# Accounting

--------------------

## URGWOM OLAM limitations

The URGWOM OLAMs are not affected directly by unit schemes. That is to say that changing the unit scheme while making no other changes to the model should produce the same results (when converted back to the original unit scheme). There are calculations in the methods, however, that are hard-coded to be carried out in specific units (not standard RW units) and assume a timestep of one day. These calculations use parameter values from table slots that have a unit type of NONE. This means that the user must know what units are used in the calculation when they set the parameter values. Details are given below.

NOTE: This has not yet been tested by actually running URGWOM. This analysis is only based on looking at how the methods are coded and the units on the slots in the model.

### Total Vol Sed (Post 2000)

Abiquiu, Chociti and Jemez use this method. It is only called from within the Reservoir Account Gain Loss method (it is not called in sim). The method uses three table slots with columns that have unit type NONE, but the parameter values entered in these columns assume a specific unit type for inflow (cfs), sediment load (tons) and a specific timestep length (day). The units for the calculations are hard-coded (not RW standard units). The three table slots are:

 - Sed Data

 - Seasonal Inflow Coeffs

 - Bed Load Coeffs

 The calcs are:

 The first two calcs (bedload, totalload) would be unaffected by a change in units as long as the parameter values (in Res.Bed Load Coeffs and Res.Seasonal Inflow Coeffs) stay the same. Changing the unit scheme will not alter values in slots with a unit type of NONE; however, any new parameter values entered in the table slots must assume flow in cfs and a timestep length of one day. The totalload calc would be affected by a change in timestep length. The third calc (sedabvpermpool) should not be affected by a change in units because it is only calculating a ratio (done in standard units), the result of which is unit-less. The ratio is then multiplied by a unit-less value and raised to a unit-less exponent.

### Pan and Ice Evaporation (Sim and Acct, K Factor)

The evaporation calculation would not be affected by a change in the unit scheme, but the K Factor (LengthperTemperature\_F) slot used in evaporation calculation (if the Pan Ice Switch slot is set to 1) assumes a time step length of one day. The K Factor is the rate component of the calculation but does not contain a time element in the unit type.

Changing the unit scheme would not affect the calculation because Temperature(C) and TemperatureInFahrenheit are two separate unit types. Changing the timestep length would require a new K Factor because the time component is not explicit in the unit (if this method would ever be applied for an alternative timestep length).

### PreRes Irrigated Area Loss Rate periodic slot and PreRes Meadow Area Loss Rate periodic slot

These slots both have unit type Length (not Velocity), thus their values are inherently dependent on the timestep length. The slots are used in acctPreResLossCalc which is called within Abiquiu, Cochiti, Elephant Butte, El Vado, Jemez and Nambe Falls Loss Calculation methods. Again, the calculations using these slots would not be affected by a change in unit scheme but would be affected by a change in timestep length.

# Model Building

--------------------

No problems were encountered when building a small test model. The exercise was updated (projects\riverware\doc\Training\CVEN5393\2013\HW4\_SourceFiles\CVEN5393\_HW04\_REVunits2013.03.22.docx), primarily by removing instructions to set units in the slot configuration dialogs and adding the instructions to set the unit scheme. The model building instructions are in the following order:

 - Set run parameters (run period and timestep length)

 - Add objects to the workspace

 - Select user methods

 - Set unit scheme

 - Link slots

 - Enter data ...

It seemed to make sense to put the unit scheme step after selecting methods so that all of the units types that would eventually be in the model would show up in the Unit Scheme Manager. This made more sense than trying to specify units before adding objects by de-selecting "Show only types that are present on the workspace" in the Unit Scheme Manager and then having to go through the entire list of unit types to pick out those that might be applicable. This is a fundamental change from the previous approach using the riverwareDB file where units should be specified before adding objects to the model. I don't know how/if new documentation places the setting of unit schemes in the overall steps of building a model.

# Objects

--------------------

## Different Results with Different Unit Schemes

### URGWOM

Running URGWOM with a different unit scheme produces different results. For example, creating a new unit scheme with all default (SI) units (with the exception of unit type Time which must be in “day” for some slots), then running the model, then switching back to the original (automatically generated) unit scheme will show different results than running the model with the automatic, transitional unit scheme. In URGWOM, the specific cause is the evaporation method used on some reaches, Inflow Exponent Pan Evap. This method uses the following formula for calculating evaporation:

The method converts to user units before raising to the exponent which has units of NONE, so changing the user unit would, in theory, require a different value for .

In URGWOM, creating an exception for Reach Inflows of “cfs” while leaving all other units in SI, will produce the same results as a model run with the original units. That is to say, that the Reach Inflows appear to be the only place in URGWOM that user units matter.

