mm per month is specified under the HRU parameter *gw_extern* with the respective isotope values of the addition provided using the HRU parameters *do18_gw_extern*, and *dh2_gw_extern*. For now we can only handle a constant amount of input with constant isotope values throughout from external watersheds.

Finally, the sum of direct runoff (rodirect), subsurface flow (ssflow), and groundwater flow (gwflow) constitutes the total flow (ro) from a given HRU.

The total flows from all HRUs within a subbasin are summed using a unit area weighting approach to compute the subbasin flow sub_mcm which is the flow in the unit of millions of cubic meters. The sub_mcm value from each subbasin is routed to its downstream subbasin and combined with the sub_mcm of that subbasin. The routed total flow from the last downstream subbasin (basin_totflow_mm) is the total flow for the entire basin expressed in mm. Water bodies on the channel network are treated as subbasins and so are parts of the routing sequence.

3.2 Lake Water Balance

The lake water balance equation can be written as

Storage change = inflow + precipitation – outflow – evaporation

or

$$\frac{\&dV_L}{dt} = I_S + I_G + P - O_S - O_G - E \qquad (18)$$

where V_L is the lake volume, I_S, I_G, O_S and O_G represent the volumetric surface and groundwater inflow and outflow fluxes, respectively, P is precipitation over the lake and E is the evaporation flux from the lake. The surface inflow I_S is the sum of the outflows flows from the subbasins that are adjacent, and flow into, the lake. The groundwater inflow I_G is the sum of the gwflow_direct flows from HRUs that are adjacent to the lake and have been parameterized to compute lake inflows. The O_G is the groundwater outflow from the lake to one or more adjacent HRUs that have been specified using the gwflow_direct methodology.

The lake bathymetric relationships are computed assuming that a third degree polynomial equation describes the lake level – area and lake level – volume relationships. Lake surface area (lakeArea) is computed by

$$\text{lakeArea} = \textit{lake\_L\_A\_c3}*\text{lakesurf\_elev}^3 + \textit{lake\_L\_A\_c2}*\text{lakesurf\_elev}^2 + \textit{lake\_L\_A\_c1}*\text{lakesurf\_elev} + \textit{lake\_L\_A\_c0} \qquad (19)$$

where lakesurf_elev is the lake surface elevation in m and the coefficients *lake_L_A_c3, lake_L_A_c2, lake_L_A_c1*, and *lake_L_A_c0* are model parameters.

The lake volume is computed by

$$\text{lakeVol} = \textit{lake\_L\_V\_c3}*\text{lakesurf\_elev}^3 + \textit{lake\_L\_V\_c2}*\text{lakesurf\_elev}^2 + \textit{lake\_L\_V\_c1}*\text{lakesurf\_elev} + \textit{lake\_L\_V\_c0} \qquad (20)$$

where lakesurf_elev is the lake surface elevation in m and the coefficients *lake_L_V_c3, lake_L_V_c2, lake_L_V_c1*, and *lake_L_V_c0* are model parameters.

The lake outflow is assumed to be a broad crested weir. The outflow from the lake in cubic meters per second (lake_out_cms) is computed as a function of the difference between lakesurf_elev and the parameter *lake_elev_outflow* which defines the elevation of the top of the outflow weir. The difference is termed the head. Lake_out_cms_i for lake i is computed by

$$\text{lake\_out\_cms}_i = \mathit{rout\_coef\_alpha}_i * (\text{head}_i - \mathit{rout\_coef\_b}_i)^{(\mathit{rout\_coef\_mi} + \mathit{rout\_coef\_ci})} \quad (21)$$

where the coefficients $\mathit{rout\_coef\_alpha}_i$, $\mathit{rout\_coef\_b}_i$, $\mathit{rout\_coef\_m}_i$, and $\mathit{rout\_coef\_c}_i$ are model paremeters for each lake i.

The lake outflow lake_out_cms_i is routed to the downstream subbasin identified in the parameter file using the parameter *subbasin_down.*

3.3 Isotope Balance Elements

3.3.1 Isotope Mixing and Fractionation

Once water reaches the ground surface, there are two major processes that would affect its stable isotope values: mixing and fractionation. During evaporation in a natural environment, the water vapor released from a liquid surface is depleted in heavy isotopes with respect to the liquid, because the isotopic species of water $^1H^2H^{16}O$ and $^1H_2{}^{18}O$ are generally less volatile than the lighter, most abundant one, $^1H_2{}^{16}O$ (Gonfiantini, 1986). The process of fractionation is considered for evaporation from canopy interception, soil reservoir and water bodies. The isotopic mass balance is mainly driven by the isotopic composition of precipitation (do18_ppt, dh2_ppt) which is an input variable through data files. In the following discussion, the use of the δ symbol in all equations is assumed to apply to the computations for both δO^{18} and δH^2 composition.

The δO^{18} and δH^2 composition of direct runoff (do18_rodirect, dh2_rodirect) is the same as that of the precipitation for that time step. The soil storage reservoir is assumed to be a well-mixed reactor and so the isotope composition (do18_sm, dh2_sm) for time step i is computed using linear mixing between the antecedent soil moisture and the new precipitation (Eq. 22.)

$$\delta sm_{\,i} = ((\delta precip_i * precip_i) + (\delta sm_{i-1} * sm\_stor_{i-1})) / (precip_i + sm\_stor_{i-1}) \quad (22)$$

The new composition value δsm is also applied to all water fluxes from the soil to the groundwater and subsurface reservoirs when soil moisture storage (sm_stor) exceeds soil storage water holding capacity (*whc*). For the groundwater recharge (infiltration) processes, the proportion of a quicker preferential flow with respect to a slower piston type flow can be specified by the modeler.

The new sm_stor value is also available for evaporation and transpiration losses. The evaporation process is one that results in fractionation of the isotope composition.

That is, during evaporation the heavier isotope molecules do not evaporate as readily as the lighter isotope molecules and so there is an enrichment of the heavier molecules in the reaction. Fractionation is assumed to affect only evaporation and not the transpiration process. Thus, the total actual evapotranspiraton loss (aet) must be divided into the two components of evaporation and transpiration, not a trivial process. An approximation of the evapotranspiraton lost to transpiration is made using vegetation cover density (covden) as the computational factor. The fractionating evaporation (Ef) can be estimated by

$$Ef = aet_i \times (1. - covden). \qquad (23)$$

$$f = (sm\_stor - Ef) / sm\_stor \qquad (24)$$

Given that f is the residual fraction of the total evaporating water body, that is what is left after evaporation, we can estimate f using equation 24.

Fractionation from evaporation of soil moisture is calculated using the fractionation equation proposed by Gonfiantini 1986 for a drying up water body (Eq. 25).

$$\delta = (\delta_o - A/B)\, f^{B} + A/B \qquad (25)$$

where δ_o is the initial isotopic composition of the water, f is the fraction of the remaining moisture and A and B are defined in Eqs. 26 and 27. The result of the linear mixing, δsm_i in equation 22, is used as the initial isotope composition of the soil moisture storage ($\delta_{o)}$ in equation 25.

$$A = (h\delta_a + \Delta\varepsilon + \varepsilon/\alpha) / (1 - h + \Delta\varepsilon) \qquad (26)$$

$$B = (h_{-} \Delta\varepsilon - \varepsilon/\alpha) / (1 - h + \Delta\varepsilon) \qquad (27)$$

Note: in equations 26 and 27 the terms $\delta_{a,}$ $\Delta\varepsilon$, ε must be expressed as ratios and not as per mil for computational efficiency (Gonfiantini, pers. Comm.).

where: δ_a is the isotopic composition of the atmospheric vapor in the 'free atmosphere' region above the surface. It can be measured or estimated from δ_{ppt} (isotope composition of precipitation) using the equation 28 (Craig and Gordon, 1965).

$$\delta_a = \delta_{ppt} / \alpha - (1 - 1/\alpha) \qquad (28)$$

The equilibrium enrichment factor, ε is expressed using Eq. 29.

$$\varepsilon = (\alpha - 1) \qquad (29)$$

α is the equilibrium fractionation factor. α can be computed as a function of temperature (T) by using the equations proposed by Majoube, 1971 (Eq. 30).

$$\alpha_{l-v} = e^{(C2/T2 + C1/T + C0)} \qquad (30)$$

For O18: C0 = -2.0667e-03, C1 = -.4156, C2 = 1.1371e+03 and for H2: C0 = 52.612e-03, C1 = -76.248, C2 = 24.844e+03; where T is the temperature in degrees Kelvin and the values for the constants C vary by isotope.

The kinetic enrichment factor, Δε is proportional to the moisture deficit (1 - h) of the atmosphere and to the relative difference of transport resistance in air between isotopic molecules (ρi/ρ - 1) (Eq. 31; Gonfiantini, 1986). The diffusion-controlled isotope fractionation of water is significant and plays a major role in determining its isotopic compositions. These kinetic isotope effects result from differences in the transport property of various isotopic water molecules ($H_2{}^{16}O$, $^{1}H_2H^{16}O$, $H_2{}^{17}O$, and $H_2{}^{18}O$) through the diffusion-controlled sub-layer of air and water (Horita et al., 2008).

$$\Delta\varepsilon = (1 - h) \cdot (\rho i/\rho - 1) \qquad (31)$$

or

$$\Delta\varepsilon = (1 - h) \cdot \theta \cdot Ck$$

$$Ck = ((D/Di)\text{^}n - 1) \qquad (32)$$

$$\theta = (\rho M/\rho) \qquad (33)$$

D and Di are the molecular diffusion coefficients of the common and heavy isotope species, respectively, and n is a turbulence parameter such that n = 1/2 for an average turbulent flow, n = 2/3 for a laminar flow, and n = 1 for static transport (Gat, 1996). n=1 fits the data well for evaporation of water through a stagnant air layer, such as from the soils (Barnes and Allison, 1988; Allison et al., 1955 in Gat, 1996; Horita et al., 2008). *θ,* an advection term factor which accounts for humidity build-up over a water surface such as a lake, can be assumed to be equal to 1 for a small water body whose evaporation flux does not perturb the ambient moisture significantly (Kebede et al., 2008). However, it was shown to be 0.88 in the case of the North American Great Lakes (Gat et al., 1994) and that a value of *θ* = 0.5 applies in the eastern Mediterranean Sea (Gat et al., 1996). The latter value appears to be the limiting factor encountered also over large open oceanic areas (Gat, 2010). It can generally be assumed to be zero for evaporation from soil.

According to Gonfiantini, 1986 the values of Δε obtained using Eqns. 34 and 35 are adequate for the evaporation conditions mostly occurring in nature:

$$\Delta\varepsilon^{18}O‰ = 14.2\ (1 – h) \Rightarrow \Delta\varepsilon^{18}O = 0.0142\ (1 – h) \qquad (34)$$

$$\Delta\varepsilon^{2}H‰ = 12.5\ (1 – h) \Rightarrow \Delta\varepsilon^{2}H = 0.0125\ (1 – h) \qquad (35)$$

The soil to subsurface reservoir flux is added to the subsurface reservoir and the isotope composition of the subsurface reservoir ($\delta ssres_i$) is then computed by equation 36.

$$\delta ssres_i = ((\delta sm2ssr * sm2ssr_i) + (\delta ssres_{i-1} * ssres\_stor_{i-1})) /$$

$$(sm2ssr_i + ssres\_stor_{i-1}) \qquad (36)$$

Any subsurface flow that occurs in this time step will have the isotope composition $\delta ssres_i$.

The soil to groundwater reservoir flux carries the isotope composition of either $\delta precip_i$ or δsm_i or a mix of both. This mix is controlled by the type of soil moisture excess flow to the groundwater reservoir (i.e fast preferential flow or a slower piston flow). This is controlled by a user-defined parameter *hru_rech_prefer* that describes the proportion of preferential flow during recharge (0 means no preferential flow while 1 indicates 100% preferential flow). The δgwres is initialized with the parameters do18_gwres and dh2_gwres which can vary by HRU. It is then computed through the mixing model of equation 37.

$$\delta gwres_{i} = ((\delta gwres_{i-1} * gw\_stor_{i-1}) + (surplus_{i} * \delta\_sm2gw_{i})) / (gw\_stor_{i-1} + surplus_{i}) \quad (37)$$

Where surplus indicates the soil moisture that exceeds storage water holding capacity and that is available for groundwater recharge.

Since the total flow from an HRU (ro) is computed by: ro = ssflow + gwflow + redirect the isotope composition of the total flow is computed by equation 38.

$$\delta ro = ((ssflow * \delta ssflow) + (gwflow * \delta gwflow) + (rodirect * \delta rodirect)) / ro \quad (38)$$

where the isotopic composition of each of the component flows is equal to that of their respective reservoir.

Equation 24 is also used for fractionation during evaporation from canopy interception after mixing between the precipitation water and the remaining intercepted water in the canopy. The throughfall (net rainfall) under the canopy will have the isotopic composition of the mixed intercepted water just before evaporation.

3.3.2 Lakes and Reservoirs

One additional computation of isotopic composition is related the surface water layer in lakes and reservoirs. A lake receives inflow from upstream areas and discharge flow to downstream areas within a watershed. The inflow volume and isotopic composition is computed for land areas above the water body using the equations discussed above. The outflow from the lake is computed as a function of the lake level using a rating curve. At this time, only natural water bodies are included in the model, but managed water bodies can be added at a later date.

The general water balance equation for a water body as shown in Eq. 18 and repeated here, is based on the mass conservation law, as in Eq. 39:

$$\frac{dV_L}{dt} = I_S + I_G + P - O_S - O_G - E \quad (39)$$

$$\text{Or} \quad \frac{dV_L}{dt} = I - O - E$$

where Inflow (I) is the surface(I_S), and groundwater flow (I_G) rates into the lake, Outflow (O) is the surface-water outflow (O_S) and groundwater (O_G) rates, P is the precipitation rate, E is the evaporation rate, V is the volume of water in storage, and t is time (Krabbenhoft et al., 1990).

In general, lakes and reservoirs are stratified systems. The stratification is mainly controlled by water temperature. Hostetler and Bartlein (1990) developed a model to simulate the seasonal variation of lake temperature and evaporation that eventually served as the basis for a predictive model of stable isotopes (Hostetler and Benson,

1994) and a hydrologic-isotopic balance model-HIBAL (Benson and Paillet, 2002). We have adopted the same approach of a stratified system with two layers separated by a thermocline. Partial mixing happens between portions of the two layers during the shift of the boundary depth due to water balance while a complete mixing (lake turnover) between the two layers is applied during user defined months.

The isotope mass balance equation for lakes and wetlands can be written as in Eq. 40.

$$\delta_L \frac{dV_L}{dt} + V_L \frac{d\delta}{dt} = \delta_{IS} I_S + \delta_{IG} I_G + \delta_P P - \delta_{OS} O_S - \delta_{OG} O_G - \delta_E E \qquad (40)$$

where the delta notation δ followed by a subscript represents the isotopic composition of each component defined above. Equation 36 can further be regrouped into Eq 41 which can then be used for estimating the isotopic concentration of the lake at a given month where *i* based on its initial value at month *i-1*.

$$\delta_i^L = \frac{\mathrm{V}_L \delta_{i-1}^L + (\delta_{IS} I_S + \delta_{IG} I_G + \delta_P P - \delta_{OS} O_S - \delta_{OG} O_G - \delta_E E)}{(\mathrm{V}_L + I_S + I_G + P - O_S - O_G - E)} \qquad (41)$$

By integrating Eq. 40 by using the mean values of individual parameters over the integration period, Gonfiantini (1986) derived Eq. 42, which should be a very close approximation to Eq 41 for a well-mixed lake the volume variations of which are not negligible.

$$\delta = \left(\delta_o \; - \frac{\delta_I + A*x}{1+B*x}\right) * f^{-\frac{1+B*x}{1-x-y}} + \frac{\delta_I + A*x}{1+B*x} \qquad (42)$$

Where x = E/I (fraction of inflow water lost by evaporation), y=Q/I (fraction lost by isotopically non-fractionating outflows), A and B are as defined in Eqs. 26 and 27 respectively.

Gonfiantini (1986) noted that Eq. 42 reduces to Eq. 25 for a drying up water body without inflow and outflow. In the case where the water body desiccates slowly and gradually, such as in isolated flood ponds and wetlands that dry out slowly through evaporation and infiltration, Eq. 42 reduces to Eq. 43 where z = E/(E+Q).

$$\delta = \left(\delta_o - \frac{A}{B}\right) * \; f^{B*z} \; + \left(\frac{A}{B}\right) \qquad (43)$$

The isotopic enrichment of an evaporating water mass (which has a composition of δ_L) is a non-equilibrium process, controlled by isotopic exchanges between liquid and vapor water phases, and the composition of the evaporated moisture (δ_E) depends on that of the ambient air moisture (δ_a).

The Craig-Gordon linear resistance model has formulated this process, by considering a molecular diffusion layer above the water surface bounded below by a saturated layer and above by a turbulent transportation layer (Eqs. 44).

$$\delta E \;= [\delta L/\alpha - h \,.\, \delta a - (\varepsilon^* + \Delta\varepsilon)] \,/\, [(1 - h) + \Delta\varepsilon/1000] \qquad (44)$$

where $\varepsilon^* = (1 - \alpha^*) \cdot 1000$ (where $\alpha^* = 1/\alpha$)

Note that δ, Δε and ε values in equations 41 are in per mil.

The relative humidity, h, is measured near the lake and does not reflect the relative humidity just over the lake surface. It is therefore normalized to the saturation vapour pressure at the temperature of the water surface. Specific humidity (g/m3) under saturation conditions can be estimated using the Magnus formula (Moeller, 1973):

$$e_s = \frac{1321.7 * e^{\frac{17.27*t}{237+t}}}{(273+t)} \qquad (45)$$

where e_s is the saturation vapor pressure and t is the temperature in degrees Celsius measured near the lake. To obtain normalized relative humidity, e_s should be calculated for air and water temperature separately by using Equation 45. Then, the relative humidity measured near the lake needs to be multiplied by the ratio of e_s calculated from air temperature to water temperature. The activity of water (aH2O) is close to unity in seawater (0.98), but becomes significantly lower in hypersaline brines (ca. 0.67 in Dead Sea brines). In such cases, the RH should also be normalised to the activity of water (i.e., ħ = h/aH2O) (Horita et al., 2008).

In equations 39 and 40, the terms Inflow and Outflow do include inflows to and outflows from groundwater. However, because of the difficulties in estimating these components they are often neglected in lake water balance computations. Here, we introduced variables termed direct groundwater inflow and groundwater outflow, to distinguish them from baseflow components of rivers, by making use of the isotope mass balance to constrain these values and hence to better estimate them.

4.0 References

Alley, W.M., 1985. Water balance models in one-month-ahead streamflow forecasting: Water Resources Research, v. 21, p. 597–606.

Barnes,C.J. and Allison, G.B. 1988. Tracing of water movement in the unsaturated zone using stable isotopes of Hydrogen and Oxygen. J. Hydrology 100:143-176.

Benson, L. and F. Paillet, 2002. HIBAL: a hydrologic-isotopic-balance model for application to paleolake systems. Quaternary Science Reviews, Vol. 21, Issue 12-13, July 2002, pp. 1521-1539.

Craig, H. & Gordon, L. I. 1965. Deuterium and oxygen 18 variations in the ocean and the marine atmosphere. In: Stable Isotopes in Oceanographic Studies and Paleotemperatures (ed. E. Tongiorgi), 9–130, Laboratorio di Geologia Nucleare, Pisa, Italy.

David, O., Ascough, J.C. II, Leavesley, G.H., Ahuja, L., 2010. Rethinking Modeling Framework Design: Object Modeling System 3.0. 2010 International Congress on Environmental Modelling and Software, Modelling for Environment's Sake, Fifth Biennial Meeting, Ottawa, Canada.

Duan, Q., Sorooshian, S., and Gupta, V.K., 1994. Optimal use of the SCE-UA global optimization method for calibrating watershed models: Journal of Hydrology, v. 158, p. 265-284.

FAO/IIASA/ISRIC/ISSCAS/JRC, 2012. *Harmonized World Soil Database (version 1.2)*. FAO, Rome, Italy and IIASA, Laxenburg, Austria.

Fekete, B. M., J. J. Gibson, P. Aggarwal, and C. J. Vorosmarty 2006. Application of isotope tracers in continental scale hydrological modeling,J. Hydrol. Amsterdam, 330, 444–456, doi:10.1016/j.jhydrol. 2006.04.029.

Gat, J. R., C. J. Bowser, and C. Kendall, 1994. The contribution of evaporation from the Great Lakes to the continental atmosphere: Estimate based on stable isotope data. Geophys. Res. Lett., 21, 557–560.

Gat, J. R. 1995. Stable isotopes in fresh and saline lakes, In A. Lerman, D. M. Imboden and J. R. Gat [eds.], Physics and chemistry of lakes. Springer-Verlag, p. 139–165.

Gat, J. R., 1996. Oxygen and hydrogen isotopes in the hydrologic cycle, *Annual Review of Earth and Planetary Sciences*, 24, 225-262.

Gat, J.R. 2010. Isotope Hydrology: A Study of the Water Cycle. Series on Environmental Science & Management, World Scientific Publishing Company Pte Limited.

Gibson, J.J. and Edwards, T.W.D. 2002. Regional surface water balance and evaporation-transpiration partitioning from a stable isotope survey of lakes in northern Canada. Global Biogeochemical Cycles 16: 25-38.

Gonfiantini, R., 1986. Environmental isotopes in lake studies, *Handbook of Environmental Isotope Geochemistry (Volume 3)*, P. Fritz and J. C. Fontes , 113-168, Elsevier, New York.

Hamon, W.R., 1961. Estimating Potential evapotranspiration: Proceedings of the American Society of Civil Engineers, Journal of the Hydraulic Division, v. 87, no. HY3, p. 107-120.

Hay, L.E., Umemoto, M., 2006. Multiple-objective stepwise calibration using Luca: U.S. Geological Survey Open-File Report 2006-1323, 25p.

Horita, J., K. Rozanski & S. Cohen 2008. Isotope effects in the evaporation of water: a status report of the Craig–Gordon model, Isotopes in Environmental and Health Studies, 44:1, 23-49

Horita, J. and Wesolowski, D. J., 1994. Liquid-vapor fractionation of oxygen and hydrogen isotopes of water from the freezing to the critical temperature, *Geochimica et Cosmochimica Acta,* 58, 3425-3437.

Hostetler, S. W., and L. V. Benson. 1994. Stable isotopes of oxygen and hydrogen in the Truckee River-Pyramid Lake surface-water system. 2. A predictive model of d18O and d2H in Pyramid Lake. Limnol. Oceanogr. 39: 356–364.

Hostetler, S.W. and P. J. Bartlein 1990. Simulation of lake evaporation with application to modeling lake level variations of Harney-Malheur Lake, Oregon. WATER RESOURCES RESEARCH, 26, 10, 2603-2612.

IAEA/WMO 2006. Global Network of Isotopes in Precipita-tion. The GNIP Database. Available at: http://isohis.iaea.org.

Jouzel., J. 1986. Isotopes in cloud physics: multiphase and multistage condensation processes. In Handbook of Environmental Isotope Geochemistry, Vol. 2, The Terrestrial Environment, P. Friz and J.C. Fontes (Eds), pp. 61–112, Elsevier, Amsterdam.

Kebede, S., Travi, Y., Rozanski, K. 2009. The δ18O and δ2H enrichment of Ethiopian lakes. J. Hydrol. 365, 173–182.

Kendall, C., and McDonnell, J.J., (eds), 1998. Isotope Tracers in Catchment Hydrology, Elsevier Science Publishers, 839 p.

Krabbenhoft, D.P., Bowser, C.J., Anderson, M.P., and Valley, J.W., 1990. Estimating groundwater exchange with lakes, 1: Use of the stable isotope method: Water Resources Research, v. 26, no. 10, p. 2445-2453.

Leavesley G.H., S.L. Markstrom, R.J. Viger, and L.E. Hay. 2005. USGS Modular Modeling System (MMS): Precipitation-Runoff Modeling System (PRMS) MMSPRMS. In V.P. Singh and D.K. Frevert, eds., Watershed Models, pp. 159–177. CRC Press, Boca Raton, FL.

Legates, D.R., and Mather, J.R., 1992. An evaluation of the average annual global water balance: Geographical Review, v. 82, p. 253–267.

Legates, D.R., and McCabe, G.J., 2005. A re-evaluation of the average annual global water balance: Physical Geography, v. 26, p. 467–479.

Mintz, Y., and Serafini, Y.A., 1992. A global monthly climatology of soil moisture and water balance: Climate Dynamics, v. 8, p. 13–27.

Majoube, M. 1971. Fractionation in O-18 between ice and water vapor. J. Chim. Phys. Phys.Chim. Biol., 68, 625–636.

Markstrom S.L., R.G. Niswonger, R.S. Regan, D.E.Prudic, and P.M. Barlow. 2008. GSFLOW: Coupled ground-water and surface-water flow model based on the integration of the Precipitation-Runoff Modeling System (PRMS) and the Modular GroundWater Flow Model (MODFLOW-2005). U.S. Geological Survey Techniques and Methods 6–D1. [online] URL: http://pubs.usgs.gov/tm/tm6d1/.

McCabe, G.J., and Wolock, D.M., 2008. Joint Variability of Global Runoff and Global Sea-Surface Temperatures: Journal of Hydrometeorology, v. 9, p. 816-824.

McCabe, G.J., and Markstrom, S.L., 2007. A monthly water-balance model driven by a graphical user interface: U.S. Geological Survey Open-File report 2007-1088, 6 p.

NASA Land Processes Distributed Active Archive Center (LP DAAC). MOD13A3, USGS/Earth Resources Observation and Science (EROS) Center, Sioux Falls, South Dakota. 2001. https://lpdaac.usgs.gov/products/modis_products_table/mod13a3

Senay, G. B.; J.P. Verdin; R. Lietzow; A.M. Melesse, 2008. Global daily reference Evapotranspiration modeling and evaluation. Journal of American Water Resources Association, 44, 969-979.

Senay, G.B., 2008. Modeling Landscape Evapotranspiration by Integrating Land Surface Phenology and a Water Balance Algorithm. Algorithms, 1, 52-68; DOI: 10.3390/a1020052

Stadnyk, T. A. , C. Delavau, N. Kouwen and T. W. D. Edwards, 2013. Towards hydrological model calibration and validation: simulation of stable water isotopes using the isoWATFLOOD model. Hydrol. Process. 27, 3791–3810.

Stets, E. G., T. C. Winter, D. O. Rosenberry, and R. G. Striegl. 2010. Quantification of surface water and groundwater flows to open- and closed-basin lakes in a headwaters watershed using a descriptive oxygen stable isotope model, Water Resour. Res., 46, W03515.

Strzepek, K.M., and Yates, D.N., 1997. Climate change impacts on the hydrologic resources of Europe—A simplified continental scale analysis: Climatic Change, v. 36, p. 79–92.

Wolock, D. M., and G. J. McCabe, 1999: Effects of potential climatic change on annual runoff in the conterminous United States. J. Amer. Water Resour. Assoc., 35, 1341–1350.

Xu C-Y, Singh VP. 1998. Dependence of evaporation on meteorological variables at different time-scales and intercomparison of estimation methods. Hydrological Processes 12: 429 – 442.

Yates, D.N., 1996, WatBal—An integrated water-balance model for climate impact assessment of river basin runoff: International Journal of Water Resources Development, v. 12, p. 121–140.

1. Overview of the Environmental Resource Assessment and Management System (eRAMS)

WBMIso is based on the eRAMS technology and platform. eRAMS is an open source modeling and GIS technology. The technology offers services under two broad categories: (i) data and modeling solutions as online services; and (ii) an environment for developers to build collaborative business analytic solutions.

Users can develop a geospatially enabled website on eRAMS. Also, developers can add models and analysis tools as add-on tools using widely used development environments such as HTML, JavaScript, and Python. The eRAMS-GIS tools and approaches are developed using open sources technologies and databases, These technologies include: PostgreSQL as the repository of GIS and other eRAMS data; PostGIS for developing geoprocessing tools, Django as the backend web framework; jQuery for interactions with the web browser; OpenLayers for managing data requests and geocoding, and MapServer for rendering GIS data as images.

All GIS tools and data management services in eRAMS are compatible with commercial software packages such as ESRI ArcGIS and Google Earth data. The system includes a graphical user interface providing user access to modeling services and GIS-enabled tools for various purposes.

eRAMS GIS facilitates access to public data via web services from a single point of access, hence enabling developers to add these data to their tools on eRAMS or other systems. These data include, but are not limited to: climate data from the NOAA, National Climatic Data Center (NCDC), and NRCS SNOTEL; U.S. Census Demographic Profile and Economic data; National Agricultural Statistic Service (NASS) Land Use; USGS National Land Cover Dataset (NLCD); USGS National Water Information System real-time for the Nation; EPA STORET/WQX and WATERS, USGS Hydrography, Transportation, and Government Boundaries.