In general, any methods that use an exponent are likely to be problematic when switching between different unit schemes. In the code, search on pow( to find these locations.

## Methods Where User Units Matter

For these methods, a change in the unit scheme can produce different results. This is due to calculations which use user-units in combination with a user-input exponent with units of NONE.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Object** | **Category** | **User Method** | **Method Name in Code** | **Slot(s)** |
| Groundwater Storage | Groundwater Outflow Calculation | Exponential Flow | ExponentialFlow | Storage |
| Groundwater Storage | Deep Percolation | Exponential Percolation | exponentialPercolation | Storage |
| Reach | depthtoFlowMethod | depthtoFlowPowerFunction\*\*? | depthtoFlowPowerFunction | area\_tmpflow\_tmp |
| Reach | Evaporation Calculation | Inflow Exponent Pan Evap | empiricalEvap | Inflow |
| Reach | Seepage Calc | Seepage and Riparian CU Loss | seepageRiparianCULoss | Inflow |
| Reservoir | Sediment Calculation | CRSS Sediment Calc\*? | CRSSSedimentCalc? |  |

The Reach Tim Lag Routing and Variable Time Lag Routing methods will only allow Time Lag in Months if the time step is Monthly and requires a unit of Month for a monthly time step. Any deviation from this will cause a run abort with an error message, so there is no problem here. The Reach Interpolated Flow GainLoss method requires units of Day in the Day of Year column of the Interpolated GainLoss Coeff Table. Again, a run abort with an error message will prevent any problems. This is also true for Reach/Drain Elev Interpolation.

## Methods with Hard-Coded Units for Calculations

These methods convert values to specific units for calculations. These calculations also use user-input parameter values with units of NONE. Changing the unit scheme should not change the results from these methods, but the user must know the hard-coded units in order to set the parameter values appropriately.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Object** | **Category** | **User Method** | **Method Name in Code** | **Hard-Coded Units** |
| Reach | routingMethodCategory | Storage Routing | storageRouting | Flow – cfsStorage Time – hr  |
| Reach | routingMethodCategory | Variable Storage Routing | varStorageRouting | Flow – cfsStorage Time – hr |
| Distribution Canal | Flow Routing | Storage Time | storTimeRouting | Flow – cfsStorage Time – hr |
| Distribution Canal | Flow Routing | Variable Storage Time | varStorTimeRouting | Flow – cfsStorage Time – hr |
| Reservoir | Sediment Transport Calculations | Perm Pool Sed (Pre2000) | acctPermPoolSed | Flow – cfsSediment Load – tonsTimestep – day |
| Reservoir | Sediment Transport Calculations | Total Vol Sed (Post2000) | acctTotalVolSed | Flow – cfsSediment Load – tonsTimestep – day |

## Transitional Unit Schemes – Automatically Generated

### URGWOM

URGWOM\_5.0\_WorkingDRAFT\_Feb2012AOP\_2-12-12\_resequencing\_tool\_all50yrs.mdl.gz

|  |  |  |
| --- | --- | --- |
| **Unit Type** | **Exceptions** | **“Precision Only” Exceptions** |
| NONE | 77 | 77 |
| Flow | 279 | 40 |
| Volume | 144 | 92 |
| Length | 9 | 0 |
| Area | 7 | 1 |
| Time | 36 | 10 |
| Energy | 0 | 0 |
| Power | 0 | 0 |
| Velocity | 40 | 4 |
| Fraction | 27 | 13 |
|  AreaPerTime | 20 | 19 |
| Concentration | 0 | 0 |
| LengthPerTemeprature\_F | 0 | 0 |
| Mass | 0 | 0 |
| PowerPerFlow | 4 | 1 |
| TemperatureInFahren | 0 | 0 |
| TimePerLength | 3 | 0 |

For unit type of NONE, many of the precision exceptions are appropriate – integer values or parameter values that need high precision.

For unit type of Flow, 179 exceptions are in cms, presumably slots that have never had a unit other than the default set. There are 58 “legitimate” exceptions, e.g. acre-ft/day instead of cfs.

The Reservoir.PreRes Irrigated Area Loss Rate and Reservoir.PreRes Meadow Area Loss Rate slots were periodic slots converted to series slots with periodic inputs in 6.3. The new slots are given an exception in default (SI) units instead of the original user units (ft). This was also observed for the Reservoir.K Factor slots (Reservoir Pan and Ice Evaporation method).

Most exceptions for unit type Velocity are for evapotranspiration rates with scale of 0.5 (in/day, correct based on the 6.1 model).

### CRSS

CRSS-R.Jan2012.v3.7b.mdl.gz

|  |  |  |
| --- | --- | --- |
| **Unit Type** | **Exceptions** | **“Precision Only” Exceptions** |
| NONE | 58 | 58 |
| Flow | ~750 |  |
| Volume | 167 | 33 |
| Length | 25 | 2 |
| Area | 4 | 4 |
| Time | 45 | 2 |
| Energy | 20 | 1 |
| Power | 3 | 1 |
| Velocity | 13 | 4 |
| Fraction | 20 | 1 |
| Concentration | ~350 |  |
| FlowPerTime | 0 | 0 |
| Mass | ~350 | 0 |
| PowerPerFlow | 4 | 0 |
| TemperatureInFahren | 0 | 0 |
| VelocityPerTemperature\_F | 0 | 0 |
| noDimension | 0 | 0 |

Exceptions were created for <Any Object>.Diversion, Diversion Capacity, Diversion Shortage, Depletion, and Incoming Available Water with units of “cms.” Then there were a lot of slot-specific exceptions that set the units for these slots back to the rule for Flow of 21.12 acre-ft/month.

### TVA

TVA\_6hr\_21400 R1 BFN WEH 03TM515.mdl.gz

|  |  |  |
| --- | --- | --- |
| **Unit Type** | **Exceptions** | **“Precision Only” Exceptions** |
| NONE | 22 | 22 |
| Flow | 97 | 29 |
| Volume | 17 | 3 |
| Length | 114 | 0 |
| Area | 0 | 0 |
| Time | 0 | 0 |
| Energy | 58 | 19 |
| Power | 11 | 11 |
| Velocity | 0 | 0 |
| Fraction | 1 | 0 |
| Concentration | 0 | 0 |
| Mass | 0 | 0 |
| PowerCost | 0 | 0 |
| PowerPerFlow | 3 | 1 |
| Value | 2 | 2 |
| ValuePerVolume | 1 | 1 |

Most exceptions for Flow, Volume and Length that are not for precision are for SI units (slots that are not looked at and have not had user units set).

### Truckee

TruckeePlanningCADSWES.mdl.gz

|  |  |  |
| --- | --- | --- |
| **Unit Type** | **Exceptions** | **“Precision Only” Exceptions** |
| NONE | 30 | 30 |
| Flow | 152 | 29 |
| Volume | 49 | 3 |
| Length | 8 | 2 |
| Area | 3 | 1 |
| Time | 5 | 0 |
| Energy | 0 | 0 |
| Power | 3 | 0 |
| Velocity | 6 | 2 |
| Fraction | 3 | 2 |
| Concentration | 0 | 0 |
| Mass | 0 | 0 |
| PowerPerFlow | 0 | 0 |

Most exceptions for Flow that are not for precision are for SI units (slots that are not looked at and have not had user units set).

# Object User Interface

--------------------

## Export and Re-Import of Objects

The following behavior occurs if an object is exported, the scheme is changed, and then the object is re-imported:

If the OBJECT was originally created in 6.2 (whether or not it is exported from a model in 6.2 or 6.3), then:

* Scalar slots are left in the units in which they were exported. An exception is created in the unit scheme for each scalar slot that has a unit different from the new scheme unit for the given unit type. Values are correct. This also includes scalar slots with expression and includes both scalar slots with values and without values (showing NaN).
* All other slots on the re-imported object are set to the scheme units for the given unit type, whether or not they contain values.

If the object was created in 6.3, all slots, including scalar slots, on the re-imported object are in the units of the new scheme, and no exceptions are created when the object is re-imported. Units for slots on the imported object do adhere to existing exceptions in the new scheme of the form <Any Object>.Slot Name where they apply. Values are converted correctly.

## Table Slots on Imported Objects

If a data object containing one or more table slots is exported, when the object is re-imported, the last column in the last table slot dialog displays units of NONE, but in the slot configuration dialog it will show the units for the column that would be expected. (Filed as gnats 5296 because I was not sure if it was even related to unit schemes.)

# Optimization

--------------------

## Optimization Solution

When the solution has alternative optima, the objective function value (in diagnostics) is the same with two different unit schemes, but the individual variable values change. Similar behavior has been observed with other changes that would not seem to directly effect the solution (e.g. solving with 2 cores instead of 1 or using a different machine to solve the same model). This is not a problem. Other than the alternative optima the optimization solution appears to be unaffected by the unit schemes.

## Diagnostics and log files

Many values displayed in optimization diagnostics do not appear to adhere to the unit scheme. This may be due to where these values come from in the optimization problem and that they are not necessarily linked to slot values. It may not be possible to express many of the values with scheme units. Details are given below.