In addition to GIS capabilities, eRAMS provides services for statistical analysis of location-based time series data, sensitivity/uncertainty analysis and automatic calibration simulation models, optimization of pollution control strategies and green infrastructure, among other tools. The eRAMS technology also includes also a collaboration component that enables creating “Groups”. Users can create a group and enlist members. Members of a group can share their data and projects with the other members with read-only or full-access privileges.

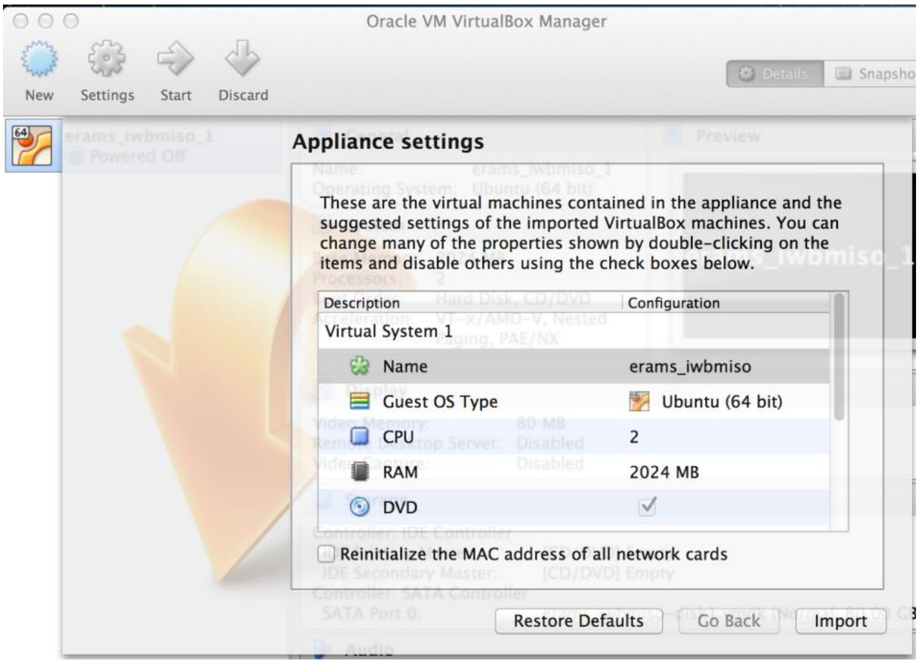
2. Download and Install the IWBMIso Application

For computers with Windows operating system (OS):

- Download the zipfile containing the virtual machine and erams server files from this link: http://www.ids.colostate.edu/downloads/erams_vm.zip. The region files for Lake Tana can be downloaded from http://www.ids.colostate.edu/downloads/erams_vm_data.zip. Extract the zip file(s) to the top of the c: drive. This is needed so that the VM can mount the erams directory at c:\erams_vm\erams.
- Install the VirtualBox software by double-clicking **VirtualBox-4.3.16-95972-Win.exe**. Keep all the default settings.
- Double-click **start.bat** to import the erams virtual machine and start it up. Depending on the computer, it may take several minutes.
- Once the VM has started, a browser window will open to localhost:8081.

For computers with Apple Mac OS (tested with Mountain Lion OS version):

- Download the zipfile containing the virtual machine and erams server files from this link: http://www.ids.colostate.edu/downloads/erams_vm.zip. The region files for Lake Tana can be downloaded from http://www.ids.colostate.edu/downloads/erams_vm_data.zip. Extract the zip file(s) to a folder of choice.
- Download VirtualBox from https://www.virtualbox.org/wiki/Downloads and install to the Applications folder
- Double-click eram_iwbmiso.ova to open VirtualBox, highlight erams_iwbmiso and click Import.



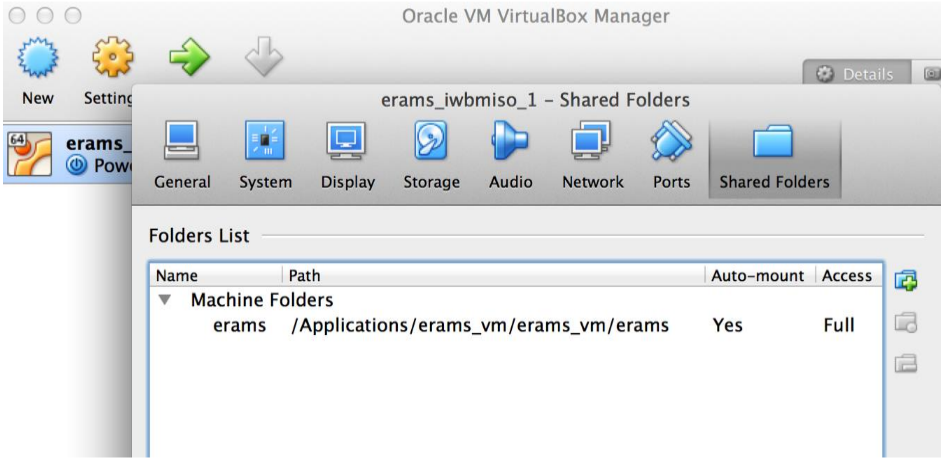
Appliance settings

These are the virtual machines contained in the appliance and the suggested settings of the imported VirtualBox machines. You can change many of the properties shown by double-clicking on the items and disable others using the check boxes below.

Description	Configuration
Virtual System 1	
Name	erams_iwbmiso
Guest OS Type	Ubuntu (64 bit)
CPU	2
RAM	2024 MB
DVD	✓

□ Reinitialize the MAC address of all network cards

- After Import go to Setting and open Shared Folders. Add shared folder path to erams, and click OK to close dialog. With **erams_iwbmiso** VM highlighted click the green arrow to start the VM.



erams_iwbmiso_1 – Shared Folders

Folders List

Name	Path	Auto-mount	Access
▼ Machine Folders			
erams	/Applications/erams_vm/erams_vm/erams	Yes	Full

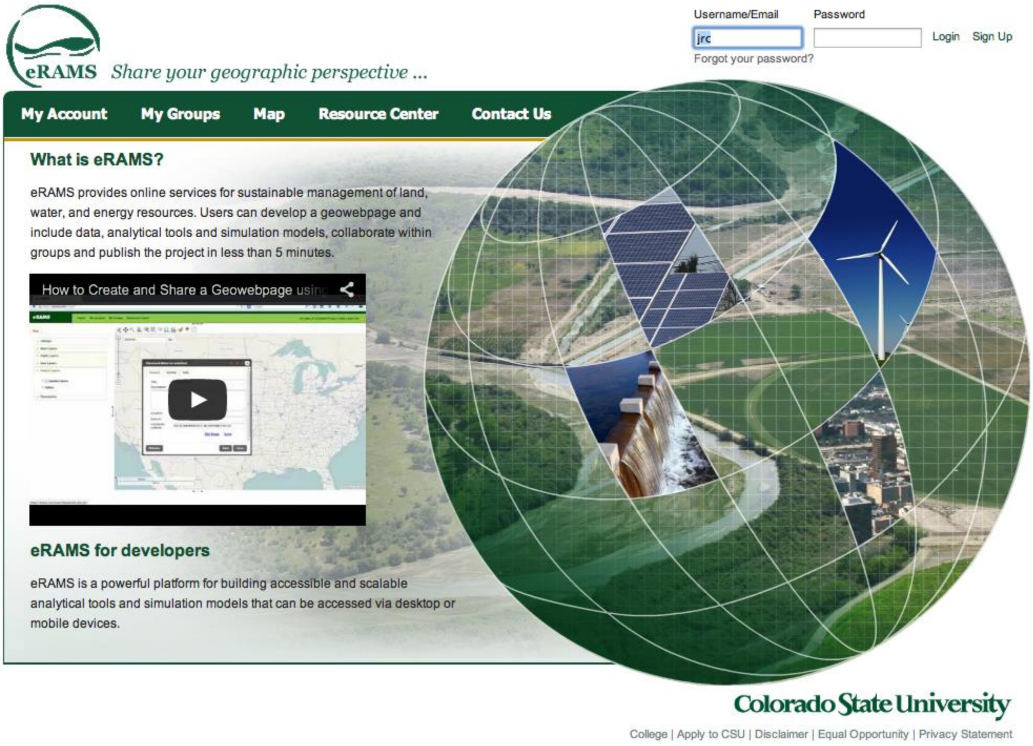
- Once the VM has started, open browser and enter "`localhost:8081`" in the URL field to display the IWBMIso application.

The Data Pre-Processing Tool also can be downloaded from:

http://www.ids.colostate.edu/downloads/WBMIso-InputData.zip

3. Getting Started: Create Login Account

Going to `localhost:8081` the browser should display the following flash page:


eRAMS
Share your geographic perspective ...
My Account
My Groups
Map
Resource Center
Contact Us
Login
Sign Up
Forgot your password?
How to Create and Share a Geowebpage usin
Colorado State University
College | Apply to CSU | Disclaimer | Equal Opportunity | Privacy Statement

Username/Email: jrc

Password:

What is eRAMS?

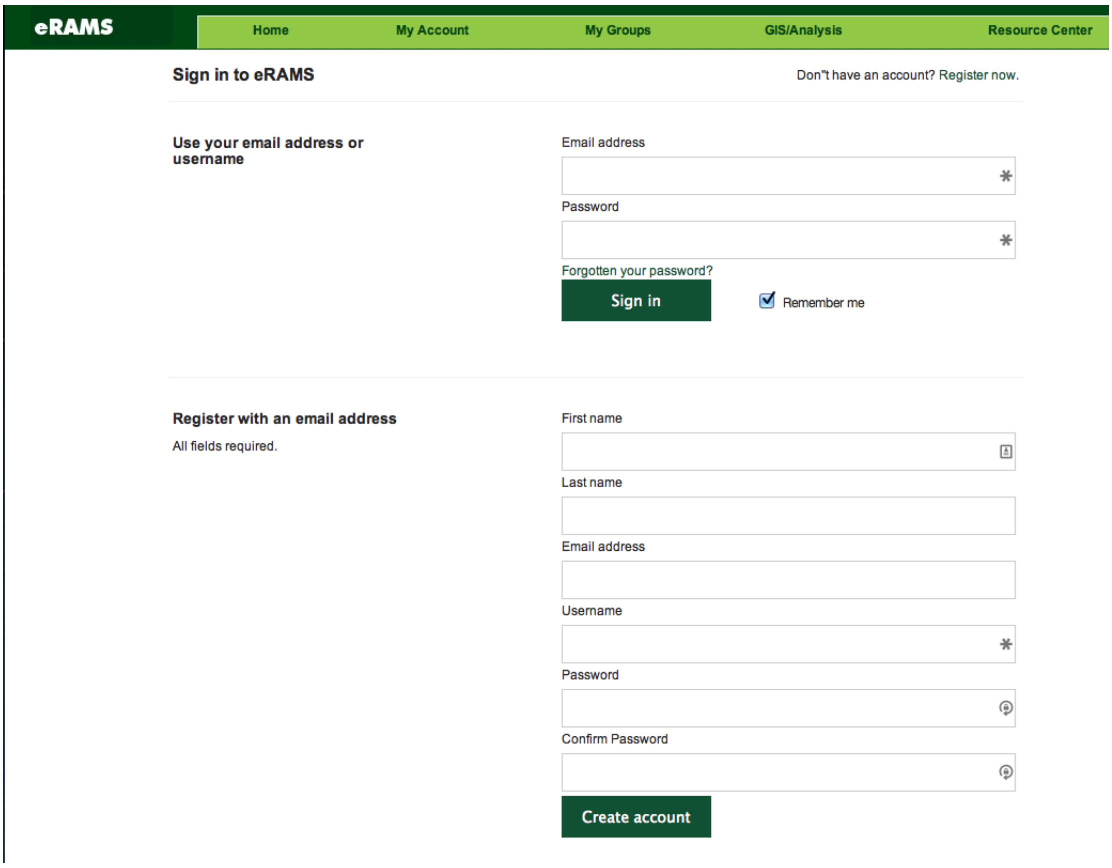
eRAMS provides online services for sustainable management of land, water, and energy resources. Users can develop a geowebpage and include data, analytical tools and simulation models, collaborate within groups and publish the project in less than 5 minutes.

eRAMS for developers

eRAMS is a powerful platform for building accessible and scalable analytical tools and simulation models that can be accessed via desktop or mobile devices.

If connected to the Internet, click on the video for an introduction and how to get started. Clicking on **Resource Center** provides additional videos providing orientation on how to use the basic features of eRAMS.

Next click Sign Up in the upper right to create an eRAMS account.



Sign in to eRAMS

Don"t have an account? Register now.

Use your email address or username

Email address

Password

Forgotten your password?

☑ Remember me

Register with an email address

All fields required.

First name

Last name

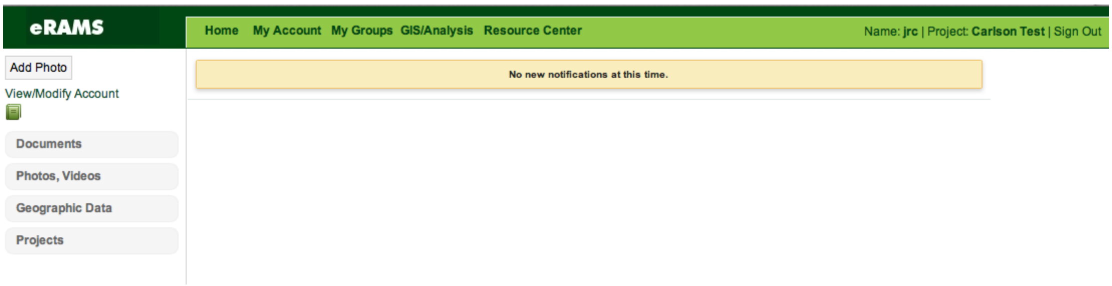
Email address

Username

Password

Confirm Password

Log in to eRAMS, which displays the following page

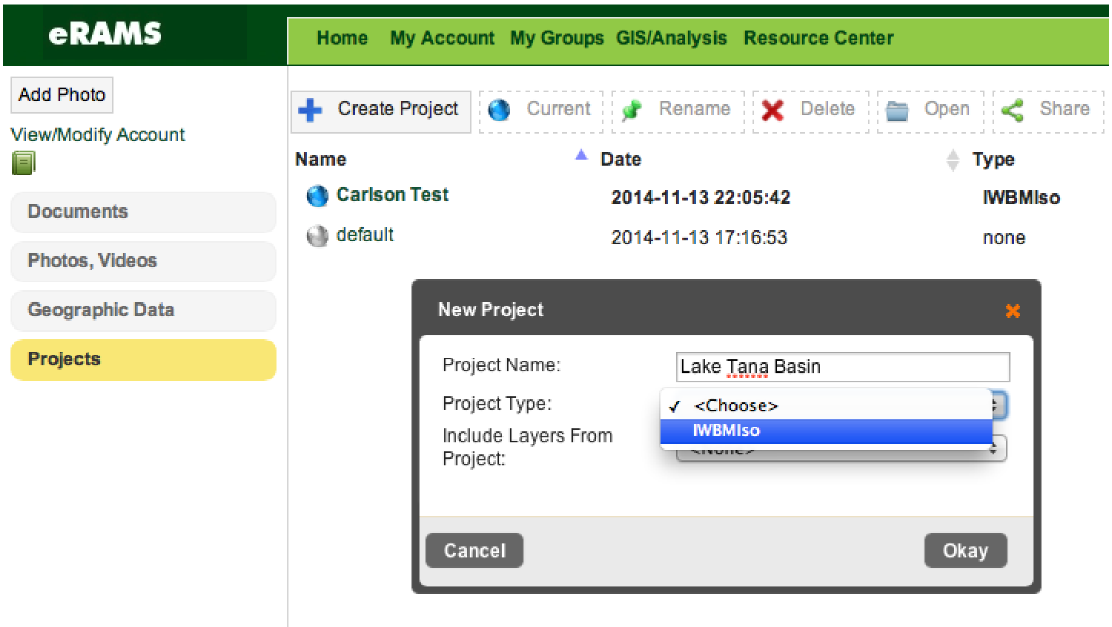


Name: jrc | Project: Carlson Test | Sign Out

No new notifications at this time.

4. Getting Started: Select or Create Project

Click **Projects** in the left panel to display a list of projects. The first time only a default project will be displayed. Click Create **Project** to open the **New Project** dialog. Name the project and select project type IWBMIso. Include layers from other projects if desired.



Name	Date	Type
Carlson Test	2014-11-13 22:05:42	IWBMIso
default	2014-11-13 17:16:53	none

New Project

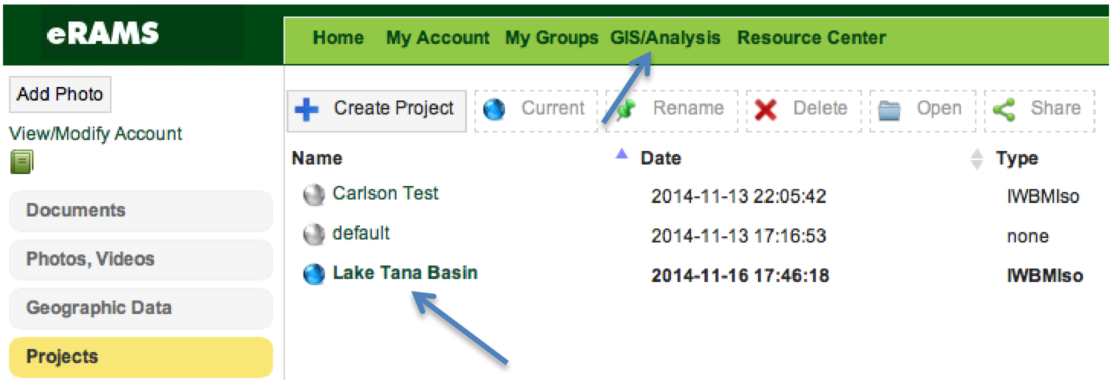
Project Name: Lake Tana Basin

Project Type: ✓ <Choose>

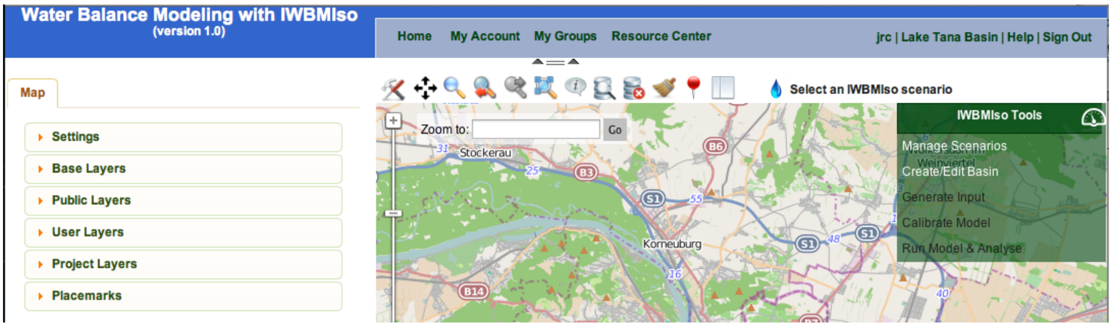
IWBMIso

Include Layers From Project: <None>

Clicking **Okay** displays the new project on the list. Click on the project name or GIS/Analysis on the top menu to go to the project.



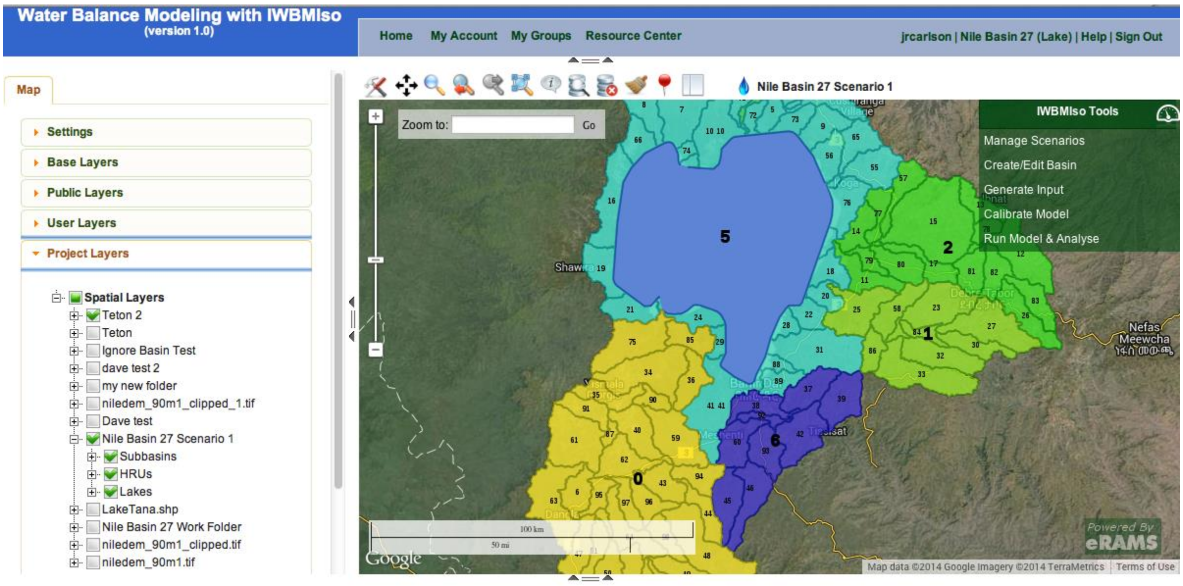
Name	Date	Type
Carlson Test	2014-11-13 22:05:42	IWBMIso
default	2014-11-13 17:16:53	none
Lake Tana Basin	2014-11-16 17:46:18	IWBMIso



A user has considerable flexibility in the way to manage projects for IWBMIso modeling. A project can have one to many adjoining or overlapping basins. A project will have one or more scenarios, each having a unique set of input variable and parameter files. To minimize complexity, a user likely will prefer one basin per project.

5. Overview of the IWBMIso Application Layout and Workflow

The basic layout for IWBMIso modeling involves the **Map** panel on the left, the map itself in the center displaying the basin(s) being modeled, the GIS toolbar above the map, and the primary modeling steps in the slideout menu on the right. The remaining sections of this user manual follow the workflow implicit in this menu.



The **Map** panel on the left side contains several accordion elements. Set the background imagery in **Base Layers**. The spatial layers associated to an IWBMIso project will be located in **Project Layers**. As the modeling worklow proceeds, the application will publish spatial layers to **Project Layers**.

From **Project Layers** turn layers off/on. Using right or left mouse click open menus and dialogs to change symbology and labels, add external layers, and perform other tasks. Consult the **Resource Center** for additional information on the functionality of the Map panel (Internet connection required for videos).

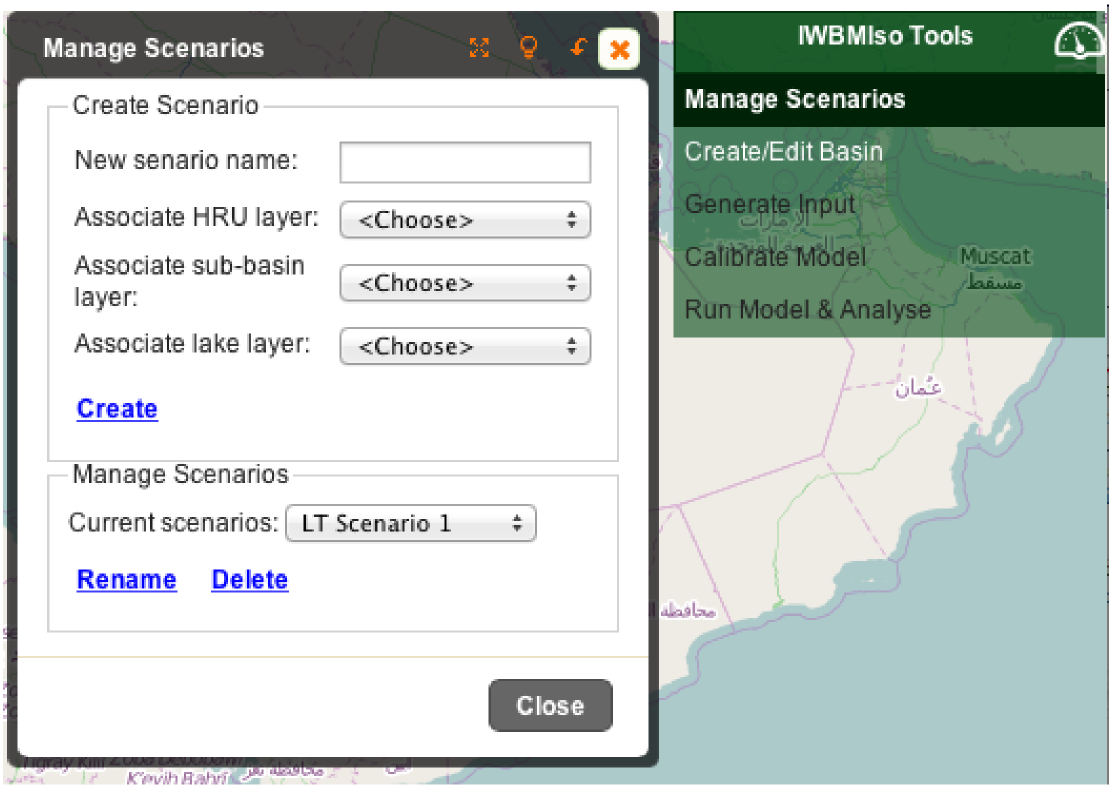
The toolbar above the map contains several standard GIS tools. Run the cursor over the icons to see the name of the tool. The left icon is **Map Tools**, which drops a menu providing digitize/edit, other standard tools, and some tools unique to eRAMS developed for other projects.

The IWBMIso workflow involves creating one or more scenarios for the project. When returning to a project, a very important step is to select the scenario from the choice list to the right of the GIS toolbar above the map. This will activate the IWBMIso menu on the right.

The following sections guide a user through the modeling workflow reflected in the IWBMIso slideout menu.

6. Manage Scenarios for a Project

A scenario contains a unique set of input variables and parameter files for a basin. One can create a scenario through **Create/Edit Basin**, involving publishing to a new or existing scenario after creating a basin, subbasins, and HRUs. The second way is to click on **Scenario Management** to open the **Manage Scenarios** dialog below.



Manage Scenarios

Create Scenario

New senario name:

Associate HRU layer: <Choose>

Associate sub-basin layer: <Choose>

Associate lake layer: <Choose>

Create

Manage Scenarios

Current scenarios: LT Scenario 1

Rename Delete

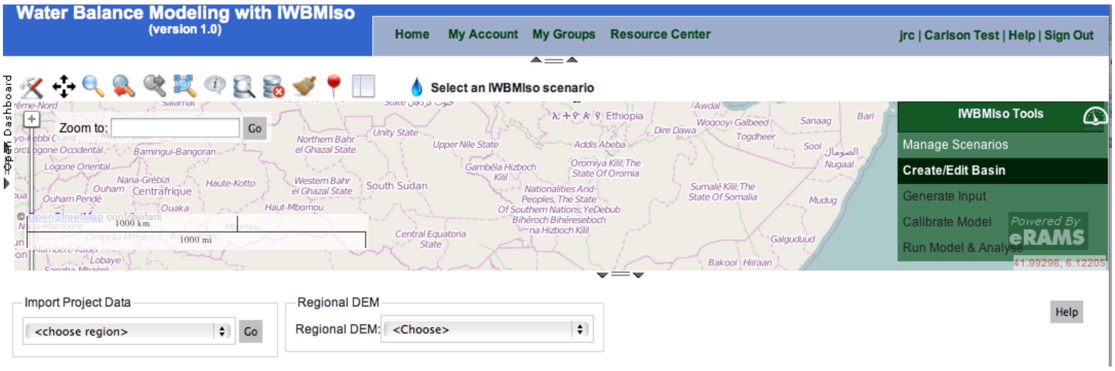
Close

Use this tool for creating scenarios using existing HRU and sub-basin layers. Also use the tool for creating a scenario without an HRU layer: keep layer entries blank and add input variable files and a parameter file later in **Generate Input**. Go to the **Generate Input** menu item and use the **Add** button to upload these files.

Use the manage scenarios section at the bottom to rename or delete existing scenarios.

7. Create/Edit Basin – Area of Interest and Work Folder

Clicking **Create/Edit Basin** in the IWBMIso menu opens the panel at the bottom.



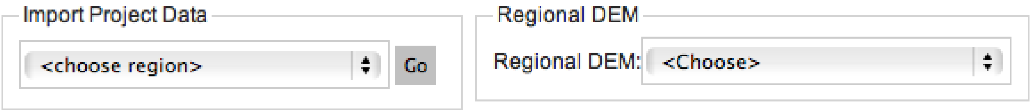
Import Project Data

<choose region> Go

Regional DEM

Regional DEM: <Choose>

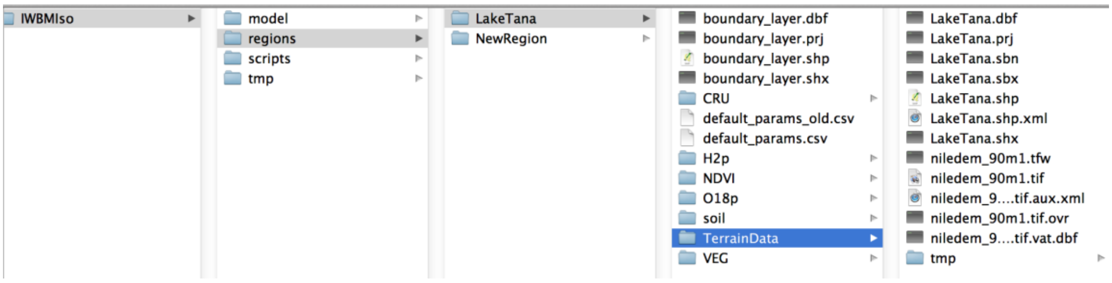
The first step to create or edit a basin is to select a region folder containing the source data you need for the project, including the DEM layer


<choose region>

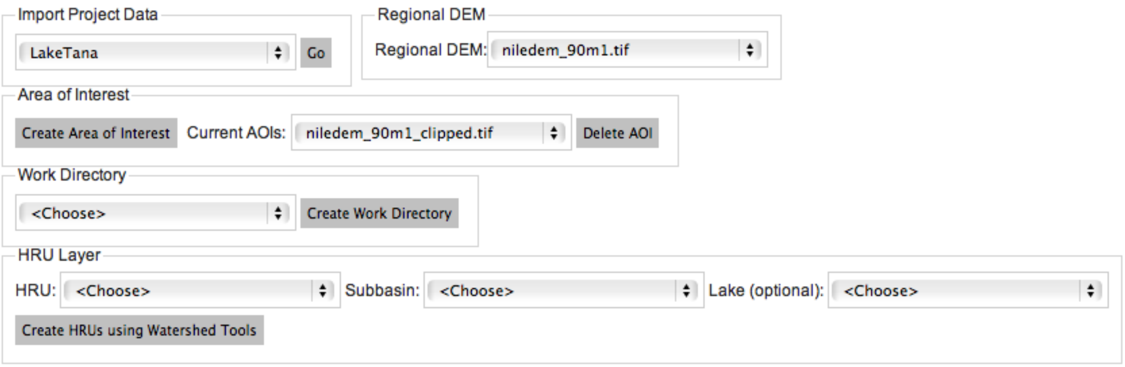
Regional DEM: <Choose>

The **Import Project Data** entry field is used to identify a previously uploaded set of data for the region of interest. These data will include as a minimum a DEM for the region containing one or more basins of interest and an associated data layer that contains the lakes within the region that will be included in the study. Other data could include the HydroSHEDS data layer for the region (http://hydrosheds.cr.usgs.gov/index.php). HydroSHEDS is a mapping product that provides hydrographic information for regional and global-scale applications in a consistent format. It offers a suite of geo-referenced data sets (vector and raster) at various scales, including river networks, watershed boundaries, drainage directions, and flow accumulations. The river network data layer is helpful when defining the rectangular area of interest (AOI) that contains the basin of interest. This AOI will be clipped from the DEM for more efficient processing.

DEM, lakes, river networks, and other data of this kind must be located in the TerrainData folder in the path …/erams/data/IWBMIso/regions/<region name>/TerrainData, for example:



In the following example, the Regional Data are named LakeTana and the Regional DEM is a 90 m resolution tif file named nile_dem_90.tif. The DEM covers the entire Nile River basin.


LakeTana

Regional DEM: niledem_90m1.tif

Current AOIs: niledem_90m1_clipped.tif

<Choose>

HRU: <Choose> Subbasin: <Choose> Lake (optional): <Choose>

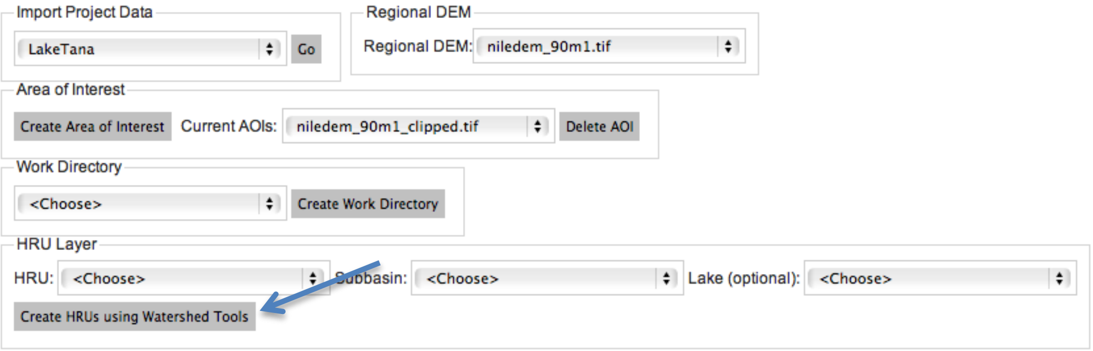
Clicking on **Create Area of Interest** enables the user to draw a rectangular box around an area that contains the Lake Tana basin. The AOI in this example is for the watershed that has the outflow point at the outflow of Lake Tana. The name of the map layer created in the clipping process is nile_dem_90_clipped.tif. If the created AOI is not sufficient to cover the basin, delete it with the Delete AOI button and then repeat the process.

After creating an AOI the next step is to create a work directory (folder), which will contain the spatial layers generated when delineating the watershed (basin). The application defaults a folder name, but it can be renamed as appropriate.

Next, click on the **Create HRUs using Watershed Tools** button to open the **Delineate Watershed** tool. This tool will guide through a process to create the subbasin and HRU spatial layers. When done you return to the **Create/Edit Basin** panel and continue to establish the spatial layers for the scenario.

8. Create/Edit Basin - Delineate Watershed

Delineate watershed is part of the **Create/Edit Basin** workflow step, but also can be opened under **Map Tools** in the GIS toolbar.



Import Project Data: LakeTana

Regional DEM: niledem_90m1.tif

Area of Interest

Current AOIs: niledem_90m1_clipped.tif

Work Directory

<Choose>

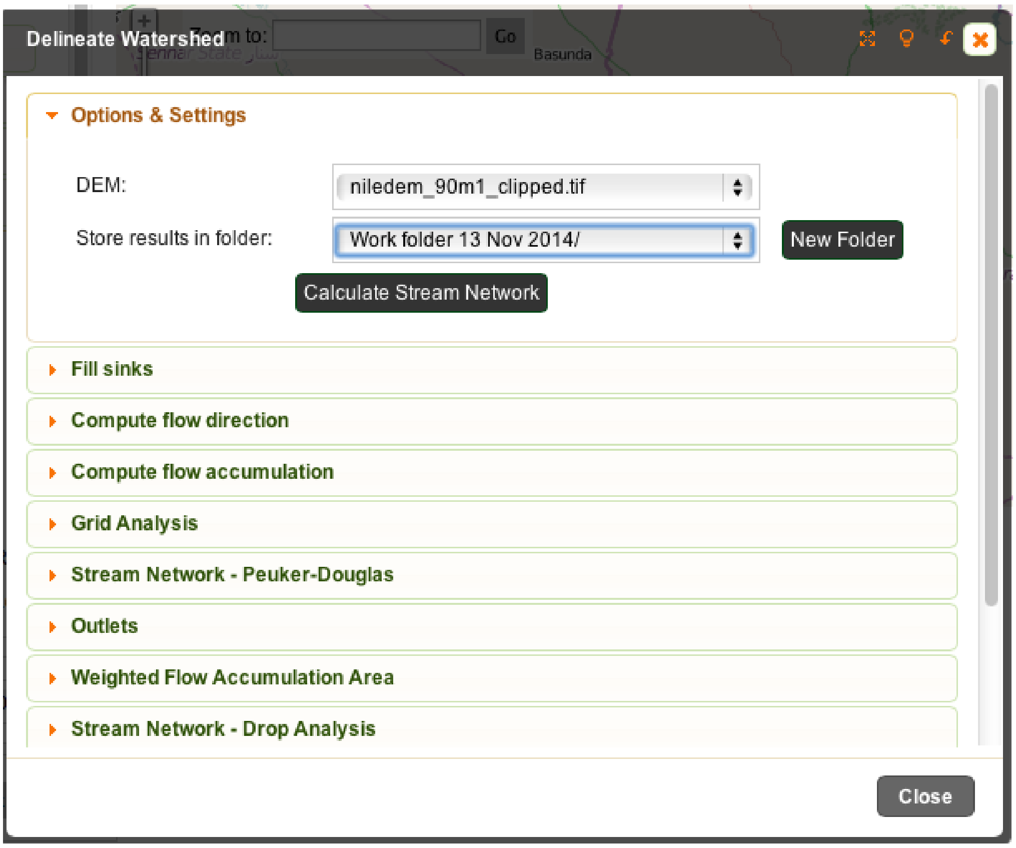
HRU Layer

HRU: <Choose> Subbasin: <Choose> Lake (optional): <Choose>

Creating a basin, subbasins, and HRUs is a multi-step process using the TAUDEM toolset version 5.1.2 (http://hydrology.usu.edu/taudem/taudem5/index.html). The tool is used to generate both the sub-watersheds (HRUs) and sub-basins for the area of interest. With each step the application defaults most of the output file names using the DEM name as the prefix.

In **Options and Settings** assign the DEM and create or assign a folder to store the intermediate and final results. The work folder will be checked for any existing intermediate rasters and fill in the drop down lists for any tools that were already run.

Click **Calculate Stream Network** in **Options and Settings** to automatically perform all calculations until **Outlets**. Otherwise manually go through the steps.



Delineate Watershed

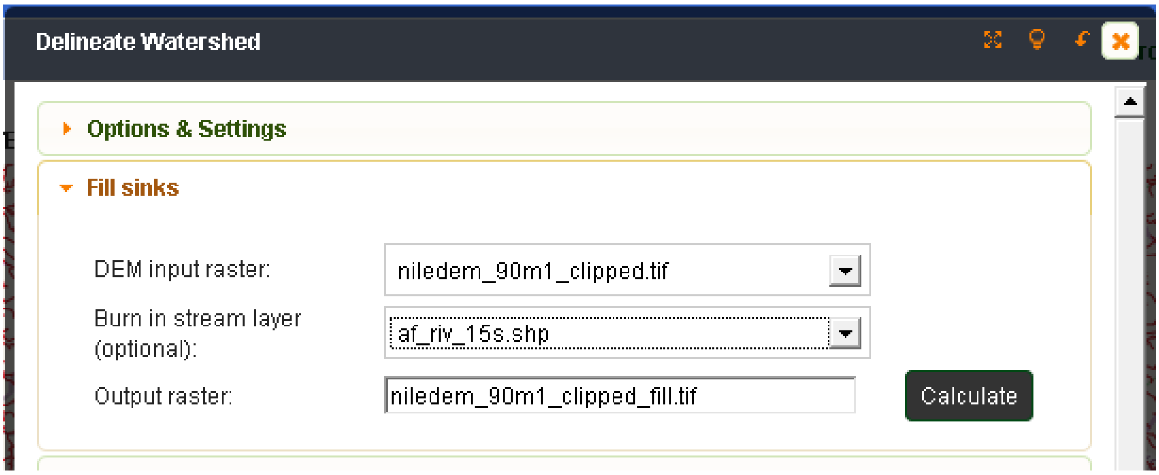
Options & Settings

DEM: niledem_90m1_clipped.tif

Store results in folder: Work folder 13 Nov 2014/

New Folder
Calculate Stream Network
Fill sinks
Compute flow direction
Compute flow accumulation
Grid Analysis
Stream Network - Peuker-Douglas
Outlets
Weighted Flow Accumulation Area
Stream Network - Drop Analysis
Close

The Fill Sinks step removes pits from the DEM so that water from any point in the DEM is able to drain off at the edge. Using a stream line layer, such as the Hydrosheds African River example below, can force channels to be created by the sream layer selected in **Burn in stream layer**; otherwise stream channels will be calculated according to the DEM. Click the **Calculate** button to generate the filled DEM raster.



Delineate Watershed

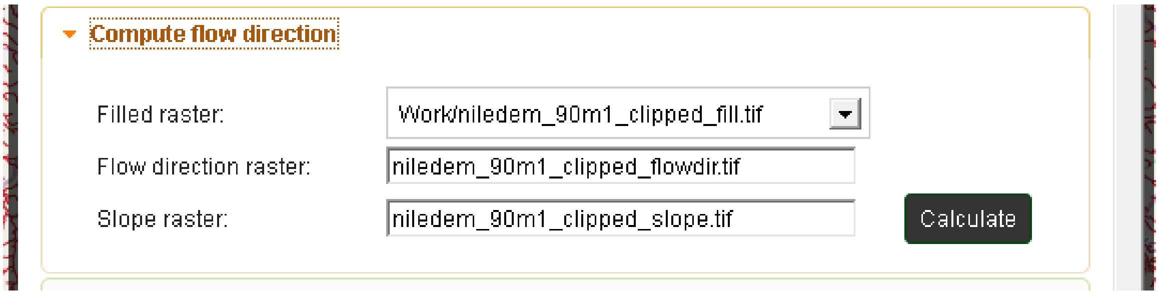
Fill sinks

DEM input raster: niledem_90m1_clipped.tif

Burn in stream layer (optional): af_riv_15s.shp

Output raster: niledem_90m1_clipped_fill.tif

In the **Compute Flow Direction** step, click **Calculate** to generate the flow direction and output rasters.

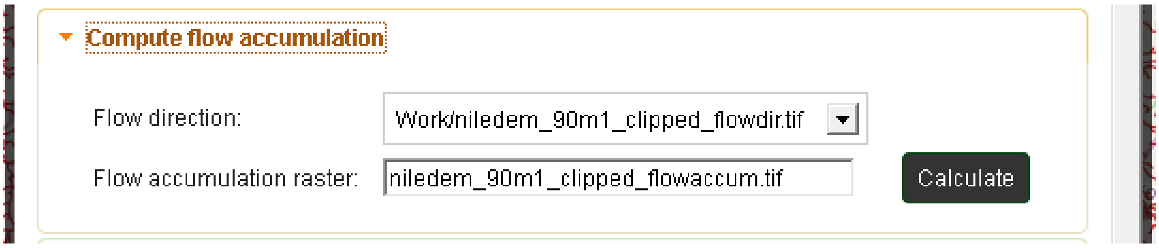

Compute flow direction

Filled raster: Work/niledem_90m1_clipped_fill.tif

Flow direction raster: niledem_90m1_clipped_flowdir.tif

Slope raster: niledem_90m1_clipped_slope.tif

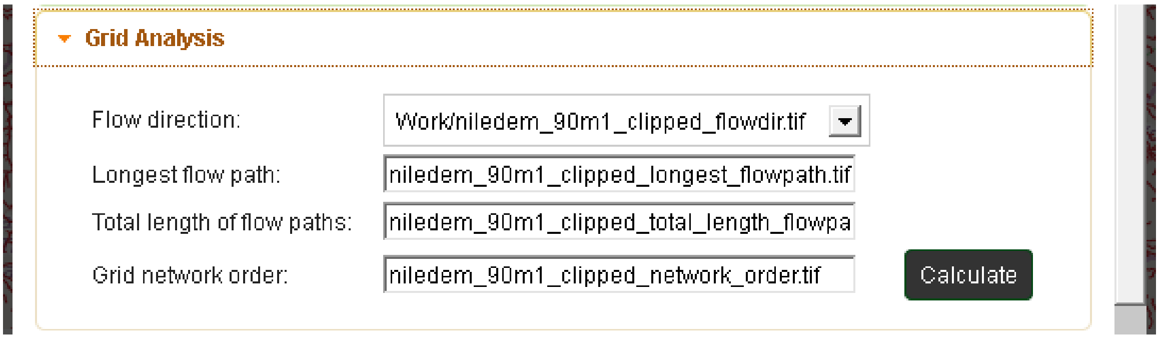
In the **Compute Flow Accumulation** step click **Calculate** to generate the flow accumulation raster.


Compute flow accumulation

Flow direction: Work/niledem_90m1_clipped_flowdir.tif

Flow accumulation raster: niledem_90m1_clipped_flowaccum.tif

In the **Grid Analysis** step click **Calculate** to generate the longest flow path, total length of flow paths and the grid network order rasters.


Grid Analysis

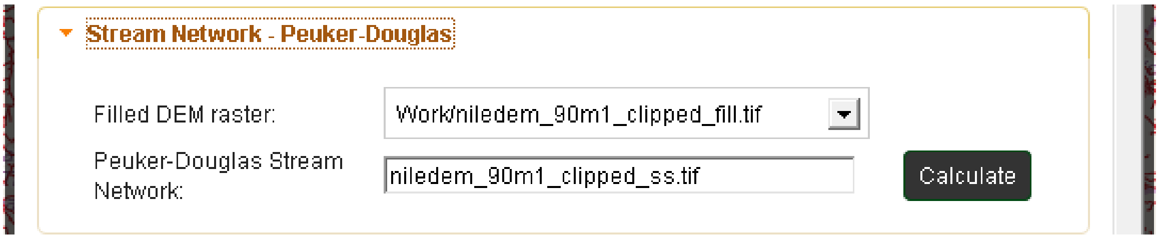
Flow direction: Work/niledem_90m1_clipped_flowdir.tif

Longest flow path: niledem_90m1_clipped_longest_flowpath.tif

Total length of flow paths: niledem_90m1_clipped_total_length_flowpa

Grid network order: niledem_90m1_clipped_network_order.tif

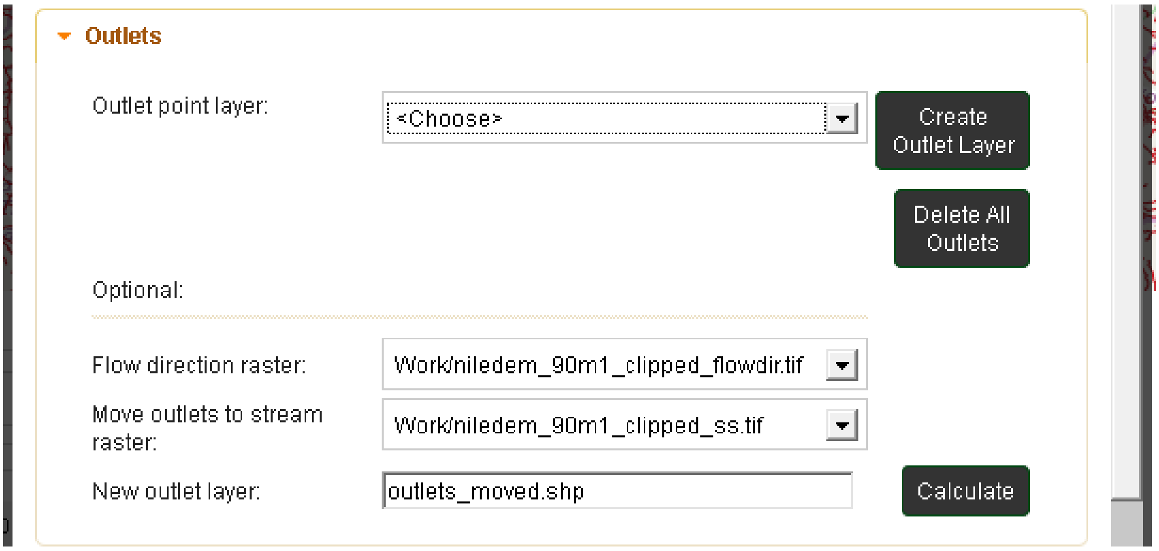
In the **Stream Network – Peuker-Douglas** step click **Calculate** to generate the Peuker-Douglas stream network raster.


Stream Network - Peuker-Douglas

Filled DEM raster: Work/niledem_90m1_clipped_fill.tif

Peuker-Douglas Stream Network: niledem_90m1_clipped_ss.tif

Outlets define the watershed and the sub-basin polygons. In order to get reasonable results, the outlets must be placed on a stream channel.



Outlet point layer: <Choose>

Optional:

Flow direction raster: Work/niledem_90m1_clipped_flowdir.tif

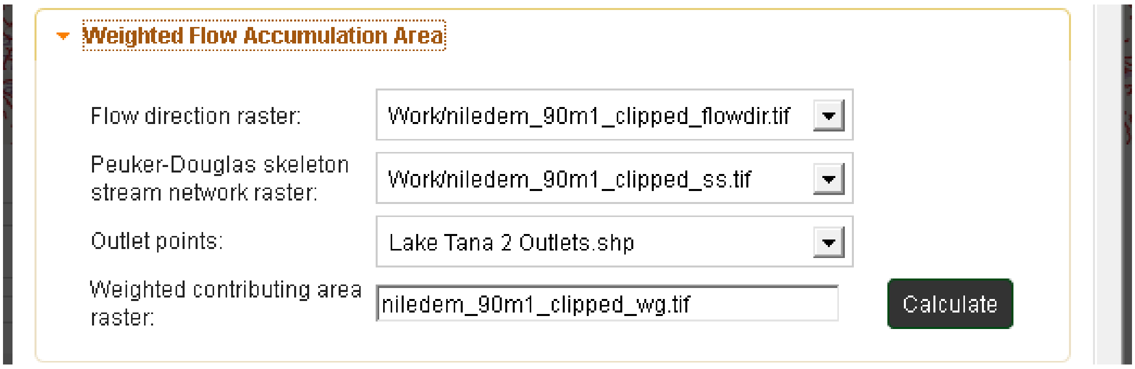
Move outlets to stream raster: Work/niledem_90m1_clipped_ss.tif

New outlet layer: outlets_moved.shp

Use **Create Outlet Layer** to add a new point layer to your **Project Layers** that will be used to store the outlet features. Also an existing outlet layer can be selected in the dropdown list instead of creating a new one. When an outlet layer is selected, the **Add Outlet** button will appear. Click that button to add outlet points interactively on the map. It is recommended that the outlets be placed on channels given by the flow accumulation raster (in our example, *niledem_90m1_clipped_flowaccum.tif*). This process requires zooming in to the extent where the outlet point can be correctly placed on a channel pixel. Adjusting the transparency and symbology of the flow accumulation raster also aids accurate location of the outlet.

Click the **Calculate** button to ensure that the outlet points that you created are on the stream network raster. Otherwise continue on to the next step.

In the **Weighted Flow Accumulation Area** step click **Calculate** to generate the weighted contributing area raster.



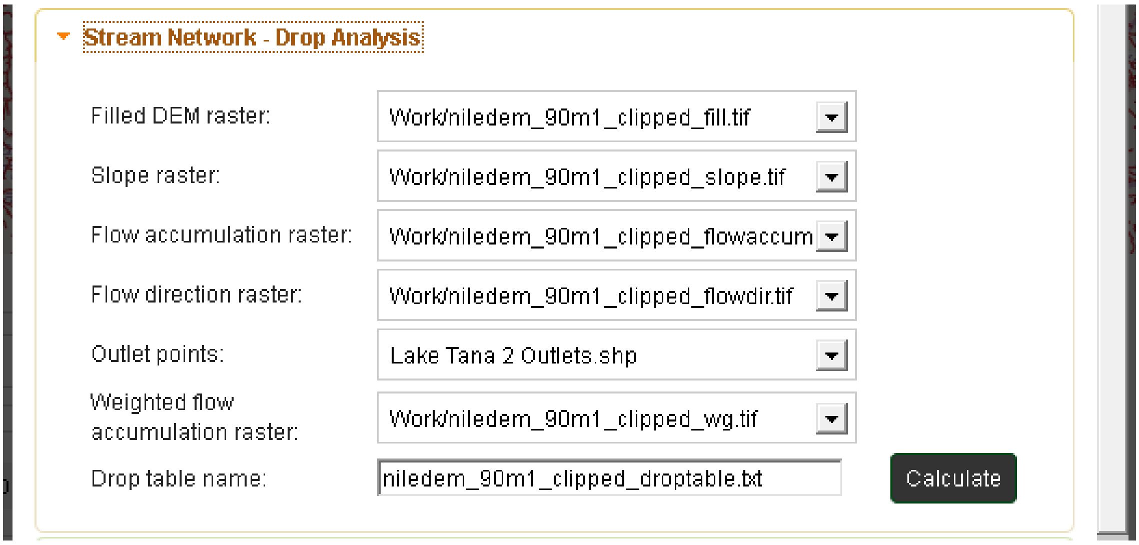
Flow direction raster: Work/niledem_90m1_clipped_flowdir.tif

Peuker-Douglas skeleton stream network raster: Work/niledem_90m1_clipped_ss.tif

Outlet points: Lake Tana 2 Outlets.shp

Weighted contributing area raster: niledem_90m1_clipped_wg.tif

In the **Stream Network – Drop Analysis** step click **Calculate** to generate the drop analysis table.



Stream Network - Drop Analysis

Filled DEM raster: Work/niledem_90m1_clipped_fill.tif

Slope raster: Work/niledem_90m1_clipped_slope.tif

Flow accumulation raster: Work/niledem_90m1_clipped_flowaccum

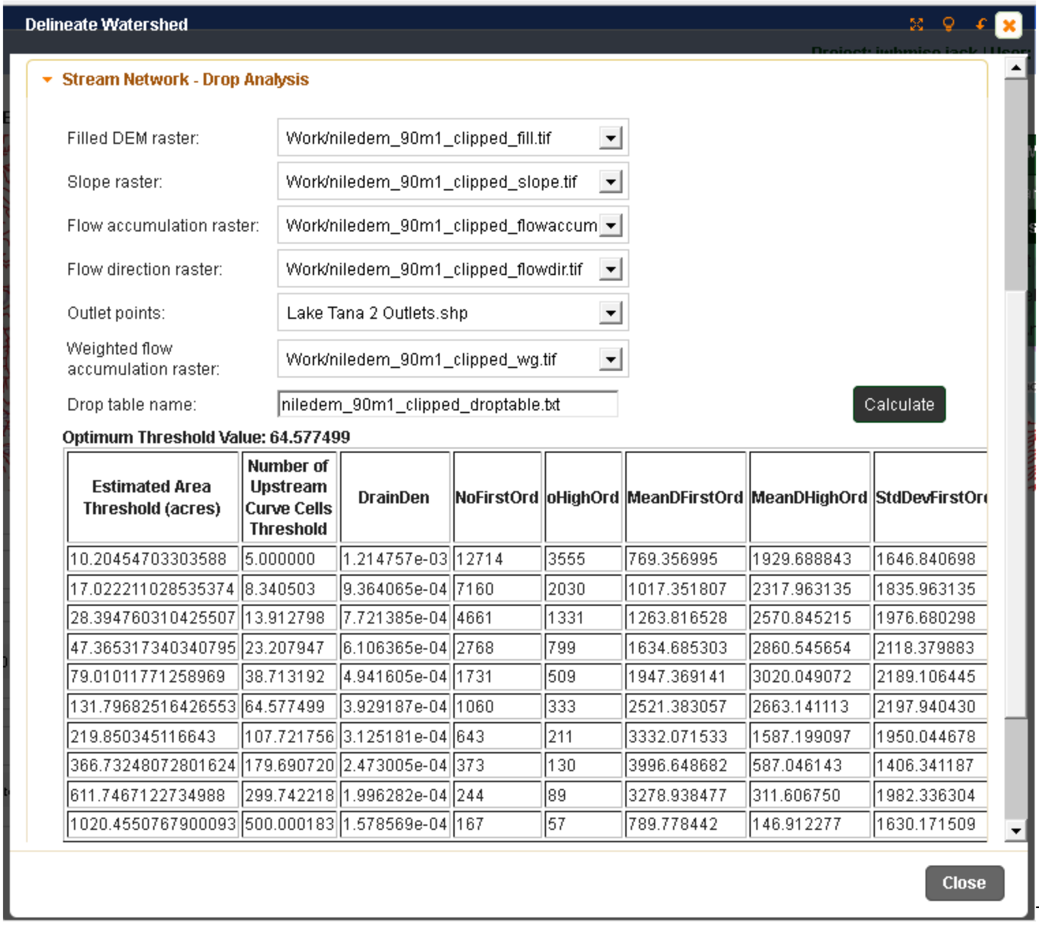
Flow direction raster: Work/niledem_90m1_clipped_flowdir.tif

Outlet points: Lake Tana 2 Outlets.shp

Weighted flow accumulation raster: Work/niledem_90m1_clipped_wg.tif

Drop table name: niledem_90m1_clipped_droptable.txt

The drop analysis table provides information as to which threshold to use in the next step for generating HRU sub-watershed polygons. For a given value of **Number of Upstream Curve Cells Threshold**, the **No FirstOrd** is the number of first order drainages (HRUs) that would be created if that threshold value were used in the next step. If fewer first order drainages than shown in the table are desired, use a higher threshold value. Iterating a few times will develop a sense for a value that satisfies the scale of the HRUs needed for modeling.