In diagnostics values with no units in a constraint expression are all displayed with 9 digits (precision depends on number of digits before the decimal).

Example:

320: OBJECT: LGS to LMN GOAL: None: ... Definition: ( 0.00 "kcfs" == ( ( ( ( LGS to LMN.Inflow [t + -2] + 0.00 "kcfs" ) \* 0.00000000 ) + ( ( LGS to LMN.Inflow [t + -1] + 0.00 "kcfs" ) \* 1.00000000 ) ) - LGS to LMN.Outflow [t] ) )

It is not clear if these should be subject to the unit scheme because they are not slot values.

In diagnostics, frozen variable expressions do not include units and always use a precision of 6.

Example:

30335: GOAL: (21) Spill Cap 12 Hour Average: Frozen variable: Grand Coulee.Turbine Release [23:00 April 17, 2006] = 7.928434 (reduced cost = 0.071609)

In this example the Turbine Release of 7.928434 is in "1000 cms". The user unit is "kcfs".

In diagnostics, objective satisfaction (green message) is always reported with precision of 6.

Example:

17374: GOAL: (3) Forebay Elevation Minimum and Maximum: Objective is 100.000000% satisfied at iteration 1.

It is not clear if these should be subject to the unit scheme because they are not slot values.

###

All constraint expressions use the unit scheme rule for the unit type not the exception for the slot that holds the constraint value (right-hand-side). Also, definition constraint expressions that use a slot config min/max value use the unit scheme rule for the unit type not the exception for the slot. This is not necessarily a problem, and in general it may not be possible to use the slot exceptions because not all right-hand-side values are taken directly from a single slot.

In diagnostics, some coefficients that are calculated automatically by RW, such as slopes from linearizations based on LP Param values, can have uncommon unit types such as AreaPerTime or LengthPerVolume. It could be likely that the units are not specified for the given unit type, so they appear in RW standard units. Changing the rule for the unit type in the unit scheme will change their units in diagnostics appropriately. There is not a problem here. The user would just need to understand that when an operation is performed on some user-input values in given units, the result value has a new unit type and thus the units on the resulting value will not be the combination of the user units on the input values.

Example:

Linearized constraint: ( ( ( 1.00000000 \* Lower Granite.Turbine Release [01:00 April 17, 2006] ) + ( 137.73 "m2/s" \* Lower Granite.Opt piece for Turbine Capacity.1 [01:00 April 17, 2006] ) ) <= 130.000 "kcfs" )

The Turbine Capacity LP Param slot has user units of "ft" and user units for flow are "kcfs". The resulting slope for the Turbine Capacity piece (flow/length) has units of "m2/s" in diagnostics.

###

The opt log does not use user units with the exception of statements about "False constraint match for hash value." In general, values in the opt log have no units and are typically the values are a scaled version of standard units (unchanged from 6.2). This should probably remain as is, but have Tim confirm. The same is true for the .lp files. Values have no units and should probably remain this way. It would not make sense to incorporate units into the constraint expressions in the .lp files.

# Units

--------------------

The automatically generated unit scheme from a model previously in 6.2.9 creates an exception for <Any Object>.Auto Best Turbine Q with unit of cms (the scheme Flow unit is kcfs). It then has an individual exception for every individual Object.Auto Best Turbine Q slot in the model that goes back to the same attributes as the scheme Flow rule. The same is true for the Operating Head (length) column of the slot and for the Auto First and Auto Max Turbine Q slots.

If the Unit Scheme Manager is open and displaying a scheme that is NOT active (we'll call it Scheme2), then if the user selects in the RW workspace Units -> Scheme2, Scheme2 will become active for all slots in the model, but the Unit Scheme Manager shows that it is Not Active, though the Activate button disappears. Switching to Scheme1 in the Unit Scheme Manager then back to Scheme2 will now show Scheme2 as active. It is just not updating the active unit scheme in the Unit Scheme Manager when the scheme is changed in the Units dropdown menu from the workspace.

The automatically generated unit scheme from a model previously in 6.2.9 creates an exception for series slots with periodic input (previously periodic slots). The exception uses default (SI) units instead of the unit that was on the original periodic slot (observed in URGWOM, PreRes Irrigated Area Loss Rate and PreRes Meadow Area Loss Rage slots and Reservoir.K Factor slots).

When a model is loaded in 6.3 for the first time, the automatically generated unit scheme shows exceptions for simulation slots only. If the model is saved (in 6.3), closed, then re-loaded in 6.3, the unit scheme now also displays numerous exceptions for supplies that were not shown before. These exceptions for supplies all appear to be in SI units. (Observed in URGWOM and Truckee)