Delineate Watershed

Stream Network - Drop Analysis

Filled DEM raster: Work/niledem_90m1_clipped_fill.tif

Slope raster: Work/niledem_90m1_clipped_slope.tif

Flow accumulation raster: Work/niledem_90m1_clipped_flowaccum

Flow direction raster: Work/niledem_90m1_clipped_flowdir.tif

Outlet points: Lake Tana 2 Outlets.shp

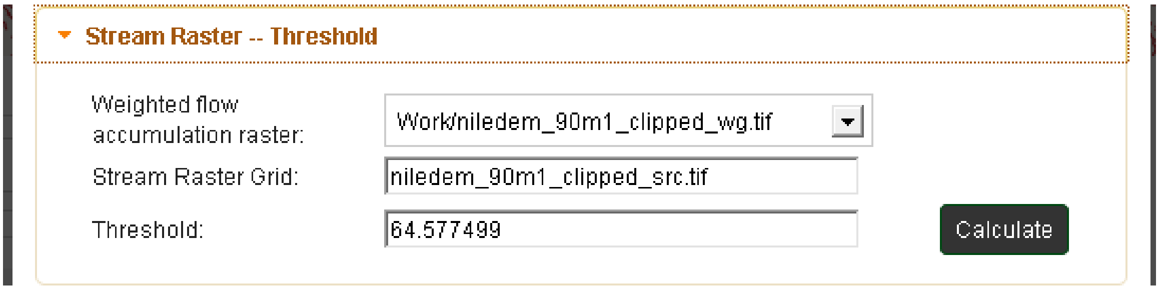
Weighted flow accumulation raster: Work/niledem_90m1_clipped_wg.tif

Drop table name: niledem_90m1_clipped_droptable.txt

Optimum Threshold Value: 64.577499

Estimated Area Threshold (acres)	Number of Upstream Curve Cells Threshold	DrainDen	NoFirstOrd	oHighOrd	MeanDFirstOrd	MeanDHighOrd	StdDevFirstOr
10.20454703303588	5.000000	1.214757e-03	12714	3555	769.356995	1929.688843	1646.840698
17.022211028535374	8.340503	9.364065e-04	7160	2030	1017.351807	2317.963135	1835.963135
28.394760310425507	13.912798	7.721385e-04	4661	1331	1263.816528	2570.845215	1976.680298
47.365317340340795	23.207947	6.106365e-04	2768	799	1634.685303	2860.545654	2118.379883
79.01011771258969	38.713192	4.941605e-04	1731	509	1947.369141	3020.049072	2189.106445
131.79682516426553	64.577499	3.929187e-04	1060	333	2521.383057	2663.141113	2197.940430
219.850345116643	107.721756	3.125181e-04	643	211	3332.071533	1587.199097	1950.044678
366.73248072801624	179.690720	2.473005e-04	373	130	3996.648682	587.046143	1406.341187
611.7467122734988	299.742218	1.996282e-04	244	89	3278.938477	311.606750	1982.336304
1020.4550767900093	500.000183	1.578569e-04	167	57	789.778442	146.912277	1630.171509

In the **Stream Raster – Threshold** step click **Calculate** to generate the stream raster grid based on the value entered into the Threshold field.



Stream Raster -- Threshold

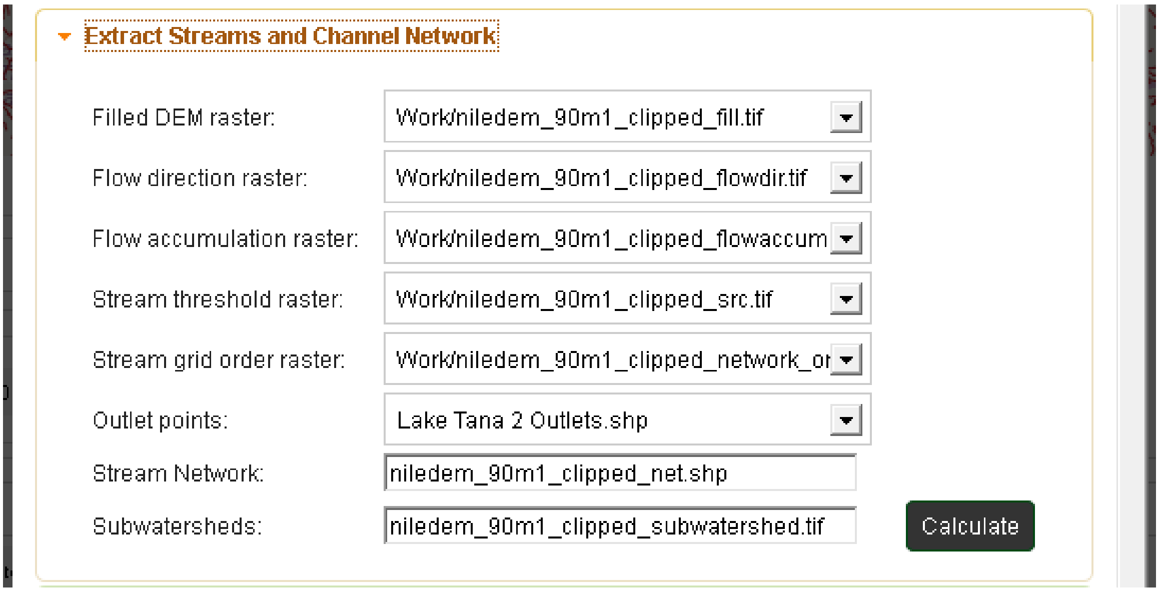
Weighted flow accumulation raster: Work/niledem_90m1_clipped_wg.tif

Stream Raster Grid: niledem_90m1_clipped_src.tif

Threshold: 64.577499

The application defaults a threshold value. As needed, it can be change to one more appropriate for the modeling scenario.

In the **Extract Streams and Channel Network** Step click **Calculate** to generate the stream network line layer and the subwatershed raster. The subwatershed raster will be exported to a polygon layer and used as the starting point for the HRU layer.



Extract Streams and Channel Network

Filled DEM raster: Work/niledem_90m1_clipped_fill.tif

Flow direction raster: Work/niledem_90m1_clipped_flowdir.tif

Flow accumulation raster: Work/niledem_90m1_clipped_flowaccum

Stream threshold raster: Work/niledem_90m1_clipped_src.tif

Stream grid order raster: Work/niledem_90m1_clipped_network_or

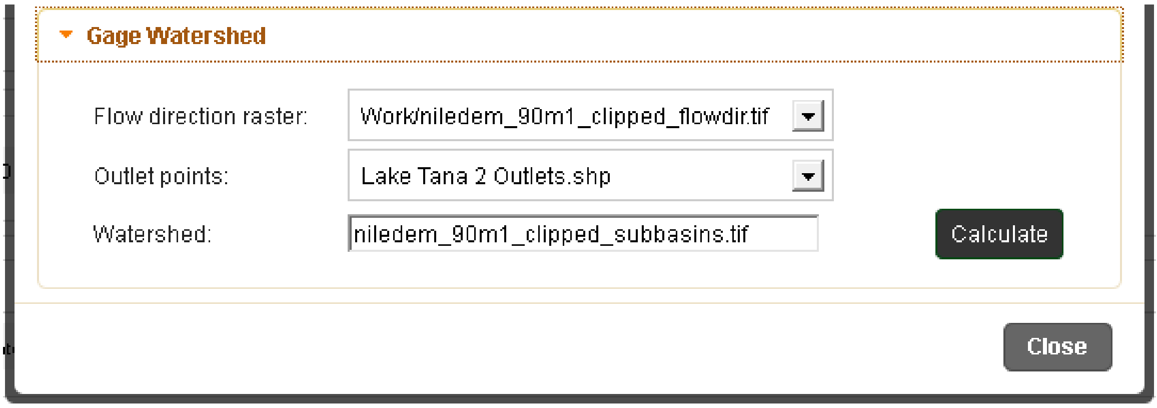
Outlet points: Lake Tana 2 Outlets.shp

Stream Network: niledem_90m1_clipped_net.shp

Subwatersheds: niledem_90m1_clipped_subwatershed.tif

To this point in the **Delineate Watershed** process most steps have generated raster layers (.tif) and placed them in the previously specified work folder. The **Extract Streams and Channel Network** step generates a HRU shape file (.shp vector layer) placed in the work folder.

In the **Gage Watershed** step click **Calculate** to generate the contributing area of each outlet point, which will become the subbasin polygon layer.



Gage Watershed

Flow direction raster: Work/niledem_90m1_clipped_flowdir.tif

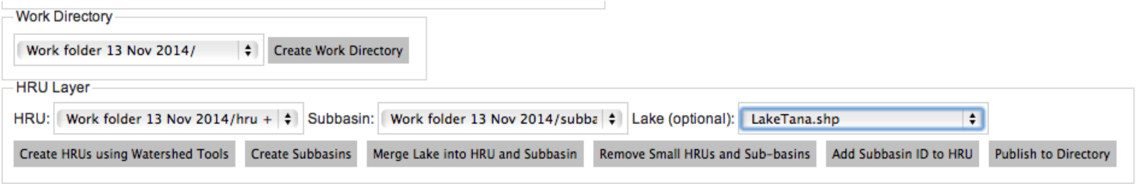
Outlet points: Lake Tana 2 Outlets.shp

Watershed: niledem_90m1_clipped_subbasins.tif

This is the last step in the **Delineate Watershed** tool. Click close and return to the **Create/Edit Basin** panel at the bottom.

9. Create/Edit Basin – Finish and Publish to Scenario

Upon completing the subbasin and HRU layers, the next step is to merge any lakes in the basin with these layers. Making sure the correct HRU and subbasin layers are selected, choose a shape file containing lake geometry and click the **Merge** button. The process results in a lake becoming an HRU and subbasin in their respective layers.



Work Directory

Work folder 13 Nov 2014/

HRU Layer

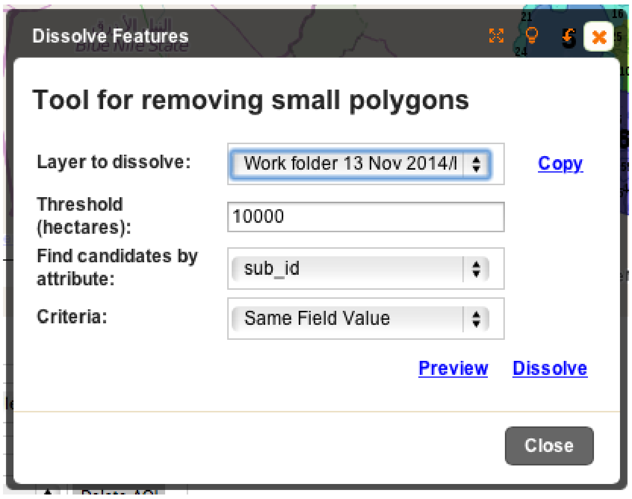
HRU: Work folder 13 Nov 2014/hru +

Subbasin: Work folder 13 Nov 2014/subba

Lake (optional): LakeTana.shp

Create Work Directory
Create HRUs using Watershed Tools
Create Subbasins
Merge Lake into HRU and Subbasin
Remove Small HRUs and Sub-basins
Add Subbasin ID to HRU
Publish to Directory

The next step is to dissolve unwanted small polygons in the HRU and subbasin layers and establish a minimum HRU or subbasin size. Set the threshold value. The bottom two fields apply conditions to the dissolve, in the example, to dissolve small HRUs into a larger one not spanning subbasins.



Dissolve Features

Tool for removing small polygons

Layer to dissolve: Work folder 13 Nov 2014/l

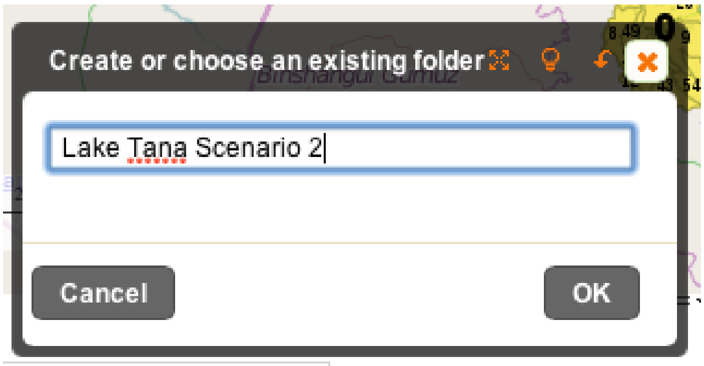
Threshold (hectares): 10000

Find candidates by attribute: sub_id

Criteria: Same Field Value

When the dissolve process is completed, click the **Add Subbasin ID to HRU** button, which updates the underlying HRU attribute table. Examine HRU attributes by right clicking the HRU layer in **Project Layers** to open the attribute table.

The final step in **Create/Edit Basin** is to publish the finished subbasins and HRUs to a scenario folder. Publish to a new scenario or overwrite and an existing scenario.



Create or choose an existing folder

Lake Tana Scenario 2

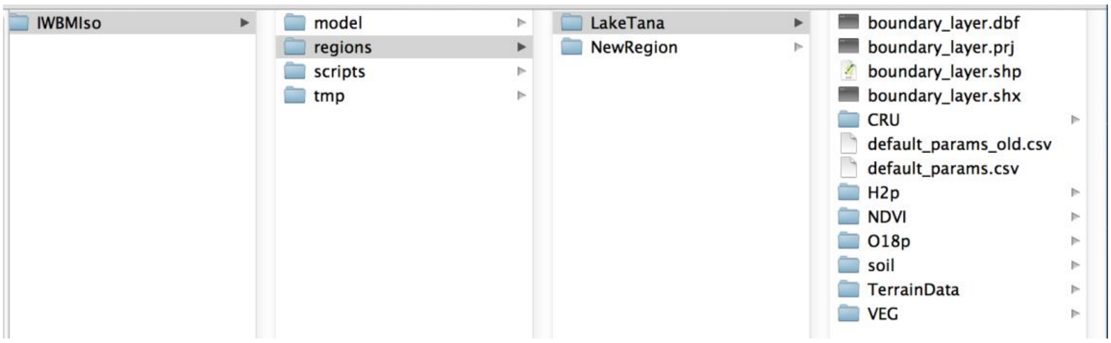
10. Generate Input – Get Source Data for Pre-Processing

The IWBMIso application contains global source datasets for precipitation, minimum/maximum temperature, relative humidity, and potential evapotranspiration from the Climate Research Unit (CRU) at the University of East Anglia. A global dataset for available soil water capacity from the FAO Soils Portal also is provided. The application also provides 90m DEM for the Nile Basin, dH2 and dO18 isotopes for Africa, and MODIS NDVI and VCF for a small region surrounding Lake Tana in Ethiopia. The datasets that come with the application are contained in a folder called “LakeTana” under a folder called “regions.” The folder path is

```
…/erams/data/IWBMIso/regions/LakeTana
```

If the project is located beyond the Lake Tana area, the user must download MODIS data for your project area. Likewise download DEM for locations beyond the Nile Basin, and isotope data for locations beyond Africa.

There are two ways to get source data for the regions folder. One way is to download needed source data as described later in this section and put in a subfolder under the regions folder. If the project is located beyond the immediate vicinity of Lake Tana, download additional MODIS data and put in the NDVI and VEG folders. If the project is outside the Nile Basin, and Africa, copy CRU and Soil data from Lake Tana to a new regions folder, then get DEM, isotope, and MODIS data for the project area and put in their respective subfolders for the new region. This remains a manual process with IWBMIso 1.0 and automating these steps will be a future enhancement.



IWBMIso

model
regions
scripts
tmp

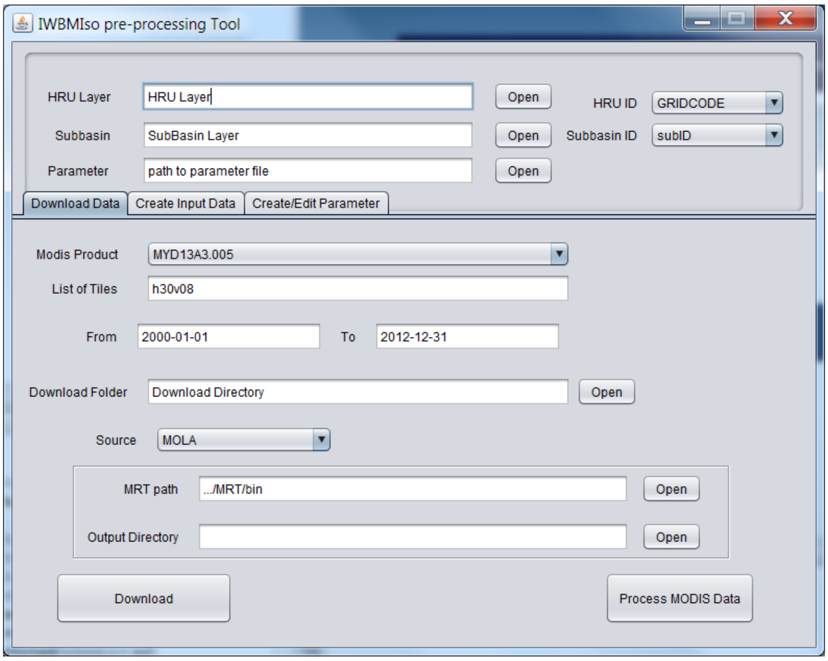
LakeTana
NewRegion

boundary_layer.dbf
boundary_layer.prj
boundary_layer.shp
boundary_layer.shx
CRU
default_params_old.csv
default_params.csv
H2p
NDVI
O18p
soil
TerrainData
VEG

The second option is to get the needed source data and use the stand-alone IWBMIso Pre-Processing tool with the scenario HRU layer. With this option the source data can be placed in any desired folder, and when ready, run the pre-processing tool to create input variable files. These files can be added your project scenario during the Generate Input Add step discussed later in this guide.

The GUI for the pre-processing tool is shown below. The data elements processed using the GUI are MODIS NDVI, MODIS VCF, CRU meteorological data, and isotope data.

The three processing tabs provide a series of downloading and processing functions, dependent on the variable being processed.



IWBMIso pre-processing Tool

HRU Layer: HRU Layer
Subbasin: SubBasin Layer
Parameter: path to parameter file
HRU ID: GRIDCODE
Subbasin ID: subID

Download Data | Create Input Data | Create/Edit Parameter

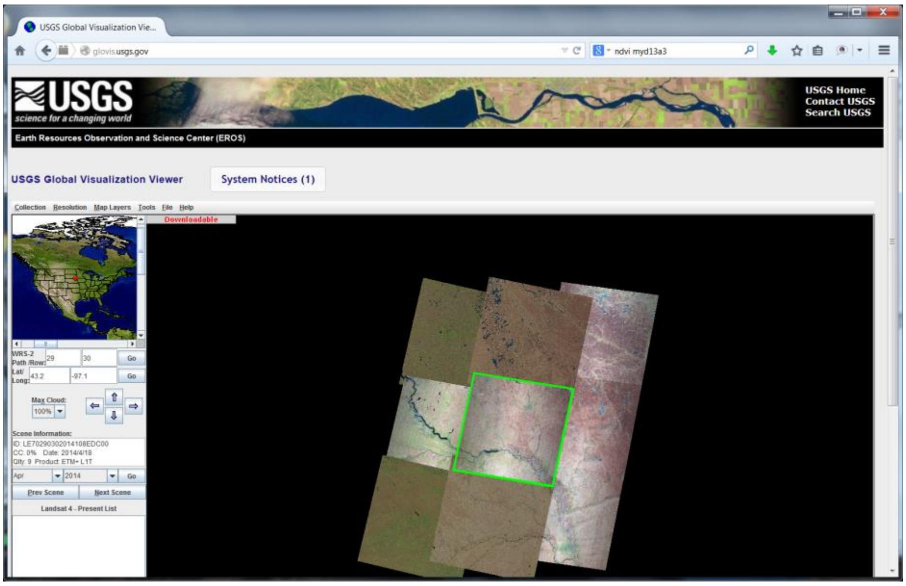
Modis Product: MYD13A3.005
List of Tiles: h30v08
From: 2000-01-01
To: 2012-12-31
Download Folder: Download Directory
Source: MOLA
MRT path: ../MRT/bin
Output Directory:

The following subsections discuss how to obtain the sources of data needed to pre-process into input variable and parameter files.

10.1. Get MODIS NDVI Source Data for the Project Region

The pre-processing tool provides a convenient way to download **MODIS MYD13A3.005 – Vegetation Indices Monthly L3 Global 1km (NDVI)** data for your project. Refer to https://lpdaac.usgs.gov/products/modis_products_table/myd13a3 for additional information about this data.

The Data Download tab provides a tool to download monthly NDVI files from MODIS Aqua data pool at http://e4ftl01.cr.usgs.gov/MOLA/MYD13A3.005/. Getting the correct data for the area requires knowing the ID of the MODIS tile(s) for that area. Use the USGS Global Visualization Viewer (http://glovis.usgs.gov/) to obtain that information.


USGS Global Visualization Vie...
glovis.usgs.gov
ndvi myd13a3
USGS
science for a changing world
USGS Home
Contact USGS
Search USGS
Earth Resources Observation and Science Center (EROS)
USGS Global Visualization Viewer
System Notices (1)
Collection Resolution Map Layers Tools File Help
Downloadable
Go
Max Cloud:
Prev Scene
Next Scene
Landsat 4 - Present List

WRS-2 Path/Row: 29 30

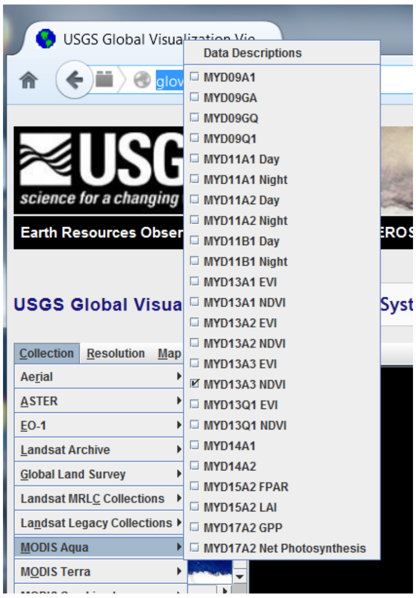
Lat/Long: 43.2 -97.1

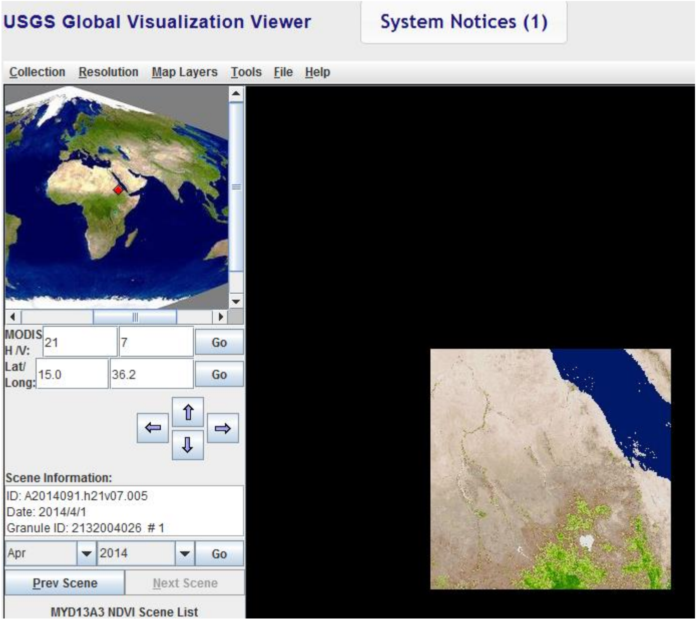
Max Cloud: 100%

Scene Information:
ID: LE70290302014108EDC00
CC: 0% Date: 2014/4/18
Qlty: 9 Product: ETM+ L1T

Apr 2014

Click on the Collection tab and select MODIS Aqua. Then select MYD13A3 NDVI in the Data Descriptions menu. Then Click on the Resolution tab and select 5000.




USGS Global Visualization Viewer
System Notices (1)
Collection Resolution Map Layers Tools File Help
Go
Go
Go
Prev Scene
Next Scene
MYD13A3 NDVI Scene List

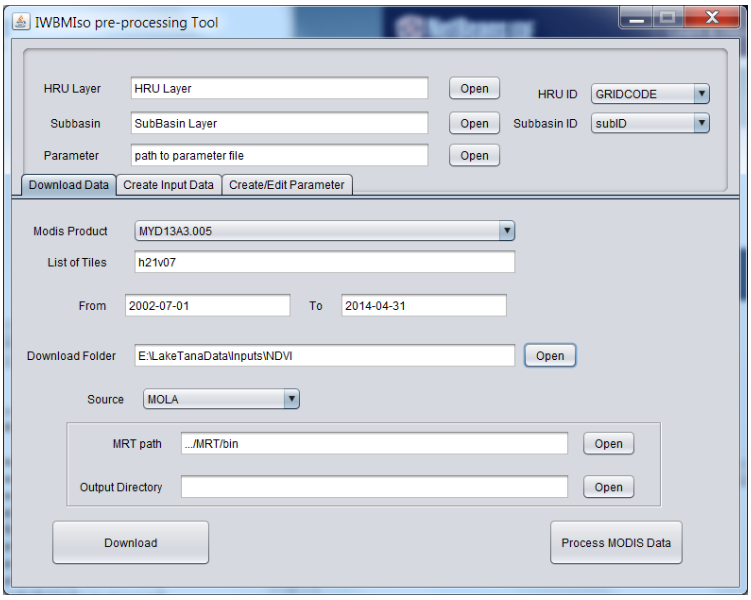
MODIS H/V: 21 7

Lat/Long: 15.0 36.2

Scene Information:
ID: A2014091.h21v07.005
Date: 2014/4/1
Granule ID: 2132004026 # 1

Apr 2014

Then using the cursor crosshair in the small global map window, click on a point that would be included in the basin of interest. A small window displaying the NDVI data for the month and year shown can then be viewed to see if this is the region of interest. Use the arrow buttons to move around the region selected to insure the tile(s) are relevant to the region. Multiple tiles may be needed for the basin. The MODIS H/V ID is shown for each tile displayed, and these IDs are needed for input to the pre-processing tool. The tile ID shown here is h21v07. Scan through the images for this location using the Prev Scene and Next Scene buttons.

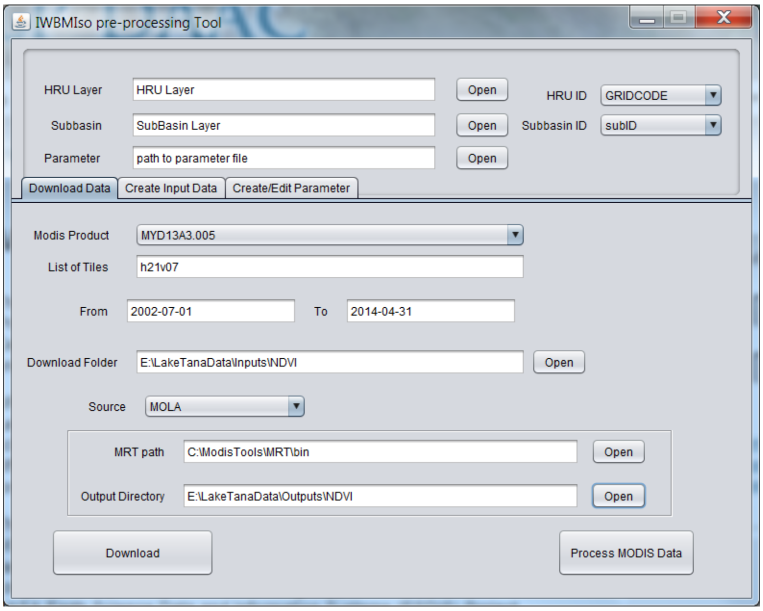

IWBMIso pre-processing Tool

HRU Layer: HRU Layer
HRU ID: GRIDCODE
Subbasin: SubBasin Layer
Subbasin ID: subID
Parameter: path to parameter file

Modis Product: MYD13A3.005
List of Tiles: h21v07
From: 2002-07-01 To: 2014-04-31
Download Folder: E:\LakeTanaData\Inputs\NDVI
Source: MOLA
MRT path: .../MRT/bin
Output Directory:

To download the monthly data files, enter the list of tiles to be downloaded, here h21v07, and then enter the period of data desired. The starting date of the data can be found on the information page (http://e4ftl01.cr.usgs.gov/MOLA/MYD13A3.005/). Then enter the directory where the downloaded files will be stored. Select the Source as MOLA, for MODIS Aqua, and then click Download.

Once the MODIS NDVI files are downloaded, the next step is to process them for use with the HRU shapefile to generate NDVI values for each HRU. To process these data install the MODIS Reprojection Tool (MRT) software on the computer containing the files. Download MRT software by first creating an account and logging in at https://lpdaac.usgs.gov/tools/modis_reprojection_tool. There is no charge for the software. Follow the installation instructions provided for MRT. The process requires the latest version of Java JRE to be able to run MRT.



HRU Layer: HRU Layer
Subbasin: SubBasin Layer
Parameter: path to parameter file
HRU ID: GRIDCODE
Subbasin ID: subID
Modis Product: MYD13A3.005
List of Tiles: h21v07
From: 2002-07-01 To: 2014-04-31
Download Folder: E:\LakeTanaData\Inputs\NDVI
Source: MOLA
MRT path: C:\ModisTools\MRT\bin
Output Directory: E:\LakeTanaData\Outputs\NDVI

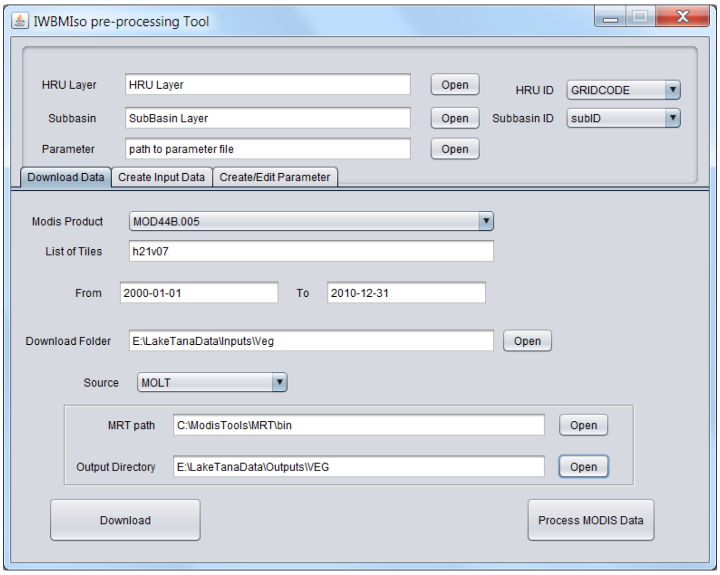
With MRT installed enter the path to the bin directory of the MRT installation. In the example above, MRT is installed in C:\ModisTools. Then enter the path for the output directory and click Process MODIS Data. The resulting output will be a tif file for each month of data downloaded. The tif files should be stored in the subdirectory NDVI under the IWBMIsoSourceData directory for further processing in eRAMS to generate the HRU NDVI data file.

10.2. Get MODIS VCF Source Data for the Project Region

The pre-processing tool provides a way to download **MODIS MOD44B.005 – Vegetation Continuous Fields Yearly L3 Global 250m (MODIS VCF)** for the project. Refer to https://lpdaac.usgs.gov/products/modis_products_table/mod44b for additional information about this data.

The Data Download tab provides a tool to download monthly NDVI files from MODIS Terra data pool at http://e4ftl01.cr.usgs.gov/MOLT/MOD44B.005/.   The correct data requires knowing the ID of the MODIS tile(s) for the area. Use the USGS Global Visualization Viewer (http://glovis.usgs.gov/) to obtain that information. Use the same procedure as described for the MODIS NDVI data described above to obtain the ID of the tile(s) for the basin area.

The data files are annual estimates of percent tree cover for each 250m pixel. The period of record for the MOD44B.05 can be found on the information page (https://lpdaac.usgs.gov/products/modis_products_table/mod44b). The current period available is 2000-2010.



HRU Layer: HRU Layer

HRU ID: GRIDCODE

Subbasin: SubBasin Layer

Subbasin ID: subID

Parameter: path to parameter file

Modis Product: MOD44B.005

List of Tiles: h21v07

From: 2000-01-01 To: 2010-12-31

Download Folder: E:\LakeTanaData\Inputs\Veg

Source: MOLT

MRT path: C:\ModisTools\MRT\bin

Output Directory: E:\LakeTanaData\Outputs\VEG

To download the MODIS VCF files, select MOD44B.005 in the MODIS Product field, enter the tile ID(s) in the List of Tiles field, the period of record to be downloaded in the From-To field, and the path to the output directory in the Download Folder field. Then select MOLT for the MODIS Terra data pool. Then click Download. When the download is complete, enter the path to the MRT bin directory and then the path to the output directory to store the processed VCF files. The output will be a tif file for each year that was downloaded. The tif files should be stored in the sub directory directory VEG under the IWBMIsoSourceData directory for further processing in eRAMS to generate the HRU cover density estimate for each HRU and these values will be entered into the IWBMIso parameter file.

10.3. Get Precipitation Isotope Data for the Project Region

Precipitation isotope data are required for each HRU for each month of the year. The data are organized in two files, one for O18 and the second for H2. Each file can be a continuous file for each month in the period of record if such data are available or each file can contain a single 12-month record of average isotope values for each HRU. This 12-month record will be used for each year of the simulation.

Various sources of gridded average monthly O18 and H2 are available. To use the IWBMIso application HRU isotope generation tool, the monthly fields of O18 and H2 must be in a tif file format. An O18 directory should be created that contains the 12 monthly tif files for O18, and similarly an H2 directory containing the monthly tif files for H2 should also be created. These directories will be placed in the appropriate regions directory for further processing by the IWBMIso application to generate the precipitation isotope values for each HRU for the single 12-month period file that will be used as input for the simulation period.

One source of gridded global precipitation isotope data is waterisotopes.org (http://wateriso.utah.edu/waterisotopes/pages/data_access/da_main.html). Gridded data sets for O18 and H2 are available at various resolutions for most land areas of the globe as georeferenced ArcGIS raster coverages. The grid files are distributed as grid files stored in a .mdb geodatabase. There are 12 monthly files for O18 and for H2. One way of converting these files is to use ArcCatalog and export the grid files to a tif format. The 12 O18 files should be placed in an O18 subfolder and the 12 H2 files should be placed in an H2 subfolder. These subdirectories should be placed in the appropriate regions directory. User the LakeTana region folder as a reference for understanding the proper location and format of the isotope data.

10.4. Get Meteorological Source Data for Your Project Region

Although the IWBMIso application contains the global CRU climate dataset needed by the model, following is the process for downloading this data from the Climatic Research Unit (CRU) at the University of East Anglia in the UK (http://catalogue.ceda.ac.uk/uuid/3f8944800cc48e1cbc29a5ee12d8542d). The CRU TS (time-series) datasets are month-by-month variation in climate over the last century or so. These are on high-resolution grids at a 0.5 deg by 0.5 deg. In the CRU TS 3.1 dataset currently being used, the variables available include cloud cover, diurnal temperature range, PET, daily mean temperature, monthly average daily minimun/maximum temperature, and vapour pressure for the period 1901-2009.

CRU TS3.0 data were originally produced and held by the CRU, however in 2007 it was decided that the BADC could provide dedicated long term support for these data and work began to port the software to the BADC systems where they could run on an operational basis. The system has now been ported to the BADC, and the BADC are responsible for updating these data on a regular (6 monthly) basis.

At present, the British Atmospheric Data Centre (BADC) holds the latest Time Series data generated by CRU for the period 1901-2012. Those are available as CRU TS 3.21 data. The BADC also holds the preliminary CRU TS3.00 datasets for the period 1901-2006 as well as the subsequent CRU TS 3.10 and CRU TS 3.20 datasets for the periods 1901-2009 and 1901-2011 respectively.

To download data, go to http://badc.nerc.ac.uk/browse/badc/cru. Register as a BADC user and establish an account. Then access the CRU TS data sets. To use one of the CRU TS datasets in the model, download the pre, pet, tmn, tmx, and vap data files. Select the ASCII format data files and download in zipped format. If selecting CRU_ts_3.10 dataset, then download the files

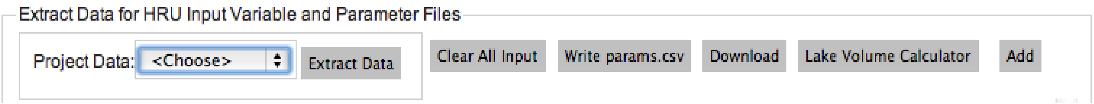
```
cru_ts_3_10_01.1901.2009.pre.dat.gz
cru_ts_3_10.1901.2012.pet.dat.gz
cru_ts_3_10.1901.2012.tmn.dat.gz
cru_ts_3_10.1901.2012.tmx.dat.gz
cru_ts_3_10.1901.2012.vap.dat.gz.
```

Create a CRU subdirectory in the IWBMIso Source Data directory and place these 5 zip files in that directory for use by eRAMS to generate the HRU based timeseries input data files for the model.

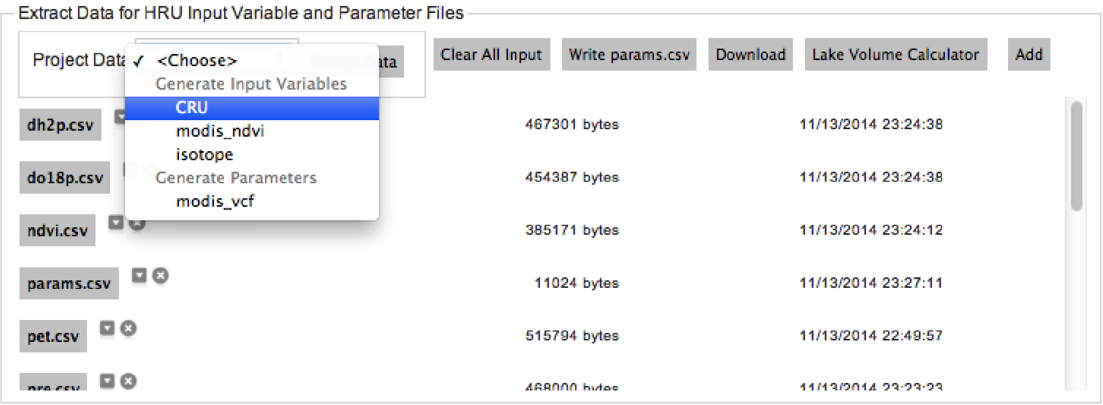
11. Generate Input – Create Input Variable and Parameter Files

This menu item is used to create or add the input variable data files necessary to run the model. In order to generate these files the user has the responsibility to create a set of source data files that will be used along with the HRU data layer created in the Create/Edit Basin step to generate time series inputs for each HRU.

The data extraction step computes the time series values of each input variable for each HRU. Clicking **Generate Input** item in the IWBMIso Tools menu will open the panel at the bottom of the browser. (Note: the **Generate Input** menu item is not active unless a scenario has been selected). Clicking the **Project Data** pull down menu lists the data elements to be generated for each HRU. Selecting a data type and then clicking on **Extract Data** will execute the data extraction process for that data element.



If selecting CRU for the data extraction process, a total of 5 input files will be created providing the input meteorological variables of precipitation, max and min temperature, potential ET, and relative humidity. Each file is in .csv format. Clicking on the file name provides the ability to view the data in tabular and graphical format.



dh2p.csv	467301 bytes	11/13/2014 23:24:38
do18p.csv	454387 bytes	11/13/2014 23:24:38
ndvi.csv	385171 bytes	11/13/2014 23:24:12
params.csv	11024 bytes	11/13/2014 23:27:11
pet.csv	515794 bytes	11/13/2014 22:49:57
pre.csv	468000 bytes	11/13/2014 23:23:23

When extracting CRU data, temporary files may display in the box below, but when finished the process has created and displayed pre.csv (precipitation), tmx.csv (maximum temperature), tmn.csv (minimum temperature), pet.csv (potential evapotranspiration), and rhn.csv (relative humidity). These CRU-based files are monthly values from 1901 to 2012 for each HRU in your scenario. Note: Extracting and processing CRU data can take several minutes, in part depending on the number of HRUs in your scenario.

Click on any of the generated files to view the data in table format. Change the value in a cell if deemed necessary. Batch edit functionality is not yet fully available.

Click on one or more columns in the table, and then click **Generate Graph** to visualize in graph format.

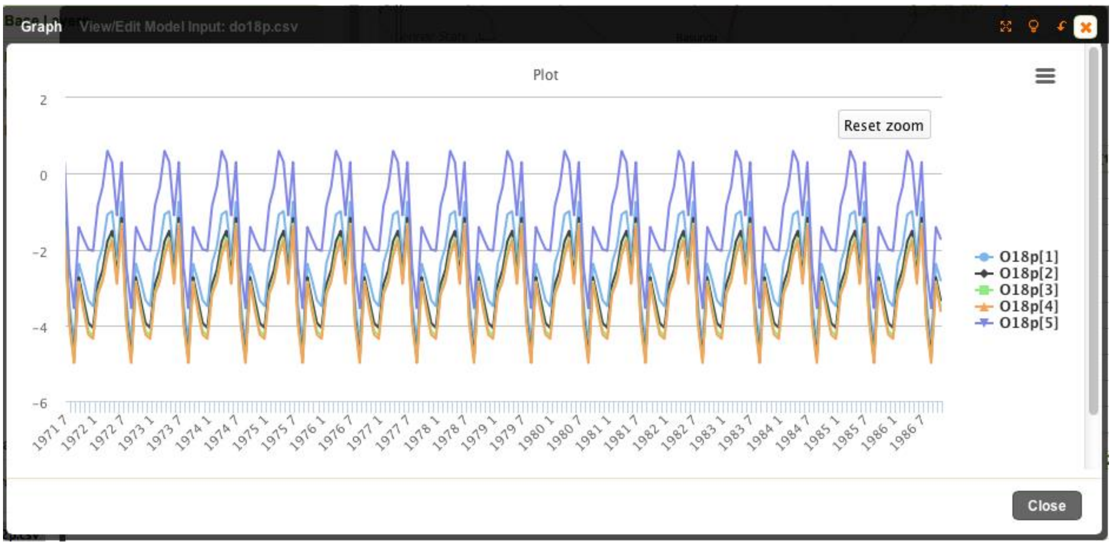
Choose **modis_ndvi** in the **Project Data** droplist, and then click **Extract Data**. The application will generate the NDVI_tif_avg.csv and NDVI_tif.csv input variable files.

Later when running the model, the user can choose which file to use in the simulation. The MODIS-based NDVI_tif.csv file contains data for the past decade and not before. To match the model simulation period, which may be several decades, during Run Model the application will use the data in a repeated fashion to fill out the period. If deciding to use NDVI_tif_avg.csv, which is one year (12 months) file, Run Model will apply the same repeat process to match the simulation period.

Extracting MODIS NDVI takes a few seconds. Open the file to view the data in table and graph format. Also edit as needed.

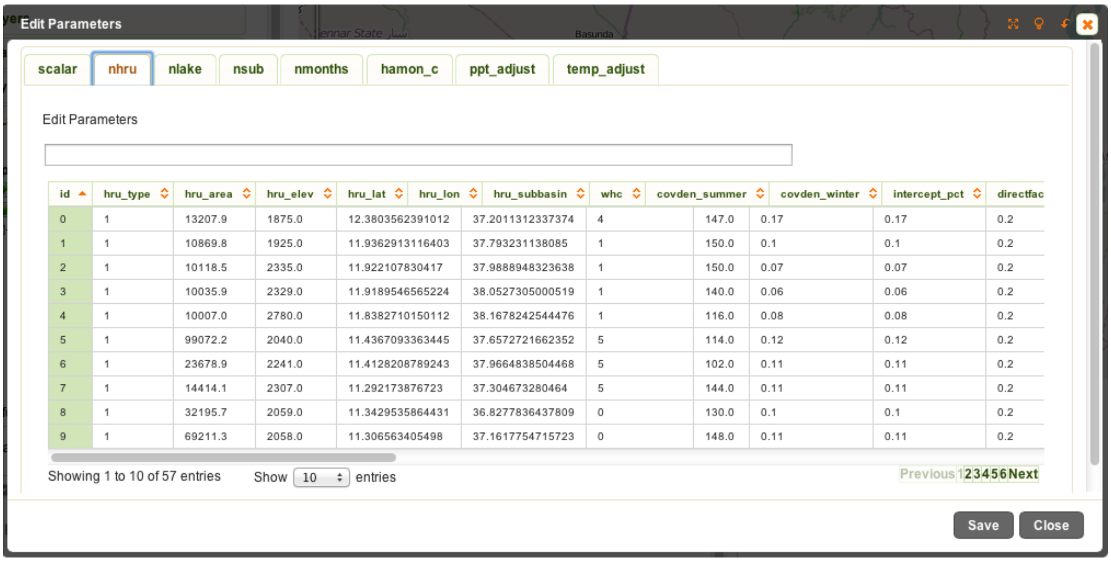
Choose **isotope** in the **Project Data** droplist and then click **Extract Data**. The application generates the H2p_tif.csv and O18p_tif.csv input variable files. These are annual monthly files containing isotope values (permils) for precipitation by HRU. The model simulation will use these tables in a repeatable fashion to match the simulation period.

The extract process takes a few seconds. Open the file to view the data in table and graph format. Also edit as needed.



Choose **modis_vcf** in the **Project Data** droplist and then click **Extract Data**. Then click **Write params.csv**. The application extracts, derives, and averages monthly cover density by HRU. The **Write params.csv** process inserts a summer and winter cover density in **params.csv**.

Note: Previously with creation of the HRU layer, a template **params.csv** for the scenario (layer) was generated in the background. When the template is created a number of the spatial and temperal parameters are computed and written in to the template. These parameters include hru_type, hru_area, hru_elev, hru_lat, hru_lon, hru_subbasin, subbasin_type, and subbasin_down. The application also automatically extracts and generates available water holding capacity from FAO Soils for each HRU in **params.csv**. All other model parameters are the responsibility of the user to provide. The template file has only default values for these parameters and these values are probably not appropriate for your model application. Use the Edit Parameters tool to make the necessary changes for your basin. It is also a good idea to check the values of the parameters that were computed automatically to insure that they are correct as well.



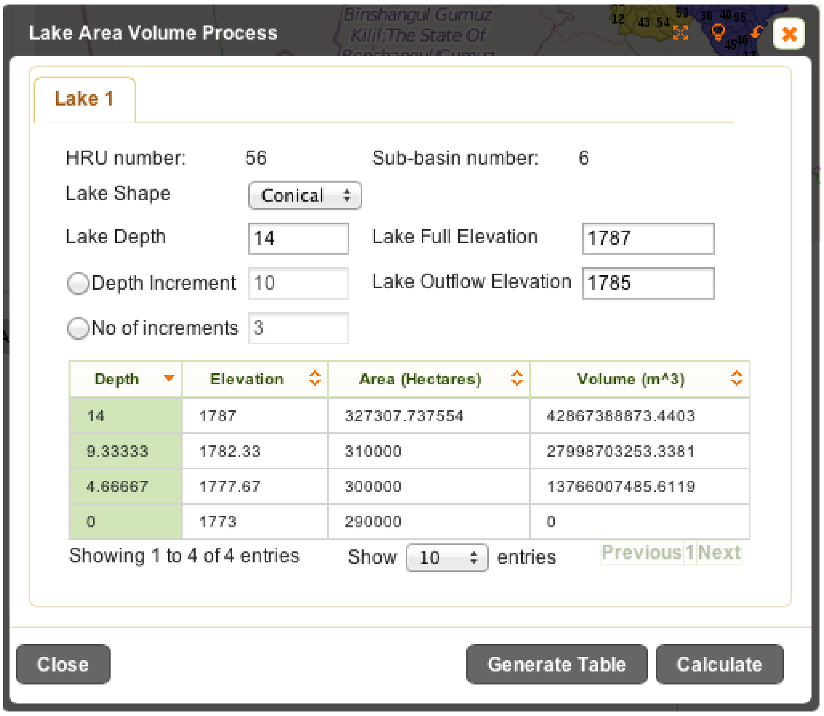
Edit Parameters

id	hru_type	hru_area	hru_elev	hru_lat	hru_lon	hru_subbasin	whc	covden_summer	covden_winter	intercept_pct	directfac
0	1	13207.9	1875.0	12.3803562391012		37.2011312337374	4	147.0	0.17	0.17	0.2
1	1	10869.8	1925.0	11.9362913116403		37.793231138085	1	150.0	0.1	0.1	0.2
2	1	10118.5	2335.0	11.922107830417		37.9888948323638	1	150.0	0.07	0.07	0.2
3	1	10035.9	2329.0	11.9189546565224		38.0527305000519	1	140.0	0.06	0.06	0.2
4	1	10007.0	2780.0	11.8382710150112		38.1678242544476	1	116.0	0.08	0.08	0.2
5	1	99072.2	2040.0	11.4367093363445		37.6572721662352	5	114.0	0.12	0.12	0.2
6	1	23678.9	2241.0	11.4128208789243		37.9664838504468	5	102.0	0.11	0.11	0.2
7	1	14414.1	2307.0	11.292173876723		37.304673280464	5	144.0	0.11	0.11	0.2
8	1	32195.7	2059.0	11.3429535864431		36.8277836437809	0	130.0	0.1	0.1	0.2
9	1	69211.3	2058.0	11.306563405498		37.1617754715723	0	148.0	0.11	0.11	0.2

Showing 1 to 10 of 57 entries Show 10 entries Previous 1 2 3 4 5 6 Next

The extract process for modis_vcf and the **Write params.csv** process each take a few seconds. Open params.csv to view and edit the data.

If the scenario has lakes, to further populate the lake table in the parameter file click the **Lake Volume Calculator** tool to open the dialog for computing lake area and volume coefficients.



HRU number: 56 Sub-basin number: 6

Lake Shape: Conical

Lake Depth: 14 Lake Full Elevation: 1787

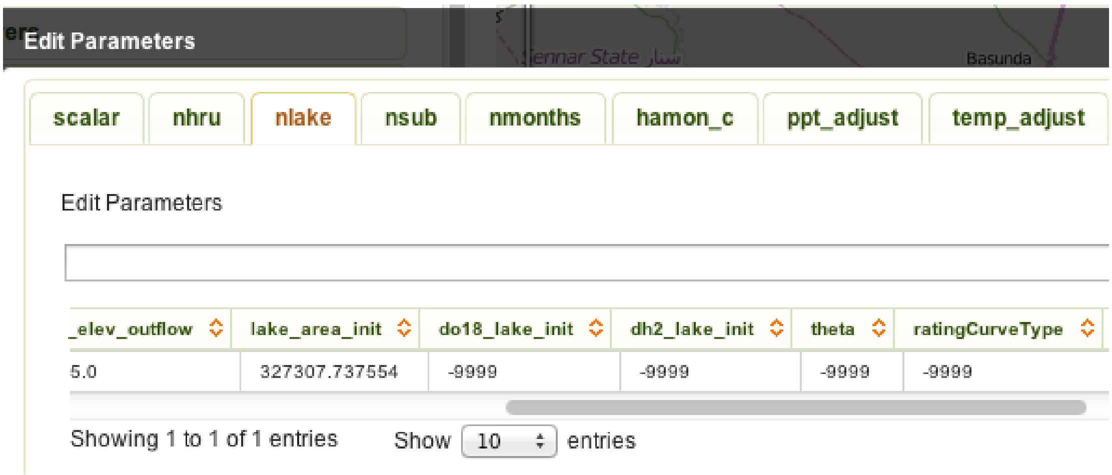
○ Depth Increment: 10 Lake Outflow Elevation: 1785

○ No of increments: 3

Depth	Elevation	Area (Hectares)	Volume (m^3)
14	1787	327307.737554	42867388873.4403
9.33333	1782.33	310000	27998703253.3381
4.66667	1777.67	300000	13766007485.6119
0	1773	290000	0

Showing 1 to 4 of 4 entries Show 10 entries Previous 1 Next

First define the area of the lake at depth intervals. Specify a lake depth, lake full elevation, lake outflow elevation, and whether to generate the lake table by depth increment or number of increments. Click **Generate Table**, followed by entering the area at each interval. Then click **Calculate**, which computes volume for each interval and area and volume coefficients, which the application inserts into the lake table of **params.csv**.



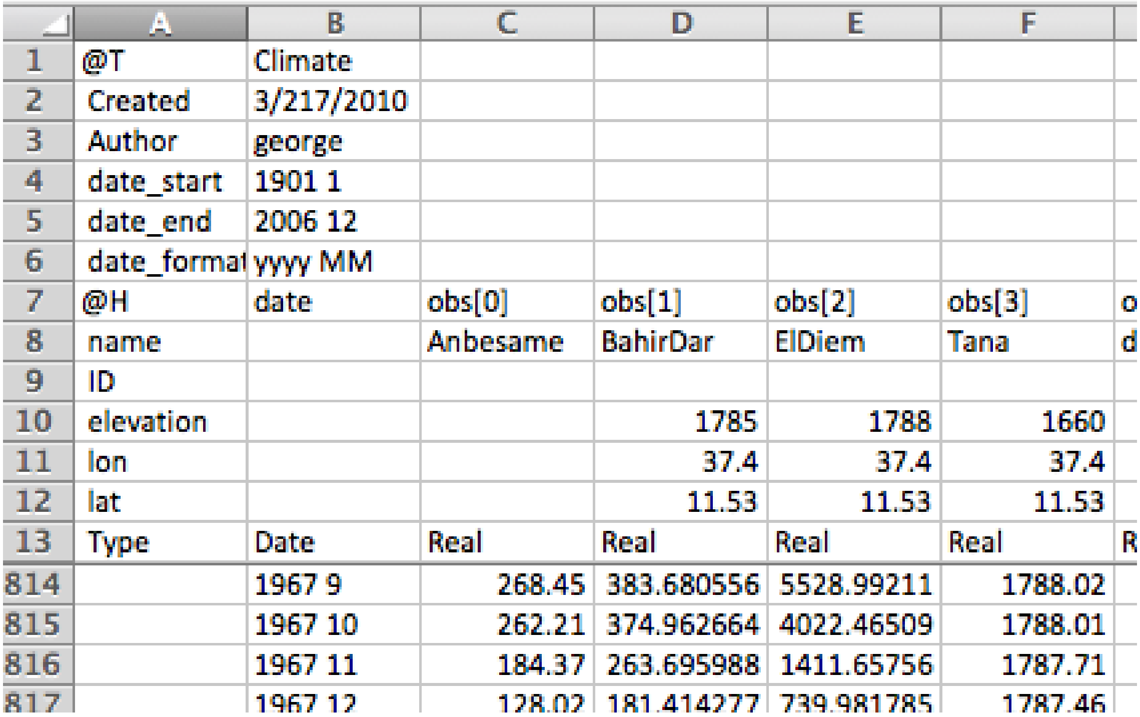
Edit Parameters

_elev_outflow	lake_area_init	do18_lake_init	dh2_lake_init	theta	ratingCurveType
5.0	327307.737554	-9999	-9999	-9999	-9999

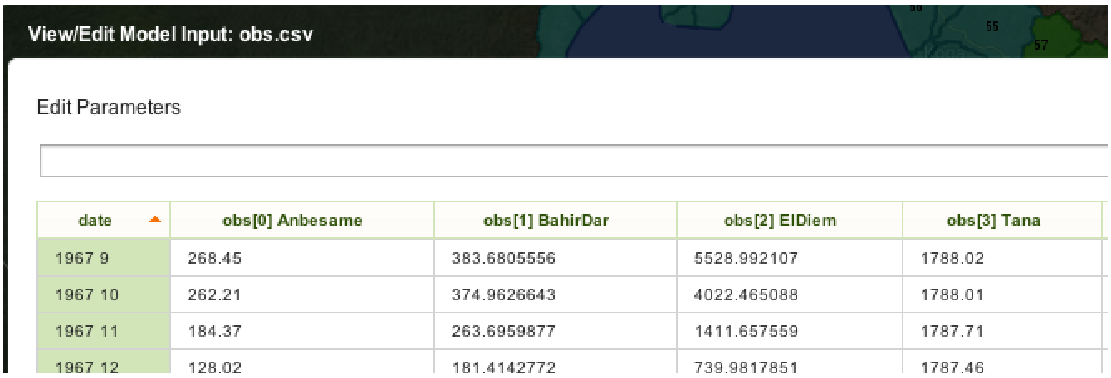
Showing 1 to 1 of 1 entries   Show 10 entries

The user also must enter a few more parameter values into the lake table, those cells populated with -9999 (which means missing data). If not in hand, consult your isotope hydrologist advisor or technical references to obtain values for these parameters

The next step is to add observed data compiled in and named as an **obs.csv** file. Make sure it has correct OMS .csv formatting. Click the **Add** button to upload this file. OMS .csv table formatting is shown below, with @T meaning table and @H meaning header. The application reads the .csv file, which can be opened and contents displayed.



	A	B	C	D	E	F	
1	@T	Climate					
2	Created	3/217/2010					
3	Author	george					
4	date_start	1901 1					
5	date_end	2006 12					
6	date_format	yyyy MM					
7	@H	date	obs[0]	obs[1]	obs[2]	obs[3]	o
8	name		Anbesame	BahirDar	ElDiem	Tana	d
9	ID						
10	elevation			1785	1788	1660	
11	lon			37.4	37.4	37.4	
12	lat			11.53	11.53	11.53	
13	Type	Date	Real	Real	Real	Real	R
814		1967 9	268.45	383.680556	5528.99211	1788.02	
815		1967 10	262.21	374.962664	4022.46509	1788.01	
816		1967 11	184.37	263.695988	1411.65756	1787.71	
817		1967 12	128.02	181.414277	739.981785	1787.46	



View/Edit Model Input: obs.csv

Edit Parameters

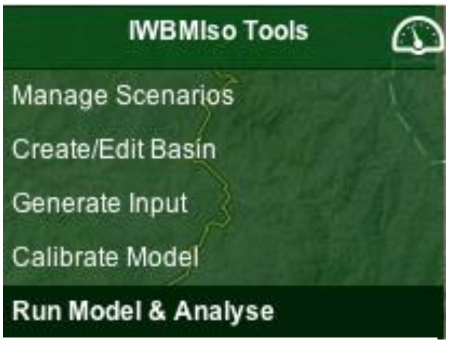
date	obs[0] Anbesame	obs[1] BahirDar	obs[2] ElDiem	obs[3] Tana
1967 9	268.45	383.6805556	5528.992107	1788.02
1967 10	262.21	374.9626643	4022.465088	1788.01
1967 11	184.37	263.6959877	1411.657559	1787.71
1967 12	128.02	181.4142772	739.9817851	1787.46

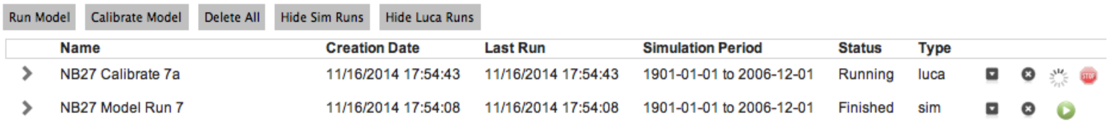
In lieu of extracting and pre-processing source data in **Generate Input**, use the **Add** button to include input variable and parameter files created externally, as long as they are corrected formatted and named. With IWBMIso 1.0 the names of these files have been standardized and cannot vary. The application will be enhanced in a future version to allow selection of input variable and parameter files having names different than the convention.

Adding the **obs.csv** completes the **Generate Input** step, providing all input files required to run the model.

12. Run Model and Analyse

Click **Run Model & Analyze** in the IWBMIso menu to display the simulation panel at the bottom of the browser. **Calibrate Model** also opens to the same bottom panel. Previous model and calibration simulations are listed in the panel and can be re-visited.

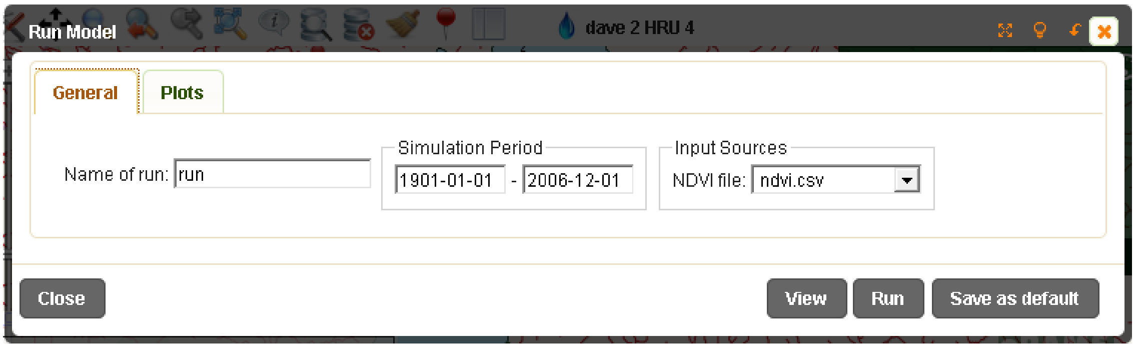




Name	Creation Date	Last Run	Simulation Period	Status	Type
NB27 Calibrate 7a	11/16/2014 17:54:43	11/16/2014 17:54:43	1901-01-01 to 2006-12-01	Running	luca
NB27 Model Run 7	11/16/2014 17:54:08	11/16/2014 17:54:08	1901-01-01 to 2006-12-01	Finished	sim

Click the **Run Model** button to open the dialog to configure a model simulation. Name the simulation and choose which NDVI file to use. Changing the default simulation period, however, requires the period fall within the date ranges for your precipitation data (pre.csv).

In future versions, the user will be able to select choices for each of the input variable categories and parameter file.

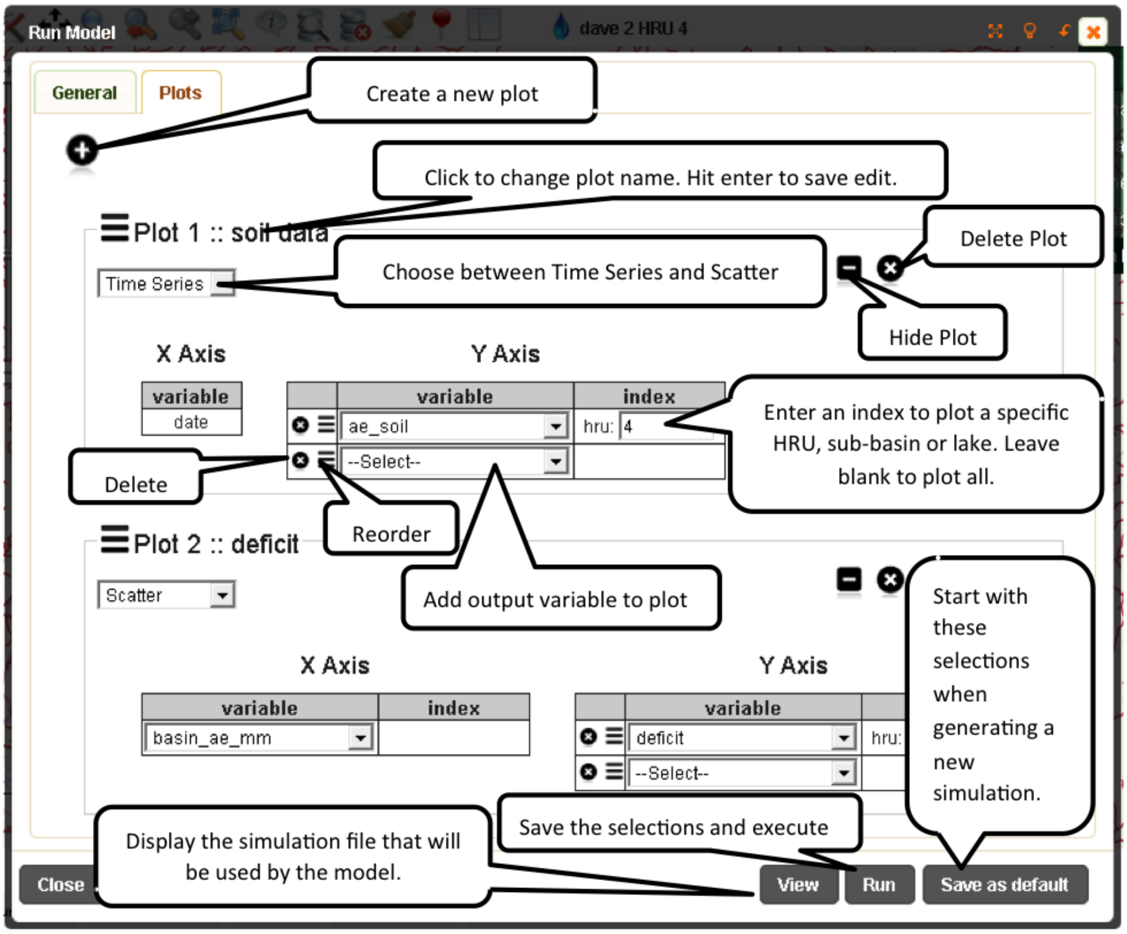


Name of run: run

Simulation Period: 1901-01-01 - 2006-12-01

Input Sources: NDVI file: ndvi.csv

Next, click on the **Plots** tab. Follow the instructions in the graphic below to specify the output to be plotted. Running the simulation also will generate four output tables, also available for graphing.



Create a new plot

Click to change plot name. Hit enter to save edit.

Delete Plot

Choose between Time Series and Scatter

Hide Plot

Enter an index to plot a specific HRU, sub-basin or lake. Leave blank to plot all.

Delete

Reorder

Add output variable to plot

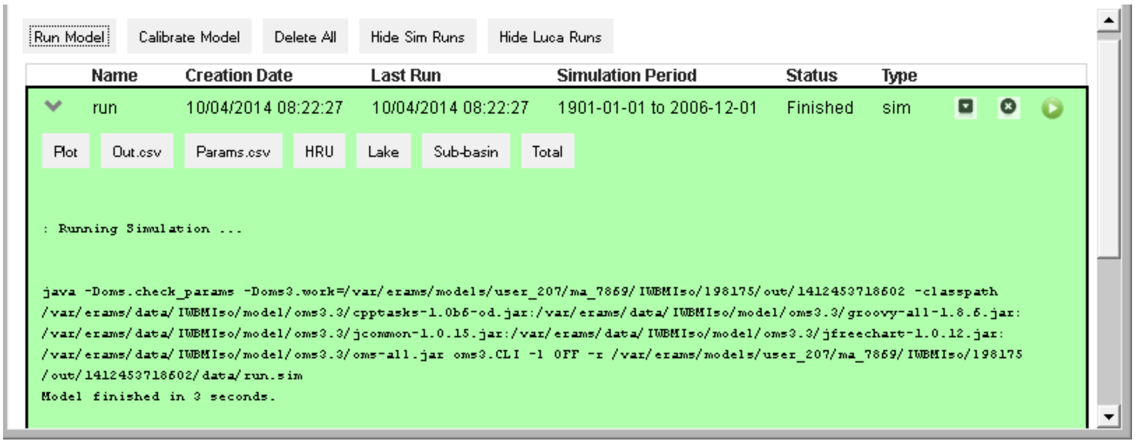
Start with these selections when generating a new simulation.

Save the selections and execute

Display the simulation file that will be used by the model.

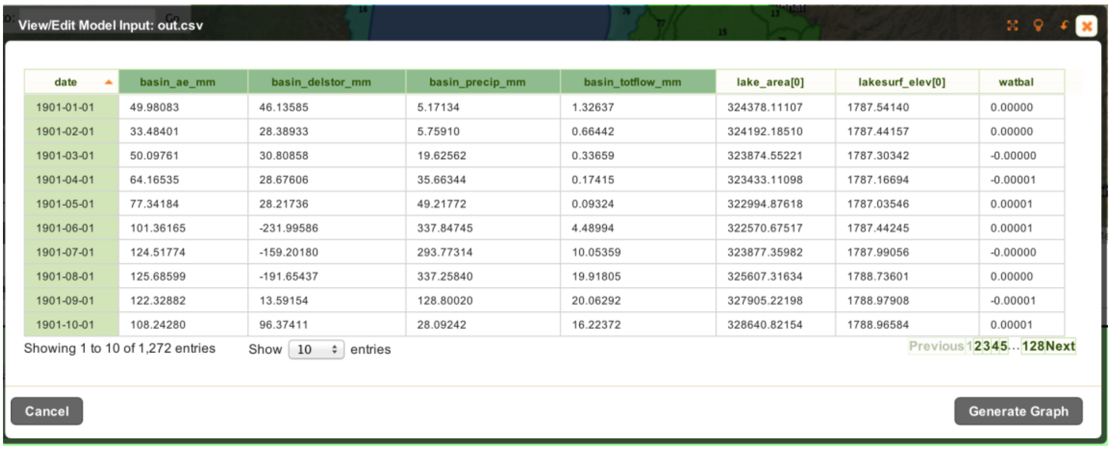
Click **Run** and the application will run the model and when finished display buttons for the output. Click **Plot** to view the graph of the specified output variables. Open **out.csv** to view data and also create graphs. Click on **params.csv** to view the parameters used in the simulation. Open and view standard **HRU**, **Lake**, and **Sub-**

Basin output tables, which also are available for graphing. Simulation progress also is logged and displayed in the green pane at the bottom.



Name	Creation Date	Last Run	Simulation Period	Status	Type
run	10/04/2014 08:22:27	10/04/2014 08:22:27	1901-01-01 to 2006-12-01	Finished	sim

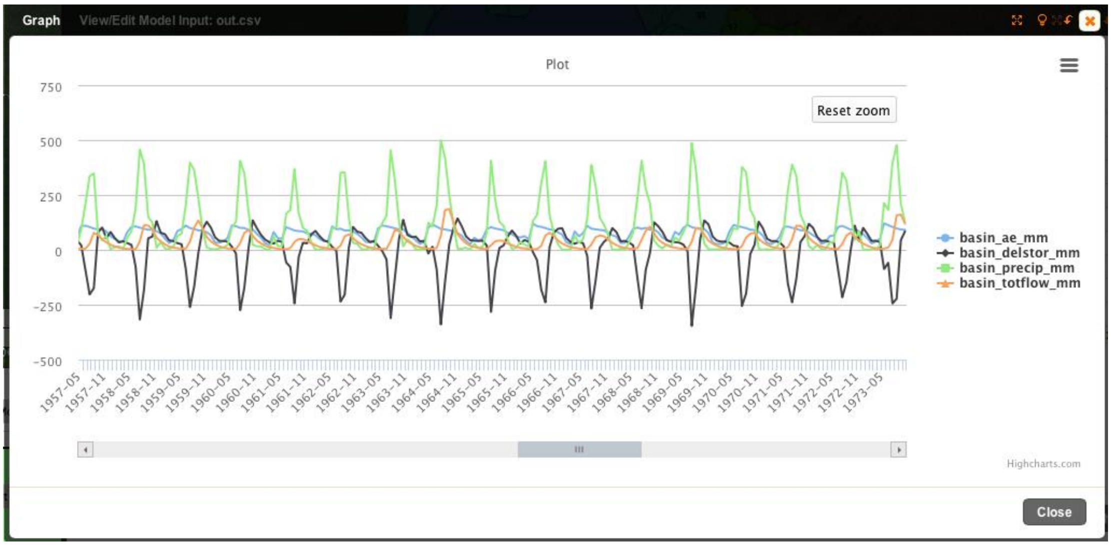
```
: Running Simulation ...


java -Doms.check_params -Doms3.work=/var/erams/models/user_207/ma_7869/IWBMIso/198175/out/1412453718602 -classpath
/var/erams/data/IWBMIso/model/oms3.3/cpptasks-1.0b6-od.jar:/var/erams/data/IWBMIso/model/oms3.3/groovy-all-1.8.6.jar:
/var/erams/data/IWBMIso/model/oms3.3/jcommon-1.0.15.jar:/var/erams/data/IWBMIso/model/oms3.3/jfreechart-1.0.12.jar:
/var/erams/data/IWBMIso/model/oms3.3/oms-all.jar oms3.CLI -l OFF -r /var/erams/models/user_207/ma_7869/IWBMIso/198175
/out/1412453718602/data/run.sim
Model finished in 3 seconds.
```



View/Edit Model Input: out.csv

date	basin_ae_mm	basin_delstor_mm	basin_precip_mm	basin_totflow_mm	lake_area[0]	lakesurf_elev[0]	watbal
1901-01-01	49.98083	46.13585	5.17134	1.32637	324378.11107	1787.54140	0.00000
1901-02-01	33.48401	28.38933	5.75910	0.66442	324192.18510	1787.44157	0.00000
1901-03-01	50.09761	30.80858	19.62562	0.33659	323874.55221	1787.30342	-0.00000
1901-04-01	64.16535	28.67606	35.66344	0.17415	323433.11098	1787.16694	-0.00001
1901-05-01	77.34184	28.21736	49.21772	0.09324	322994.87618	1787.03546	0.00001
1901-06-01	101.36165	-231.99586	337.84745	4.48994	322570.67517	1787.44245	0.00001
1901-07-01	124.51774	-159.20180	293.77314	10.05359	323877.35982	1787.99056	-0.00000
1901-08-01	125.68599	-191.65437	337.25840	19.91805	325607.31634	1788.73601	0.00000
1901-09-01	122.32882	13.59154	128.80020	20.06292	327905.22198	1788.97908	-0.00001
1901-10-01	108.24280	96.37411	28.09242	16.22372	328640.82154	1788.96584	0.00001

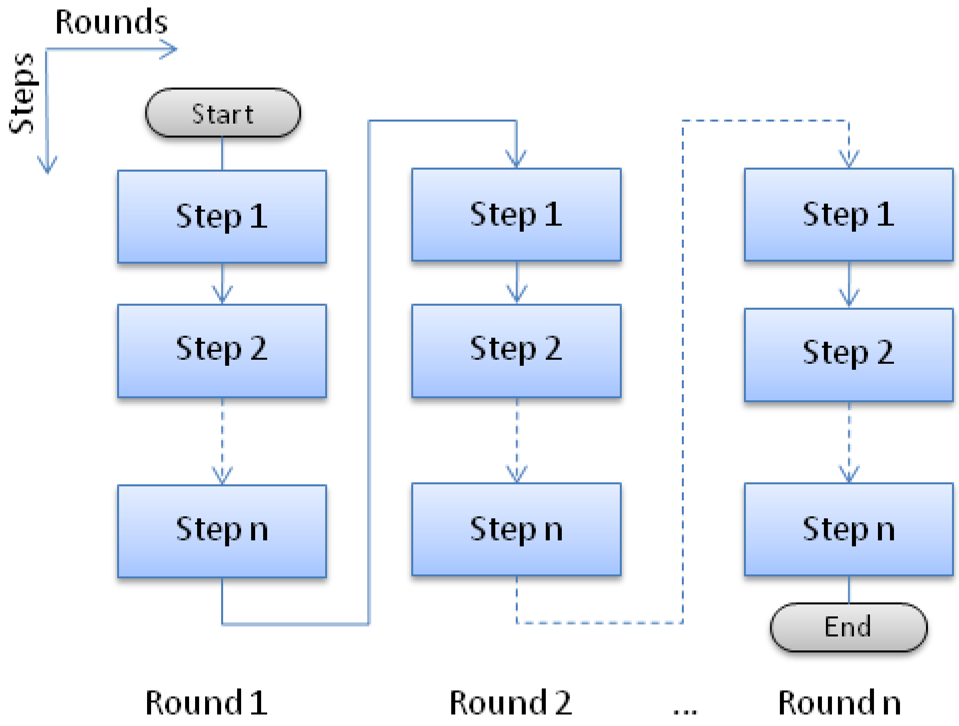
Showing 1 to 10 of 1,272 entries Show 10 entries Previous 12345…128Next



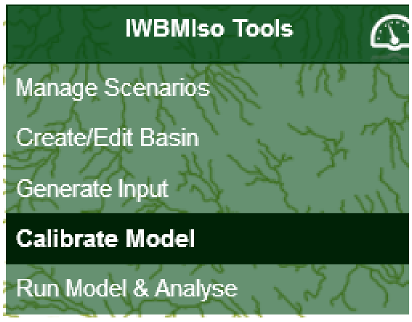
13. Calibrate Model

The calibration toolset used for IWBMIso is Luca (Let us calibrate) (Hay and Umemoto, 2006). As implemented in IWBMIso, Luca provides a user-friendly graphical user interface (GUI) that provides a systematic way of building and executing a calibration procedure. The calibration procedure uses the Shuffled Complex Evolution global search algorithm (Duan and others, 1994) to calibrate user selected model parameters.

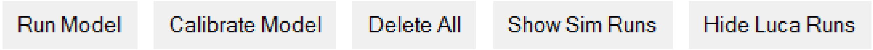
The Luca framework is based on the concepts of steps and rounds (see Figure below). A step is composed of a user-defined set of parameters to be calibrated using one or more user-selected objective functions. A round is the sequential computation of all defined steps. The calibration proceeds one step at a time. At the completion of a step, the calibrated values of the parameters selected in that step are written to the working parameter file and passed into the next step. This sequence of passing the working parameter file to the next step is repeated until all steps are executed. The sequence of calibration steps is then repeated for the designated number of rounds.



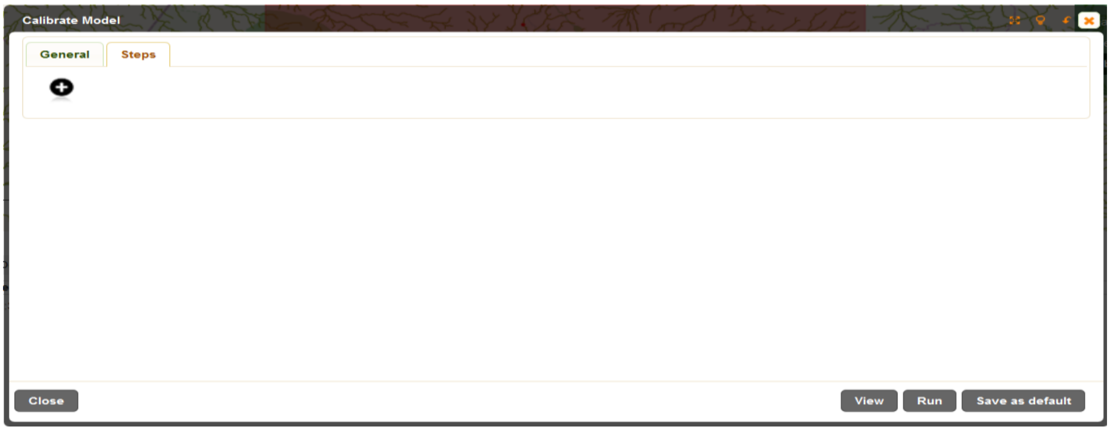
To begin a calibration the user selects **Calibrate Model** from the IWBMIso Tools slideout menu.



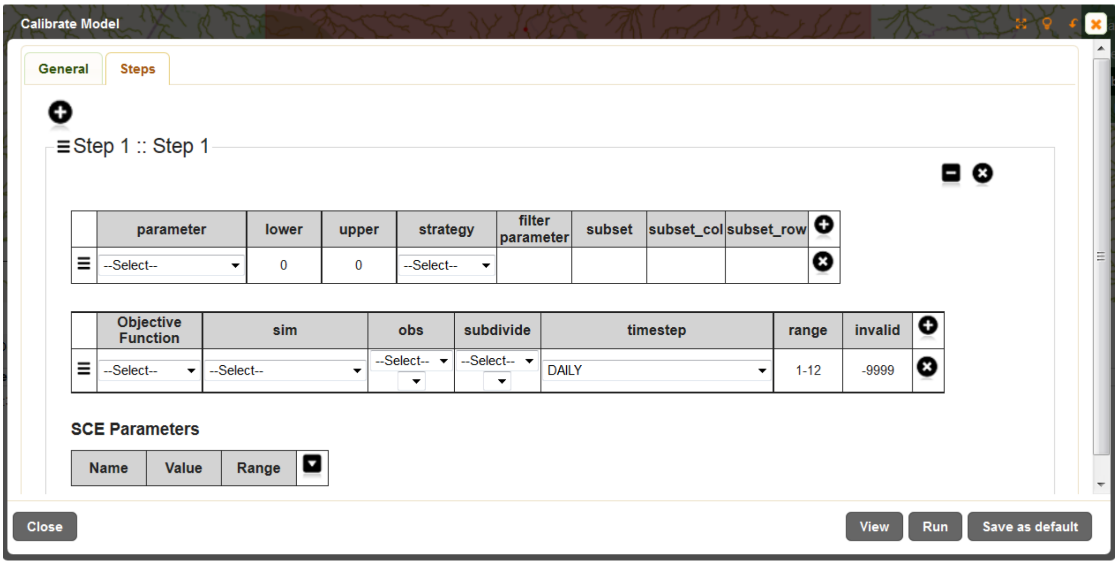
The user is then presented with the following choices



Clicking on **Calibrate Model** then presents the following screen.



Clicking on the + sign will open a menu to be completed in which the user will enter or select the parameter and objective function elements provided using pull down menus and data entry fields.



☰Step 1 :: Step 1

	parameter	lower	upper	strategy	filter parameter	subset	subset_col	subset_row	⊕
☰	--Select--	0	0	--Select--					⊗

	Objective Function	sim	obs	subdivide	timestep	range	invalid	⊕
☰	--Select--	--Select--	--Select--	--Select--	DAILY	1-12	-9999	⊗

SCE Parameters

Name	Value	Range

The first element in the step is the **parameter** element. If more than one parameter is to be calibrated in this step, clicking on the + sign at the end of the element will open additional parameter elements. For each parameter, a set of properties are available to specify details of the calibration procedure for each selected parameter.

The table below describes the sub-elements contained in the parameter element of a step.

Name	Description	Type	Required
`lower`	the lower boundary	double	y
`upper`	the upper boundary	double	y
`calib_strategy`	the calibration strategy	`MEAN` | `INDIVIDUAL`| `BINARY`	n (default:`MEAN`)
`filter_param`	The name of the filter parameter for calibration selection	string	n (default: match subset to parameter value index instead of this filterParam value)
`subset`	selects which subset of params are to be used for calibration	String ("m-n,p,...")	n. (default: include all parameter's values for calibration)
subset_col	Used as the column subset for 2D array tables	String	n
subset_row	Used as the row subset for 2D array tables	String	n

The *calib_strategy* has one of three possible approaches that can be selected.

MEAN: Optimize the parameter based on the mean value of all the parameter instances. The instances to be used can be constrained by the filter_param and subset options described below. When the **mean** is chosen, the mean parameter value (instead of each individual parameter value) is calibrated. Each time SCE generates a value for the mean, individual parameter values are generated based on the new mean such that the mean-value distribution is preserved. This option is a good choice when a spatially distributed parameter is chosen for calibration.

INDIVIDUAL: Optimize each parameter instance independently. The instances to be used can be constrained by the filter_param and subset options described below. The INDIVIDUAL option is typically used when there are external measures available against which to compare model computed results for each individual parameter value. An example of this is the calibration of the monthly parameter *hamon_c* in the Hamon potential evapotranspiration (PET) equation. The user can specify an external file with measured or estimated mean monthly PET that can be used with the simulated mean monthly PET for each month to compute the selected objective function.

BINARY: Treat each individual parameter instance as a binary value. The BINARY option is used only for those parameters that consist of values of 0 or1. An example might be a model that uses 0 and 1 to specify a subset of temperature of precipitation stations to use in the computation of distributed temperature and precipitation in a basin. All or a subset of individual values of a binary parameter can be selected using the range element.

filter_param and ***subset*** can be used to select which of the spatially or temporally distributed parameters are to be used in calibration. The user can specify neither of these, *subset* only, or both. If neither of these is included, then all parameter values will be included in calibration. If only *subset* is specified without *filter_param*, then only parameter values with indexes matching any value in the *subset* will be included in calibration. If both *filter_param* and *subset* are specified, then parameter values for which the corresponding *filter_param* value is equal to any value in the subset will be included in calibration

For 2D parameters, use *subset_col* and *subset_row* in place of subset in order to specify the desired columns and rows. If *subset* is specified, it will act as *subset_col*. *subset* selection for 2D arrays using a *filter_param* is untested, and so is not supported.

The **Objective Function** element specifies the mathematical form of the objective function, the time period over which the objective function is computed, and the simulated (sim) and observed (obs) variables to be used in the objective function, and the path to the variable locations. Luca also supports the use of multiple objective functions for a step. The **Objective Function** element properties as shown in the table below.

Name	Description	Type	Required
`Objective Function`	the objective function	See objective function table below	yes
`sim`	the simulated variable	String	yes
`obs`	the observed variable	String	yes
`subdivide`	subdivide file containing information about which dates to use for this objective function.		no
`timestep`	the time step for simulated and observed values	`RAW | TIME_STEP | DAILY | MONTHLY_MEAN | MEAN_MONTHLY | TANNUAL_MEAN | PERIOD_MEDIAN | PERIOD_MIN | PERIOD_MAX | PERIOD_STDEV`	n (default: `DAILY`)
`weight`	the objective function weight	double (`0 - 1.0`)	n 2)
`range`	Defines which months to include in analysis	String ("m-n,p,...") where m,n,p are 1-12 for Jan-Dec	n (default "`1-12`")
`invalid`	Value that indicates no data	`double`	Default = -9999

The **method** defines the mathematical function that will be used to compute the objective function. The methods available and their names are shown in the table below.

Objective Function methods

Name	Description	Equation
`ABSDIF`	Absolute difference	$\sum_{i=1}^{n} Q_{i,o} - Q_{i,s}$
`ABSDIFLOG`	Absolute difference Log	$\sum_{i=1}^{n} \lvert \ln Q_{i,o} - \ln Q_{i,s} \rvert$
`AVE`	Absolute Volume Error	$\lvert \sum_{i=1}^{n} Q_{i,s} - Q_{i,o} \rvert$
`IOA`	Index of Agreement	$1 - \frac{\sum_{i=1}^{n} \lvert Q_{i,o} - Q_{i,s} \rvert}{\sum_{i=1}^{n} \lvert Q_{i,s} - Q_o \rvert + \lvert Q_{i,o} - Q_o \rvert}$
`IOA2`	Index of Agreement (Pow 2)	$1 - \frac{\sum_{i=1}^{n} \left(Q_{i,o} - Q_{i,s}\right)^2}{\sum_{i=1}^{n} \left(Q_{i,s} - Q_o\right)^2 + \left(Q_{i,o} - Q_o\right)^2}$
`NS`	Nash-Sutcliffe	$1 - \frac{\sum_{i=1}^{n} \left(Q_{i,o} - Q_{i,s}\right)^2}{\sum_{i=1}^{n} \left(Q_{i,o} - Q_o\right)^2}$
`NSLOG`	Log of Nash-Sutcliffe	$1 - \frac{\sum_{i=1}^{n} \lvert \ln Q_{i,o} - \ln Q_{i,s} \rvert}{\sum_{i=1}^{n} \lvert \ln Q_{i,o} - \ln Q_o \rvert}$
`NS2LOG`	Log of Nash-Sutcliffe (Pow 2)	$1 - \frac{\sum_{i=1}^{n} \left(\ln Q_{i,o} - \ln Q_{i,s}\right)^2}{\sum_{i=1}^{n} \left(\ln Q_{i,o} - \ln Q_o\right)^2}$
`BIAS`	BIAS	$\frac{\sum_{i=1}^{n} \left(Q_{i,o} - Q_{i,s}\right)}{\sum_{i=1}^{n} Q_{i,o}}$
`PMCC`	Pearson product-moment correlation coefficient	$\frac{\sum_{i=1}^{n} \left(X_i - X\right)\left(Y_i - Y\right)}{\sqrt{\sum_{i=1}^{n} \left(X_i - X\right)^2} \sqrt{\sum_{i=1}^{n} \left(Y_i - Y\right)^2}}$
`RMSE`	Root Mean Square Error	$\sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(Q_s - Q_o\right)^2}$
`TRMSE`	Transformed Root Mean Square Error	$\sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(Z_s - Z_o\right)^2}$, $Z = \frac{\left(1+Q\right)^{0.3} - 1}{0.3}$

The **timestep** defines how the user-selected simulated and observed variable will be used to compute the objective function (OF). The available timesteps are

RAW: take data exactly as-is without applying the period-range or assuming any timescale. Timestamp on data is completely ignored. It is up to the user to ensure that the time of the Observed and Simulated data match line for line.

TIME_STEP: take data as-is but apply the period-range, without assuming any timescale. It is up to the user to ensure that the time of the Observed and Simulated data match line for line.

DAILY: a value for each day.

MEAN_MONTHLY: the arithmetic mean of the monthly means of a given month during a specified period of years. For example, the mean monthly value for January is the mean of all January mean values computed over the period of years used in the OF computation.

MONTHLY_MEAN: the arithmetic mean of values for a given month. For example, the monthly mean is computed as the sum of the means of all months in the period specified, divided by the total number of months in the period.

ANNUAL_MEAN: the mean of all data within each year. The annual period (i.e. year) starts in start_month_of_year.

PERIOD_MEDIAN: the median of all data within the period_range of months each year. The annual period (i.e. year) starts in start_month_of_year

PERIOD_MIN: the minimum of all data within the period_range of months each year. The annual period (i.e. year) starts in start_month_of_year

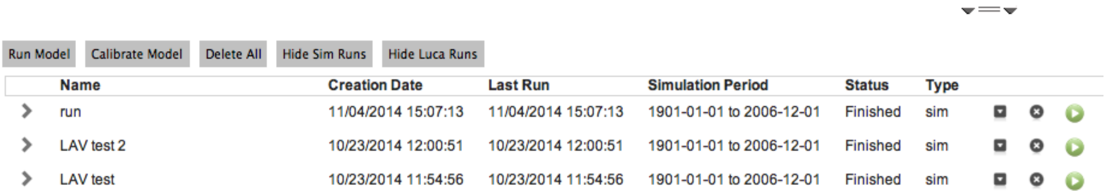
PERIOD_MAX: use the maximum of all data within the period_range of months each year. The annual period (i.e. year) starts in start_month_of_year

The **weight** is used when multiple objective functions are defined for a step. The user can assign a different weight to each objective function. The sum of the weights assigned must equal 1.0. If no weights are assigned by the user, equal weight will be given to each objective function.

The **period_range** allows the user to specify a period of months for the objective function computation. If the period selected is 1-5, then the objective function will be computed for the months of January through May. If the start_month_of_year is October (10) and the selected period of objective function computation is October through February, the period_range can be specified as (10-12, 1-2) or (1-2, 10-12). The order of the sequence of months is not critical.

A **subdivide_value** is used when an external file is provided to specify specific times when the associated objective function is to be computed. For example, if a user wants to calibrate the parameters in this step using only days below a specified streamflow rate (low flow period), then an external file can be prepared that contains a date and an integer value for each day within the data range. For example, in the external file, the value 1 could be assigned to specify all days that are below the user-selected streamflow value. All other days would be assigned an integer value different from 1. The objective function then will only be computed on days that the value in the external file equals the subdivide_value, in this example case a value of 1.

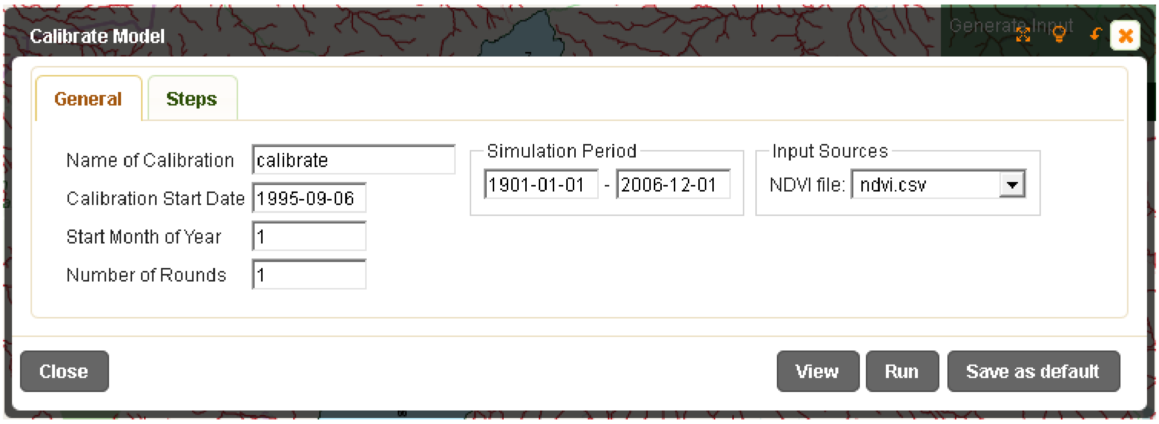
When clicking the **Calibrate Model** menu item to open the simulation panel at the bottom of the browser, this displays the same panel opened when clicking the **Run Model & Analyse** menu item, primarily because running and calibrating the model is an iterative process.



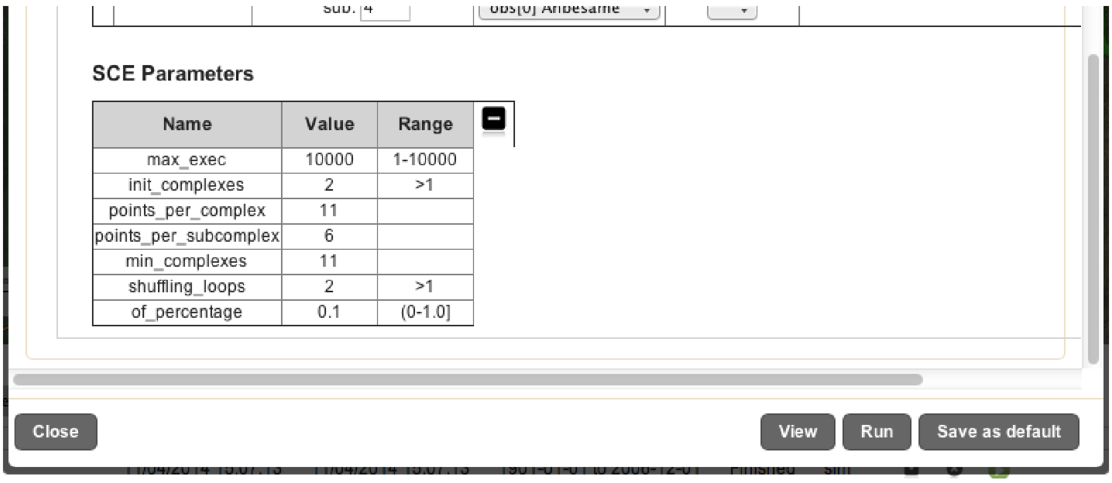
Name	Creation Date	Last Run	Simulation Period	Status	Type
run	11/04/2014 15:07:13	11/04/2014 15:07:13	1901-01-01 to 2006-12-01	Finished	sim
LAV test 2	10/23/2014 12:00:51	10/23/2014 12:00:51	1901-01-01 to 2006-12-01	Finished	sim
LAV test	10/23/2014 11:54:56	10/23/2014 11:54:56	1901-01-01 to 2006-12-01	Finished	sim

In most situations the user will run the model first without calibration, analyze the output, and then decide whether to calibrate certain model parameters. If deciding to calibrate, the user usually will complete several calibrate and run model cycles until achieving a good fit with observed data.

General inputs are similar to **Run Model** mode. New inputs are the start month of the year and the number of rounds that should be run. Then click on the **Steps** table. After completing the parameter and objective function tables for each round and step discussed previously at the beginning of this section, proceed to the **SCE Parameters** table.



Name of Calibration: calibrate
Calibration Start Date: 1995-09-06
Start Month of Year: 1
Number of Rounds: 1
Simulation Period: 1901-01-01 - 2006-12-01
Input Sources
NDVI file: ndvi.csv



SCE Parameters

Name	Value	Range
max_exec	10000	1-10000
init_complexes	2	>1
points_per_complex	11	
points_per_subcomplex	6	
min_complexes	11	
shuffling_loops	2	>1
of_percentage	0.1	(0-1.0]

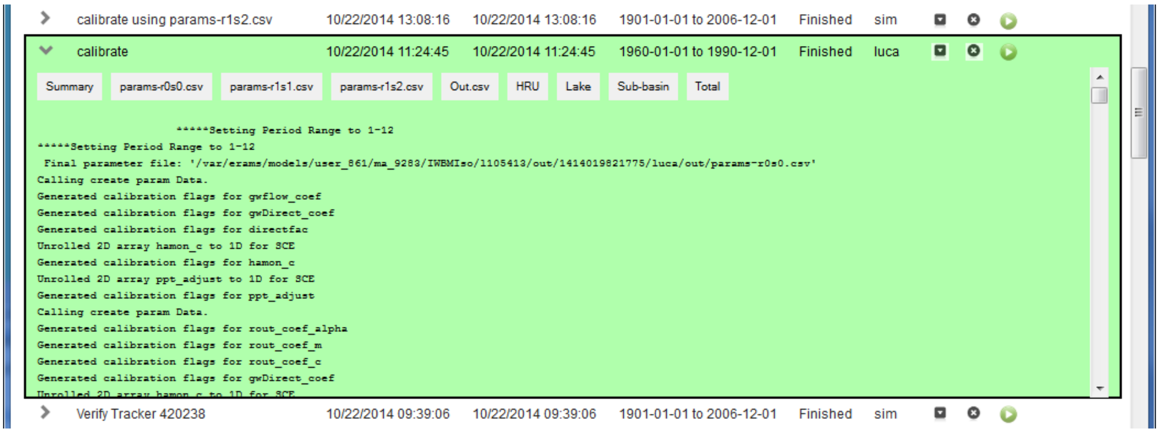
The application has defaulted values into the **SCE Parameters** table, but as needed edit by clicking the cell in the **Value** column and changing the value. Press enter when finished.

Click the **View** button to display the calibration file that will be generated. Click **Save as default** to start with these selections for the next new calibration run.

When calibration setup is finished, click **Run** to execute the model in calibration mode. The application alerts when the calibration is finished and displays buttons for viewing the output, including the parameter files generated after each round and step.

All model runs and calibrations are displayed in a list. Click the table header columns to sort. Click on a row to show options for displaying the plot and table outputs. Download output files by clicking on the down pointer button. Click the X button to delete the run. Click the right pointer button to rerun the simulation with new parameters and settings.

Console output from the model is displayed in the bottom portion of the green panel. Calibration runs generate a large amount of output, therefore only the head and tail of the output is shown.



calibrate using params-r1s2.csv	10/22/2014 13:08:16	10/22/2014 13:08:16	1901-01-01 to 2006-12-01	Finished	sim
calibrate	10/22/2014 11:24:45	10/22/2014 11:24:45	1960-01-01 to 1990-12-01	Finished	luca
Verify Tracker 420238	10/22/2014 09:39:06	10/22/2014 09:39:06	1901-01-01 to 2006-12-01	Finished	sim

```
                    *****Setting Period Range to 1-12
*****Setting Period Range to 1-12
 Final parameter file: '/var/erams/models/user_861/ma_9283/IWBMIso/1105413/out/1414019821775/luca/out/params-r0s0.csv'
Calling create param Data.
Generated calibration flags for gwflow_coef
Generated calibration flags for gwDirect_coef
Generated calibration flags for directfac
Unrolled 2D array hamon_c to 1D for SCE
Generated calibration flags for hamon_c
Unrolled 2D array ppt_adjust to 1D for SCE
Generated calibration flags for ppt_adjust
Calling create param Data.
Generated calibration flags for rout_coef_alpha
Generated calibration flags for rout_coef_m
Generated calibration flags for rout_coef_c
Generated calibration flags for gwDirect_coef
Unrolled 2D array hamon_c to 1D for SCE
```

14. Run Model With Calibrated Parameters

Calibrations generate a params-xxx.csv file after each round and step. Click the button of any of these files to open the **Run Model** dialog (described in Section 13) and run the model with calibrated parameters. In future versions, users will be able to go directly to the **Run Model** dialog and select the parameter file you want to use, including choosing calibrated parameter file.

Appendix 1 – IWBMIso Model and Model Component Parameters, Input Variables, and Output Variables – Definitions, Units and Dimensions

IWBMIso

Model and Model Component Parameters, Input Variables, and Output Variables

Definitions, Units, and Dimensions

Chapter 1. Model Component

1. Component 'WbmLakeIsoSum'

Name `wbm.WbmLakeIsoSum`

1.1. Parameter

aet_method - `int` Method for estimation of actual ET (aet).

covden_summer - `double[]` Summer Vegetation cover density of an HRU.

covden_winter - `double[]` Winter Vegetation cover density of an HRU.

dh2_gw_extern - `double[]` dh2 of External groundwater contribution.

dh2_lake_init - `double[]` Initial dh2 of the soil zone.

directfac - `double[]` Coefficient for computing direct surface runoff from precipitation.

do18_gw_extern - `double[]` do18 of External groundwater contribution.

do18_lake_init - `double[]` Initial do18 of the soil zone.

endTime - `Calendar` Ending date of the simulation.

gw_extern - `double[]` External groundwater contribution.

gwDirect_casc - `int[]` Index of upstrean groundwater reservoir.

gwDirect_coef - `double[]` Coefficient for computing direct groundwater flow as a percentage of groundwater storage.

gwDirect_pct - `double[]` Percentage of gwDirect_casc contribution to HRU.

gwflow_coef - `double[]` Coefficient for computing groundwater flow as a percentage of groundwater storage.

gwres_stor_init - `double[]` Initial storage in the groundwater reservoir.

hamon_c - `double[][]` Monthly Hamon PET coefficients for each subbasin.

hru_area - `double[]` Area of each HRU in hecatres.

hru_isoStation - `int[]` Id of Isotope station assigned to an HRU.

hru_lat - `double[]` HRU latitude

hru_lon - `double[]` HRU longitude

hru_pstation - `int[]` Id of Precipitation station used for HRU with precipitation adjustment.

hru_rech_prefer - `double[]` Percentage of preferential recharge (0-1).

hru_rhnStation - `int[]` Id of rletaive humidity station assigned to an HRU.

hru_subbasin - `int[]` Index of subbasin assigned to each HRU

hru_tstation - `int[]` Id of Temperature station used for HRU with temp. adjustment.

hru_type - `int[]`	.
inputFileDir - `String`	Path for input files.
intercept_pct - `double[]`	maximum storage capacity per HRU [percent]
interceptionOn - `int`	interception flag: ON or OFF
isoH2FileName - `String`	Precipitation Delta H-2 file name.
isoO18FileName - `String`	Precipitation Delta O-18 file name.
lake_area_coef_unit - `int[]`	Area unit of measure used to compute lake elevation-area coefficients - 0 = km; 1 = hectare
lake_dynamic_depth - `double[]`	Lake dynamic depth at which lake-groundwater interaction is expected.
lake_elev_init - `double[]`	initial lake surface elevation
lake_elev_outflow - `double[]`	elevation of the lake outflow
lake_L_A_c0 - `double[]`	lake bathym elev-area coefficient, c0
lake_L_A_c1 - `double[]`	lake bathym elev-area coefficient, c1
lake_L_A_c2 - `double[]`	lake bathym elev-area coefficient, c2
lake_L_A_c3 - `double[]`	lake bathym elev-area coefficient, c3
lake_L_V_c0 - `double[]`	lake bathym elev-volume coefficient, c0
lake_L_V_c1 - `double[]`	lake bathym elev-volume coefficient, c1
lake_L_V_c2 - `double[]`	lake bathym elev-volume coefficient, c2
lake_L_V_c3 - `double[]`	lake bathym elev-volume coefficient, c3
lake_max_depth - `double[]`	Lake maximum known depth.
lake_mix_depth - `double[]`	Lake depth over which isotope exchange occurs during evaporation.
lake_sub - `int[]`	Index of subbasin assigned to each lake
lakeObsId - `int[]`	ID lake level observed data if, ratingCurveType = outflow data provided
lakeType - `int[]`	Type of Lake: 1:well_mixed, 2:Monomictic, 3:Dimictic, 4:Meromictic, 9: gradually desiccating isolated pond/wetland
mixingMonth - `boolean[]`	Lake complete mixing/turnover months (1 if true).
ndviFileName - `String`	NDVI file name.
nhru - `int`	Number of HRUs.
niso - `int`	Number of Isotope measurement stations.
nlake - `int`	Number of lakes.
nmonths - `int`	Number of months.
nrain - `int`	Number of rain gauges.
nsub - `int`	Number of subbasins.

ntemp - `int`

Number of temperature measurement stations.

obsFileName - `String`

Measured/Observed river discharge and lake level file name.

outlet_id - `int`

Runoff station id to be used as measured basin outflow.

pet_method - `int`

method for estimating pet.

petFileName - `String`

PET file name.

ppt_adjust - `double[][]`

Monthly precipitation adjustment factor for each HRU.

pptFileName - `String`

precipitation file name.

print_freq - `int`

Frequency for output data file (0=none; 1=run totals; 2=yearly; 4=monthly; or additive combinations) For combinations, add index numbers, e.g., monthly plus yearly = 6; yearly plus total = 3

print_type - `int`

Type of output data file (0=measured and simulated flow only; 1=water balance table; 2=detailed output)

ratingCurveType - `int[]`

Type of lake discharge-lake level rating curve, 1: a*(h-b)^m + c, 2:outflow data provided

rhnFileName - `String`

Relative humidity file name.

rout_coef_alpha - `double[]`

Linear routing coefficient to compute lake outflow as function of storage above outflow elevation.

rout_coef_b - `double[]`

coefficient to compute lake outflow as function of storage above outflow elevation.

rout_coef_c - `double[]`

coefficient to compute lake outflow as function of storage above outflow elevation.

rout_coef_m - `double[]`

Linear routing coefficient to compute lake outflow as function of storage above outflow elevation.

soil2gw_max - `double[]`

Maximum flux from soil zone to groundwater per month.

ssflow_coef - `double[]`

Coefficient for computing subsurface flow as a percentage of subsurface reservoir storage.

startTime - `Calendar`

Start of simulation

sub_type - `int[]`

Sub basin type (1=Land, 2=water).

subbasin_down - `int[]`

Downstream subbasin for each subbasin

sumFileHru - `File`

File name for user selected Monthly Mean summary output file for Hrus.

sumFileLake - `File`

File name for user selected summary output file for lakes.

sumFileSub - `File`

File name for user selected summary output file for subbasins.

sumFileTot - `File`

File name for user selected summary output file for the total basin.

summerMonth - `int[]`

Summer months.

temp_adjust - `double[][]`

Monthly temperature adjustment factor for each HRU.

theta - `double[]`

fractionation parameter: theta small water bodies =1, great lakes = 0.88; mediterranean sea =0.5

tmaxFileName - `String` Maximum temperature file name.

tminFileName - `String` Minimum temperature file name.

whc - `double[]` Soil water holding capacity.

xmeltcoeff - `double[]` Snowmelt rate coefficient.

Chapter 2. Sub Component

1. 'wbm'

1.1. Component 'Climate'

Name wbm.Climate

1.1.1. Parameter

hru_isoStation - int[] Id of Isotope station assigned to an HRU.

hru_pstation - int[] Id of Precipitation station used for HRU with precipitation adjustment.

hru_rhnStation - int[] Id of rletaive humidity station assigned to an HRU.

hru_subbasin [none] - int[] Index of subbasin assigned to each HRU

hru_tstation - int[] Id of Temperature station used for HRU with temp. adjustment.

nhru - int Number of hydrologic response units (HRUs).

niso - int Number of Isotope measurement stations.

nrain - int Number of rain gauges.

ntemp - int Number of temperature measurement stations.

ppt_adjust [none] - double[][] Monthly precipitation adjustment factor for each HRU.

temp_adjust [none] - double[][] Monthly temperature adjustment factor for each HRU.

1.1.2. Variables (In)

date [yyyy mm] - Calendar Date of the current time step

dh2_ppt [dh2] - double[] Measured monthly average precipitation dh2 at selected stations

do18_ppt [do18] - double[] Measured monthly average precipitation do18 at selected stations

ndvi [none] - double[] Long term Satellite derived NDVI per HRU.

precip [mm] - double[] Measured monthly precipitation.

rhn [none] - double[] Measured monthly average normalized relative humidity.

tmax [deg C] - double[] Measured monthly maximum temperature.

tmin [deg C] - double[] Measured monthly minimum temperature.

1.1.3. Variables (Out)

hru_H2 [delta h2] - double[] Delta h2 of the hru precipitation.

hru_O18 [delta o18] - double[] Delta o18 of the hru precipitation.

hru_ppt [mm] - double[] Monthly HRU precipitation.

hru_rhn [mm] - double[] Measured monthly average normalized relative humidity

hru_temp `[deg C] - double[]` Monthly HRU temperature.

1.2. Component 'StationUpdater'

Name `wbm.StationUpdater`

1.2.1. Parameter

endTime `- Calendar` End of simulation

inputFileDir `[none] - String` Path for input files.

isoH2FileName `[none] - String` Precipitation Delta H-2 file name.

isoO18FileName `[none] - String` Precipitation Delta O-18 file name.

ndviFileName `[none] - String` NDVI file name.

obsFileName `[none] - String` Measured/Observed river discharge and lake level file name.

pet_method `[none] - int` method for estimating pet.

petFileName `[none] - String` PET file name.

pptFileName `[none] - String` precipitation file name.

rhnFileName `[none] - String` Relative humidity file name.

startTime `- Calendar` Start of simulation

tmaxFileName `[none] - String` Maximum temperature file name.

tminFileName `[none] - String` Minimum temperature file name.

1.2.2. Variables (Out)

date `- Calendar`

dh2_ppt `[dh2] - double[]` Measured monthly average precipitation dh2 at selected stations

do18_ppt `[do18] - double[]` Measured monthly average precipitation do18 at selected stations

hru_ndvi `[none] - double[]` Long term Satellite derived NDVI per HRU.

moreData `- boolean`

obs `[none] - double[]` Monthly observation data.

pet `[inches] - double[]` Mean monthly PET for each HRU

precip `[inches] - double[]` Measured monthly total precipitation at each rain gage

rhn `[none] - double[]` Measured monthly average normalized relative humidity.

tmax `[degrees] - double[]` Measured monthly mean maximum temperature at each measurement station

tmin `[degrees] - double[]` Measured monthly mean minimum temperature at each measurement station

1.3. Component 'WbHrusIso'

Isotope-enabled Monthly water balance model.

Name `wbm.WbHrusIso`

Author George H. Leavesley, Dagnachew L. Belachew - ghleavesley@colostate.edu, d.belachew@iaea.org

Keyword Water Balance, Stable Isotope, Fractionation

Version $Id: Main.java 861 2010-01-21 01:54:38Z ghleavesley $

Source $URL: http://svn.javaforge.com/svn/oms/branches/oms3.prj.wbm/src/wbm/WbHruIso.java $

License http://www.gnu.org/licenses/gpl-2.0.html

1.3.1. Parameter

aet_method `[none] - int` Method for estimation of actual ET (aet).

covden_summer `[decimal percent] - double[]` Summer Vegetation cover density of an HRU.

covden_winter `[decimal percent] - double[]` Winter Vegetation cover density of an HRU.

dh2_gw_extern `[dh2] - double[]` dh2 of External groundwater contribution.

directfac `[decimal fraction] - double[]` Coefficient for computing direct surface runoff from precipitation.

do18_gw_extern `[do18] - double[]` do18 of External groundwater contribution.

gw_extern `[mm] - double[]` External groundwater contribution.

gwDirect_casc `[none] - int[]` Index of upstrean groundwater reservoir.

gwDirect_coef `[percentage] - double[]` Coefficient for computing direct groundwater flow as a percentage of groundwater storage.

gwDirect_pct `[none] - double[]` Percentage of gwDirect_casc contribution to HRU.

gwflow_coef `[percentage] - double[]` Coefficient for computing groundwater flow as a percentage of groundwater storage.

gwres_stor_init `[mm] - double[]` Initial storage in the groundwater reservoir.

hamon_c `[none] - double[][]` Monthly Hamon PET coefficients for each subbasin.

hru_area `[hectares] - double[]` Area of each HRUin hectares.

hru_lat `[dec deg] - double[]` HRU latitude

hru_lon `[dec deg] - double[]` HRU longitude

hru_rech_prefer `[percent] - double[]` Percentage of preferential recharge (0-1).

hru_subbasin `[none] - int[]` Index of subbasin assigned to each HRU

hru_type `[none] - int[]` HRU type (1 = land, 2 = lake).

intercept_pct `[none] - double[]` maximum storage capacity per HRU [percent]

interceptionOn `[none] - int` interception flag: ON or OFF

lake_sub `[none] - int[]` Index of subbasin assigned to each lake

nhru - int Number of HRUs.

nlake - int Number of lakes.

nmonths - int Number of months.

nsub - int Number of Subbasins.

pet_method [none] - int method for estimating pet.

soil2gw_max [mm/month] - double[] Maximum flux from soil zone to groundwater per month.

ssflow_coef [decimal fraction] - double[] Coefficient for computing subsurface flow as a percentage of subsurface reservoir storage.

subbasin_down [none] - int[] Downstream subbasin for each subbasin

summerMonth [none] - int[] Summer months.

whc [mm] - double[] Soil water holding capacity.

xmeltcoeff [decimal fraction] - double[] Snowmelt rate coefficient.

1.3.2. Variables (In)

date [yyyy MM] - Calendar Current date.

dh2_sub_lake [do18] - double[] Delta h2 of the subbasin lake surface.

do18_sub_lake [do18] - double[] Delta o18 of the subbasin lake surface.

hru_H2 [delta h2] - double[] Delta h2 of the hru precipitation.

hru_ndvi [none] - double[] NDVI for each HRU.

hru_O18 [delta o18] - double[] Delta o18 of the hru precipitation.

hru_ppt [mm] - double[] Monthly HRU precipitation.

hru_rhn [mm] - double[] Measured monthly average normalized relative humidity

hru_temp [deg C] - double[] Monthly HRU mean monthly temperature.

lake_area [hectares] - double[] Lake area

lakeDynamic_stor [none] - double[] Lake Groundwater reservoir storage.

pet [mm] - double[] Satellite estimated potential ET.

1.3.3. Variables (Out)

ae [mm] - double[] Actual evapotranspiration.

ae_soil [mm] - double[] Actual evapotranspiration from VegET.

deficit [mm] - double[] The difference between actual and potential evapotranspiration when PET is greater than AET.

dh2_gwflow [delta h2] - double[] Delta h2 of the groundwater flow.

dh2_gwflow_up [none] - double[] dH2 of groundwater flow from upstream reservoir.

dh2_gwres [delta h2] - double[] Delta h2 of the groundwater reservoir.

dh2_intc [delta o18] - double[] Delta h2 of the interception reservoir.

dh2_ro [delta h2] - double[] Delta h2 of the total hru runoff.

dh2_rodirect [delta h2] - double[] Delta h2 of the direct runoff.

dh2_sm [delta h2] - double[] Delta h2 of the soil reservoir.

dh2_sm2gw [delta h2] - double[] Delta h2 of the soil water excess moving to the groundwater reservoir.

dh2_sm2ssr [delta h2] - double[] Delta h2 of the soil water excess moving to the subsurface reservoir..

dh2_ssflow [delta h2] - double[] Delta h2 of the subsurface flow.

dh2_ssr2gw [delta h2] - double[] Delta h2 of the soil water excess moving to the groundwater reservoir.

dh2_ssres [delta h2] - double[]Delta h2 of the subsurface reservior.

dh2_ssrflow [delta h2] - double[] Delta h2 of the subsurface flow.

do18_gwflow [delta o18] - double[] Delta o18 of the groundwater flow.

do18_gwflow_up [none] - double[] dO18 of groundwater flow from upstream reservoir.

do18_gwres [delta o18] - double[] Delta o18 of the groundwater reservoir.

do18_intc [delta o18] - double[] Delta o18 of the interception reservoir.

do18_ro [delta o18] - double[] Delta o18 of the total hru runoff.

do18_rodirect [delta o18] - double[] Delta o18 of the direct runoff.

do18_sm [delta o18] - double[] Delta o18 of the soil reservoir.

do18_sm2gw [delta o18] - double[] Delta o18 of the soil water excess moving to the groundwater reservoir.

do18_sm2ssr [delta o18] - double[] Delta o18 of the soil water excess moving to the subsurface reservoir..

do18_ssflow [delta o18] - double[] Delta o18 of the subsurface flow.

do18_ssr2gw [delta o18] - double[] Delta o18 of the soil water excess moving to the groundwater reservoir.

do18_ssres [delta o18] - double[] Delta o18 of the subsurface reservior.

do18_ssrflow [delta o18] - double[] Delta o18 of the subsurface flow.

gw_recharge [none] - double[] groundwater recharge.

gwflow [mm] - double[] Groundwater flow.

gwflow_direct [mm] - double[] Direct Groundwater flow.

gwflow_loss [m3] - double[] groundwater loss from HRU to outside of the basin.

gwflow_up [mm] - double[] Routed groundwater flow from upstream reservoir.

gwres_stor [mm] - double[] Groundwater reservoir storage.

interception - `double[]` state variable interception

intercStorage - `double[]` state variable interception storage

pe `[mm] - double[]` Potential evapotranspiration.

prain `[mm] - double[]` Precipitation on an hru in the form of rain, minus direct runoff.

psnow `[mm] - double[]` Precipitation in the form of snow.

ro `[mm] - double[]` Total HRU runoff (surface + subsurface + gw).

rodirect `[mm] - double[]` Direct surface runoff.

sm_stor `[mm] - double[]` Soil moisture storage.

smeltfrac `[decimal fraction] - double[]` The fraction of snstor that melts in a month.

snowpratio `[ratio] - double[]` The ratio of psnow to total hru preciptation (hru_ppt).

snroff `[mm] - double[]` Snowmelt.

snstor `[mm] - double[]` Snowpack storage in water equivalent.

ssflow `[mm] - double[]` Subsurface flow.

ssres_stor `[mm] - double[]` Subsurface reservoir storage.

sub_lkgwout `[m3] - double[]` Gw outflow from lake to hrus in m3.

sub_lkgwout_m `[m] - double[]` Gw outflow from lake to hrus in m.

throughfall - `double[]` state variable throughfall

wetinput `[mm] - double[]` The sum of prain + snroff.

1.3.4. Bibliography

- Leavesley, G. H., Lichty, R. W., Troutman, B. M., and Saindon, L. G., 1983, Precipitation-runoff modeling system--user's manual: U. S. Geological Survey Water Resources Investigations report 83-4238, 207 p.

1.4. Component 'WbSubbasinIso'

Monthly water balance model subbasin component.

Name `wbm.WbSubbasinIso`

Author George H. Leavesley, Dagnachew L. Belachew - ghleavesley@colostate.edu, d.belachew@iaea.org

Keyword Water Balance

Version $Id: Main.java 861 2010-01-21 01:54:38Z ghleavesley $

Source $URL: http://svn.javaforge.com/svn/oms/branches/oms3.prj.wbm/src/wbm/WbSubbasinIso.java $

License http://www.gnu.org/licenses/gpl-2.0.html

1.4.1. Parameter

dh2_lake_init `[dh2] - double[]` Initial dh2 of the soil zone.

do18_lake_init [do18] - double[] Initial do18 of the soil zone.

gw_extern [mm] - double[] External groundwater contribution.

gwres_stor_init [mm] - double[] Initial storage in the groundwater reservoir.

hru_area [hectares] - double[] Area of each HRU in hecatres.

hru_subbasin [none] - int[] Index of subbasin assigned to each HRU

hru_type [none] - int[] .

lake_area_coef_unit [none] - int[] Area unit of measure used to compute lake elevation-area coefficients - 0 = km; 1 = hectare

lake_dynamic_depth [m] - double[] Lake dynamic depth at which lake-groundwater interaction is expected.

lake_elev_init [m] - double[] initial lake surface elevation

lake_elev_outflow [m] - double[] elevation of the lake outflow

lake_L_A_c0 [none] - double[] lake bathym elev-area coefficient, c0

lake_L_A_c1 [none] - double[] lake bathym elev-area coefficient, c1

lake_L_A_c2 [none] - double[] lake bathym elev-area coefficient, c2

lake_L_A_c3 [none] - double[] lake bathym elev-area coefficient, c3

lake_L_V_c0 [none] - double[] lake bathym elev-volume coefficient, c0

lake_L_V_c1 [none] - double[] lake bathym elev-volume coefficient, c1

lake_L_V_c2 [none] - double[] lake bathym elev-volume coefficient, c2

lake_L_V_c3 [none] - double[] lake bathym elev-volume coefficient, c3

lake_max_depth [m] - double[] Lake maximum known depth.

lake_mix_depth [m] - double[] Lake depth over which isotope exchange occurs during evaporation.

lake_sub [none] - int[] Index of subbasin assigned to each lake

lakeObsId [none] - int[] ID lake level observed data if, ratingCurveType = outflow data provided

lakeType [none] - int[] Type of Lake: 1:well_mixed, 2:Monomictic, 3:Dimictic, 4:Meromictic, 9: gradually desiccating isolated pond/wetland

mixingMonth [none] - boolean[] Lake complete mixing/turnover months (1 if true).

nhru - int Number of HRUs.

nlake - int Number of lakes.

nsub - int Number of subbasins.

outlet_id [none] - int Runoff station id to be used as measured basin outflow.

ratingCurveType [none] - int[] Type of lake discharge-lake level rating curve, 1: a*(h-b)^m + c, 2:outflow data provided

rout_coef_alpha [none] - double[] Linear routing coefficient to compute lake outflow as function of storage above outflow elevation.

rout_coef_b [none] - double[] coefficient to compute lake outflow as function of storage above outflow elevation.

rout_coef_c [none] - double[] coefficient to compute lake outflow as function of storage above outflow elevation.

rout_coef_m [none] - double[] Linear routing coefficient to compute lake outflow as function of storage above outflow elevation.

sub_type [none] - int[] Sub basin type (1=Land, 2=water).

subbasin_down [none] - int[] Downstream subbasin for each subbasin

theta [none] - double[] fractionation parameter: theta small water bodies =1, great lakes = 0.88; mediterranean sea =0.5

whc [mm] - double[] Soil water holding capacity.

1.4.2. Variables (In)

ae [mm] - double[] Actual evapotranspiration.

date - Calendar Current date.

deficit [mm] - double[] The difference between actual and potential evapotranspiration when PET is greater than AET.

dh2_gwflow [delta h2] - double[] Delta h2 of the groundwater flow.

dh2_gwflow_up [dh2] - double[] dh2 of groundwater flow from upstream reservoir.

dh2_ro [delta h2] - double[] Delta h2 of the total hru runoff.

dh2_rodirect [delta h2] - double[] Delta h2 of the sirect runoff.

dh2_sm [delta h2] - double[] Delta h2 of the soil reservoir.

dh2_sm2gwr [delta h2] - double[] Delta h2 of the soil water excess moving to the groundwater reservoir.

dh2_sm2ssr [delta h2] - double[] Delta h2 of the soil water excess moving to the subsurface reservoir..

dh2_ssflow [delta h2] - double[] Delta h2 of the subsurface flow.

dh2_ssres [delta h2] - double[]Delta h2 of the subsurface reservior.

dh2_ssrflow [delta h2] - double[] Delta h2 of the subsurface flow.

do18_gwflow [delta o18] - double[] Delta o18 of the groundwater flow.

do18_gwflow_up [do18] - double[] dO18 of groundwater flow from upstream reservoir.

do18_ro [delta o18] - double[] Delta o18 of the total hru runoff.

do18_rodirect [delta o18] - double[] Delta o18 of the direct runoff.

do18_sm [delta o18] - double[] Delta o18 of the soil reservoir.

do18_sm2gwr `[delta o18] - double[]` Delta o18 of the soil water excess moving to the groundwater reservoir.

do18_sm2ssr `[delta o18] - double[]` Delta o18 of the soil water excess moving to the subsurface reservoir..

do18_ssflow `[delta o18] - double[]` Delta o18 of the subsurface flow.

do18_ssres `[delta o18] - double[]` Delta o18 of the subsurface reservior.

do18_ssrflow `[delta o18] - double[]` Delta o18 of the subsurface flow.

gw_recharge `[none] - double[]` groundwater recharge.

gwflow `[mm] - double[]` Groundwater flow.

gwflow_direct `[mm] - double[]` Direct Groundwater flow.

gwflow_loss `[m3] - double[]` groundwater loss from HRU to outside of the basin.

gwflow_up `[none] - double[]` Routed groundwater flow from upstream reservoir.

gwres_stor `[mm] - double[]` Groundwater reservoir storage.

hru_H2 `[delta h2] - double[]` Delta h2 of the hru precipitation.

hru_O18 `[delta o18] - double[]` Delta o18 of the hru precipitation.

hru_ppt `[mm] - double[]` Monthly HRU precipitation.

hru_rhn `[mm] - double[]` Measured monthly average normalized relative humidity

hru_temp `[deg C] - double[]` Monthly HRU mean monthly temperature.

obs `[none] - double[]` Monthly observation data.

pe `[mm] - double[]` Potential evapotranspiration.

prain `[mm] - double[]` Precipitation on an hru in the form of rain, minus direct runoff.

psnow `[mm] - double[]` Precipitation in the form of snow.

ro `[mm] - double[]` Total HRU runoff (surface + subsurface + gw).

rodirect `[mm] - double[]` Direct surface runoff.

sm_stor `[mm] - double[]` Soil moisture storage.

smeltfrac `[decimal fraction] - double[]` The fraction of snstor that melts in a month.

snowpratio `[ratio] - double[]` The ratio of psnow to total hru preciptation (hru_ppt).

snroff `[mm] - double[]` Snowmelt.

snstor `[mm] - double[]` Snowpack storage in water equivalent.

ssflow `[mm] - double[]` Subsurface flow.

ssres_stor `[mm] - double[]` Subsurface reservoir storage.

sub_lkgwout `[m3] - double[]` Gw outflow from lake to hrus in m3.

sub_lkgwout_m `[m] - double[]` Gw outflow from lake to hrus in m.

wetinput [mm] - double[] The sum of prain + snroff.

1.4.3. Variables (Out)

basin_ae [m3] - double Area-weighted basin actual evapotranspiration.

basin_ae_mm [mm] - double Basin ae in mm.

basin_area [hectares] - double Basin area.

basin_delstor [m3] - double Change in total basin storage.

basin_delstor_mm [mm] - double Basin delstor in mm.

basin_gw_gain [m3] - double Basin groundwater inflow expressed as volume in m3.

basin_gw_loss [m3] - double Basin groundwater outflow expressed as volume in m3.

basin_gwflow [m3] - double Area-weighted basin groundwagter flow.

basin_gwflow_mm [mm] - double Basin gwflow in mm.

basin_gwres_stor [m3] - double Area-weighted basin groundwater reservoir storage.

basin_lake_stor [m3] - double Basin lake storage.

basin_outvol [m3] - double Basin outflow express as volume in m3.

basin_pe [m3] - double Area-weighted basin potential evapotranspiration.

basin_pe_mm [mm] - double Basin pe in mm.

basin_prain [m3] - double Area-weighted basin precipitation occurring as rain.

basin_precip [m3] - double Area-weighted basin precipitation.

basin_precip_mm [mm] - double Basin precipitation in mm.

basin_psnow [m3] - double Area-weighted basin precipitation occurring as snow.

basin_rodirect [m3] - double Area-weighted basin surface runoff.

basin_rodirect_mm [mm] - double Basin precipitation in mm.

basin_snow_stor [m3] - double Area-weighted basin snowpack storage.

basin_snow_stor_mm [m3] - double Area-weighted basin snowpack storage in mm.

basin_snroff [m3] - double Area-weighted basin snowmelt runoff.

basin_soil_stor [m3] - double Area-weighted basin soil moisture storage.

basin_ssflow [m3] - double Area-weighted basin subsurface flow.

basin_ssflow_mm [mm] - double Basin ssflow in mm.

basin_ssres_stor [m3] - double Area-weighted basin subsurface reservoir storage.

basin_stor_lastyr [m3] - double Basin storage at end of last calander year.

basin_storage [m3] - double Total basin storage.

basin_totflow_mm [mm] - double Basin total flow in mm.

dh2_sub_gwflow [dh2] - double[] Delta h2 of the subbasin groundwater reservoir outflow.

dh2_sub_lake [dh2] - double[] Delta h2 of the subbasin lake surface.

dh2_sub_lake_2 [dh2] - double[] Delta h2 of the subbasin lake hypolimnion surface.

dh2_sub_lkgwin [dh2] - double[] dh2 of groundwater flow from upstream reservoirs to lake.

dh2_sub_precip [dh2] - double[] Delta h2 of the subbasin precipitation.

dh2_sub_rodirect [dh2] - double[] Delta h2 of the subbasin direct surface runoff.

dh2_sub_ssflow [dh2] - double[]Delta h2 of the subbasin subsurface reservoir outflow.

dh2_vflow_down [dh2] - double[] Delta h2 of the subbasin outflow to the downstream subbasin.

dh2_vflow_sub [dh2] - double[] Delta h2 of the subbasin outflow.

dh2_vsub [dh2] - double[] Delta h2 of the subbasin total runoff volume.

do18_gwflow_MM [do18] - double[][] Long term Monthly mean dO18 of groundwater flow per HRU.

do18_ro_MM [do18] - double[][] Long term Monthly mean dO18 of runoff per HRU.

do18_sub_gwflow [do18] - double[] Delta o18 of the subbasin groundwater reservoir outflow.

do18_sub_lake [do18] - double[] Delta o18 of the subbasin lake surface.

do18_sub_lake_2 [do18] - double[] Delta o18 of the subbasin lake hypolimnion surface.

do18_sub_lkgwin [do18] - double[] dO18 of groundwater flow from upstream reservoirs to lake.

do18_sub_precip [do18] - double[] Delta o18 of the subbasin precipitation.

do18_sub_rodirect [do18] - double[] Delta o18 of the subbasin direct surface runoff.

do18_sub_ssflow [do18] - double[] Delta o18 of the subbasin subsurface reservoir outflow.

do18_vflow_down [do18] - double[] Delta o18 of the subbasin outflow to the downstream subbasin.

do18_vflow_sub [do18] - double[] Delta o18 of the subbasin outflow.

do18_vsub [do18] - double[] Delta o18 of the subbasin total runoff volume.

gwdirect_sub [m3] - double[] Subbasin direct groundwater outflow

gwflow_MM [mm] - double[][] Long term Monthly mean groundwater flow per HRU.

gwloss_sub [m3] - double[] Total groundwater loss from the basin.

head [m] - double[] Lake depth above the outflow elevation.

lake_ae_mm [mm] - double[] Area-weighted lake evaporation in mm.

lake_area [hectares] - double[]Lake area

lake_out_m [m] - double[] Lake outflow expressed as change in lake depth.

lake_precip_mm [mm] - double[]Area-weighted lake precipitation in mm.

lake_stor [m3] - double[] Lake storage.

lakeDynamic_stor `[none] - double[]` Lake Groundwater reservoir storage.

lakesurf_elev `[m] - double[]` lake surface elevation

land_area `[hectares] - double` Land area

ro_MM `[mm] - double[][]` Long term Monthly mean runoff per HRU.

runoff_mm `[mm] - double` Basin observed runoff in mm.

sub_ae `[m3] - double[]` Sum of hru ae on each subbasin from associated HRUs

sub_ae_mm `[mm] - double[]` Subbasin actual evapotranspiration in mm.

sub_area `[hectares] - double[]` Subbasin area

sub_cfs `[cfs] - double[]` Total outflow from each subbasin

sub_cms `[cms] - double[]` Total outflow from each subbasin

sub_gw_recharge `[mm] - double[]` Total groundwater recharge from each subbasin

sub_gw_recharge_mm `[mm] - double[]` Subbasin groundwater recharge in mm.

sub_gwdirect_mm `[mm] - double[]` Subbasin groundwater direct flow in mm.

sub_gwflow `[m3] - double[]` Subbasin groundwater flow

sub_gwflow_mm `[mm] - double[]` Subbasin groundwater flow in mm.

sub_lkae_mm `[mm] - double[]` Actual evapotranspiration from lake surface area in mm.

sub_lkdelstor `[m3] - double[]` Storage change in lake volume in m3.

sub_lkgwin `[m3] - double[]` Gw inflow to lake from hrus in m3.

sub_lkinflow `[m] - double[]` Lake inflow from river(s) in m.

sub_lkoutflow `[m3] - double[]` Lake outflow in m3.

sub_mcm `[mcm] - double[]` Total outflow from each subbasin

sub_mm `[mm] - double[]` Total outflow from each subbasin

sub_pe `[m3] - double[]` Sum of hru pe on each subbasin from associated HRUs

sub_pe_mm `[mm] - double[]` Subbasin potential evapotranspiration in mm.

sub_prain `[m3] - double[]` Sum of hru_ppt occurring as rain on each subbasin from associated HRUs

sub_precip `[m3] - double[]` Sum of hru precip on each subbasin from associated HRUs

sub_precip_mm `[mm] - double[]` Subbasin precipitation in mm.

sub_psnow `[m3] - double[]` Sum of hru_ppt occurring as snow on each subbasin from associated HRUs

sub_rodirect `[m3] - double[]` Subbasin surface flow

sub_rodirect_mm `[mm] - double[]` Subbasin direct runoff in mm.

sub_ssflow `[m3] - double[]` Subbasin subsurface flow

sub_ssflow_mm `[mm] - double[]` Subbasin subsurface flow in mm.

sub_totflow_mm `[mm] - double[]` Subbasin total flow in mm.

vflow_sub `[m3] - double[]` Subbasin outflow volume

vsub `[m3] - double[]` Subbasin outflow volume

vsub_gwdirect `[m3] - double[]` Subbasin direct groundwater outflow volume

watbal `[m3] - double` Water balance error for a given timestep

watbal_sum `[m3] - double` Accumulated water balance error for the entire model run.

1.4.4. Bibliography

- Leavesley, G. H., Lichty, R. W., Troutman, B. M., and Saindon, L. G., 1983, Precipitation-runoff modeling system--user's manual: U. S. Geological Survey Water Resources Investigations report 83-4238, 207 p./n

1.5. Component 'WbSummaryIsoSub'

Monthly water balance model subbasin component.

Name `wbm.WbSummaryIsoSub`

Author George H. Leavesley, Dagnachew L. Belachew - ghleavesley@colostate.edu, d.belachew@iaea.org

Keyword Water Balance

Version $Id: Main.java 861 2010-01-21 01:54:38Z ghleavesley $

Source $URL: http://svn.javaforge.com/svn/oms/branches/oms3.prj.wbm/src/wbm/WbSummaryIsoSub.java $

License http://www.gnu.org/licenses/gpl-2.0.html

1.5.1. Parameter

endTime `[yyyy-mm] - Calendar` Ending date of the simulation.

lake_sub `[none] - int[]` Index of subbasin assigned to each lake

nhru `- int` Number of HRUs.

nlake `- int` Number of lakes.

nsub `- int` Number of subbasins.

outlet_id `[none] - int` Runoff station id to be used as measured basin outflow.

print_freq `- int` Frequency for output data file (0=none; 1=run totals; 2=yearly; 4=monthly; or additive combinations) For combinations, add index numbers, e.g., monthly plus yearly = 6; yearly plus total = 3

print_type `- int` Type of output data file (0=measured and simulated flow only; 1=water balance table; 2=detailed output)

sub_type `[none] - int[]` Sub basin type (1=Land, 2=water).

sumFileHru `- File` File name for user selected Monthly Mean summary output file for Hrus.

sumFileLake `- File` File name for user selected summary output file for lakes.

sumFileSub - File File name for user selected summary output file for subbasins.

sumFileTot - File File name for user selected summary output file for the total basin.

1.5.2. Variables (In)

basin_ae [m3] - double Area-weighted basin actual evapotranspiration.

basin_ae_mm [mm] - double Basin ae in mm.

basin_area [hectares] - double Basin area.

basin_delstor [m3] - double Change in total basin storage.

basin_delstor_mm [mm] - double Basin delstor in mm.

basin_gwflow [m3] - double Area-weighted basin groundwagter flow.

basin_gwflow_mm [mm] - double Basin gwflow in mm.

basin_gwres_stor [m3] - double Area-weighted basin groundwater reservoir storage.

basin_lake_stor [m3] - double Basin lake storage.

basin_outvol [m3] - double Basin outflow express as volume in m3.

basin_pe [m3] - double Area-weighted basin potential evapotranspiration.

basin_pe_mm [mm] - double Basin pe in mm.

basin_prain [m3] - double Area-weighted basin precipitation occurring as rain.

basin_precip [m3] - double Area-weighted basin precipitation.

basin_precip_mm [mm] - double Basin precipitation in mm.

basin_psnow [m3] - double Area-weighted basin precipitation occurring as snow.

basin_rodirect [m3] - double Area-weighted basin surface runoff.

basin_rodirect_mm [mm] - double Basin precipitation in mm.

basin_snroff [m3] - double Area-weighted basin snowmelt runoff.

basin_soil_stor [m3] - double Area-weighted basin soil moisture storage.

basin_ssflow [m3] - double Area-weighted basin subsurface flow.

basin_ssflow_mm [mm] - double Basin ssflow in mm.

basin_ssres_stor [m3] - double Area-weighted basin subsurface reservoir storage.

basin_stor_lastyr [m3] - double Basin storage at the end of a calander year.

basin_storage [m3] - double Total basin storage.

basin_totflow_mm [mm] - double Basin total flow in mm.

date [yyyy-mm] - Calendar Current date.

dh2_sub_gwflow [dh2] - double[] Delta h2 of the subbasin groundwater reservoir outflow.

dh2_sub_lake [dh2] - double[] Delta h2 of the subbasin lake surface.

dh2_sub_precip [dh2] - double[] Delta h2 of the subbasin precipitation.

dh2_sub_rodirect [dh2] - double[] Delta h2 of the subbasin direct surface runoff.

dh2_sub_ssflow [dh2] - double[] Delta h2 of the subbasin subsurface reservoir outflow.

dh2_vflow_down [dh2] - double[] Delta h2 of the subbasin outflow to the downstream subbasin.

dh2_vflow_sub [dh2] - double[] Delta h2 of the subbasin outflow.

dh2_vsub [dh2] - double[] Delta h2 of the subbasin total runoff volume.

do18_gwflow_MM [do18] - double[][] Long term Monthly mean dO18 of groundwater flow per HRU.

do18_ro_MM [do18] - double[][] Long term Monthly mean dO18 of runoff per HRU.

do18_sub_gwflow [do18] - double[] Delta o18 of the subbasin groundwater reservoir outflow.

do18_sub_lake [do18] - double[] Delta o18 of the subbasin lake surface.

do18_sub_precip [do18] - double[] Delta o18 of the subbasin precipitation.

do18_sub_rodirect [do18] - double[] Delta o18 of the subbasin direct surface runoff.

do18_sub_ssflow [do18] - double[] Delta o18 of the subbasin subsurface reservoir outflow.

do18_vflow_down [do18] - double[] Delta o18 of the subbasin outflow to the downstream subbasin.

do18_vflow_sub [do18] - double[] Delta o18 of the subbasin outflow.

do18_vsub [do18] - double[] Delta o18 of the subbasin total runoff volume.

gwflow_MM [mm] - double[][] Long term Monthly mean groundwater flow per HRU.

head [m] - double[] Lake depth above the outflow elevation.

lake_ae_mm [mm] - double[] Area-weighted basin precipitation.in mm.

lake_area [hectares] - double[] Lake area

lake_bottom [m] - double[] Elevation of the lake bottom.

lake_out_m [m] - double[] Lake outflow expressed as change in lake depth.

lake_precip_mm [mm] - double[] Observed runoff in mm.

lake_stor [m3] - double[] Lake storage.

lakesurf_elev [m] - double[] lake surface elevation

land_area [hectares] - double Land area

obs [none] - double[] Monthly observation data.

ro_MM [mm] - double[][] Long term Monthly mean runoff per HRU.

runoff_mm [mm] - double Basin observed runoff in mm.

sub_ae [m3] - double[] Sum of hru ae on each subbasin from associated HRUs

sub_ae_mm [mm] - double[] Subbasin actual evapotranspiration in mm.

sub_area [hectares] - double[] Subbasin area

sub_cfs [cfs] - double[] Total outflow from each subbasin

sub_cms [cms] - double[] Total outflow from each subbasin

sub_gw_recharge [mm] - double[] Total groundwater recharge from each subbasin

sub_gw_recharge_mm [mm] - double[] Subbasin groundwater recharge in mm.

sub_gwdirect_mm [mm] - double[] Subbasin groundwater direct flow in mm.

sub_gwflow [m3] - double[] Subbasin groundwater flow

sub_gwflow_mm [mm] - double[] Subbasin groundwater flow in mm.

sub_lkdelstor [m3] - double[] Storage change in lake volume in m3.

sub_lkgwin [m3] - double[] Gw inflow to lake from hrus in m3.

sub_lkgwout [m3] - double[] Gw outflow from lake to hrus in m3.

sub_lkinflow [m3] - double[] Lake inflow from river(s) in m3.

sub_lkoutflow [m3] - double[] Lake outflow to river in m3.

sub_mcm [mcm] - double[] Total outflow from each subbasin

sub_mm [mm] - double[] Total outflow from each subbasin

sub_pe [m3] - double[] Sum of hru pe on each subbasin from associated HRUs

sub_pe_mm [mm] - double[] Subbasin potential evapotranspiration in mm.

sub_prain [m3] - double[] Sum of hru_ppt on each subbasin from associated HRUs

sub_precip [m3] - double[] Sum of hru precip on each subbasin from associated HRUs

sub_precip_mm [mm] - double[] Subbasin precipitation in mm.

sub_rodirect [m3] - double[] Subbasin surface flow

sub_rodirect_mm [mm] - double[] Subbasin direct runoff in mm.

sub_ssflow [m3] - double[] Subbasin subsurface flow

sub_ssflow_mm [mm] - double[] Subbasin subsurface flow in mm.

sub_totflow_mm [mm] - double[] Subbasin total flow in mm.

vflow_sub [m3] - double[] Subbasin outflow volume

vsub [m3] - double[] Subbasin outflow volume

watbal [m3] - double Water balance error for a given timestep

watbal_sum [m3] - double Accumulated water balance error for the entire model run.

1.5.3. Variables (Out)

sub_gw_recharge_yr [mm] - double[] Annual groundwater recharge for each subbasin.

sub_ppt_yr [mm] - double[] Annual precipitation by year for each subbasin