Unit Schemes in RiverWare 6.1

Author: Phil Weinstein

This document describes a new architecture for control of the units and precision with which numeric slot values are displayed and interpreted in RiverWare.

0.1 Document Status

5-31-2011: Almost ready for review; Needs a “proof” pass.

0.2 Other Relevant Documents

- RiverWare Technical Documentation (“online help”) -- “Units” Section.
  The RiverWare 6.0 edition (11-22-2010) is available at this URL:
  http://cadswes.colorado.edu/PDF/RiverWare/documentation/Units.pdf

0.3 Contents

1.0 Overview

Currently, in RiverWare 6.0 and earlier versions of RiverWare, display units and "precision" (fractional decimal digits) with which numeric slot values are displayed are configured individually on each slot or slot column instance. Series slots for "flow" (volume per time) entities on accounts support distinct configurations for flow and volume units. There are tools within RiverWare for reassigning the display units and precision on individual slots, or on multiple slots in a single operation.

There is a fixed "standard unit" specified for each unit type (e.g. "m3" for volume). Generally, slots on newly created objects (including supplies and exchanges) are initialized with the standard unit for the slot's unit type -- or, if defined, to the unit specified for the slot's unit type or slot entity in the external "riverwareDB" file (see the referenced online help document). However, slots on newly created *accounts* are initialized with the unit settings specified in the Accounting System Configuration dialog (which persist as settings on the internal "prototype account" instances stored in the RiverWare model).

This document describes an alternative to this way of specifying slot display units and precision. Multiple named "Unit Schemes" will be stored within the RiverWare model. Initially, this will be used in conjunction with the current slot / slot column instance configurations (in one of several possible ways, explored in this document). Probably, when sufficiently comprehensive capabilities are implemented with the Unit Schemes architecture, the older instance configurations mechanism will be removed.

2.0 Fundamental capabilities of the existing slot / slot column instance configuration mechanism

This section discusses particular aspects of the current unit system which are relevant to adding the ability to define and apply multiple alternative "Unit Schemes" to numeric slot value display in RiverWare.

2.1 RiverWare Resource Database -- riverwareDB file

The initial configuration for slots and slot columns on new simulation objects (i.e. for only physical slots), including display units and precision is, defined in the optional external "riverwareDB" file. This effectively supports assigning a fixed set of properties (e.g. display scale and units -- see below) based on the following slot and slot column "entity" (name) matches:

- <object type> <slot name>
- <object type> <slot name> <column name>
- <slot name>
- <slot name> <column name>
- <unit type>

Technically, at a low level, there is support for slot "instance" matches (i.e. particular actual slots on particular actual objects). But the RiverWare Resource Database management system isn't structured in a way that that is actually usable for slots on the users' simulation objects -- and an "instance" capability isn't described in the RiverWare Technical Documentation.

> To be more specific, internally, there are, in fact, "slot instance based" queries to the internal (in-memory) representation of the resource database, but those are for only the hidden "prototype" simulation objects. After the internal resource database has been loaded from the riverwareDB file, all queries to the database (with one

minor exception to support a single DMI-related attribute) are done for the purpose of initializing the prototype simulation objects before a RiverWare model is loaded. (In the current RiverWare 6.1 development build, 305110 queries are made to the database using 11670 unique keys, most of which are references to physical slots and slot columns on the set of prototype simulation objects).

The way in which prototype simulation objects are used in this mechanism isn't really fundamental. Those prototype object slot and slot column references are equivalent to references made to those entities on object *types* (e.g. "Storage Reservoir"). A reimplementation of the internal resource database (in the course of the Unit Schemes architecture) should probably use only "entity based" keys, as enumerated above.

As described in the RiverWare Technical Documentation, the following slot and slot column attributes (properties) are supported. (Some of these attributes are relevant only for series slots).

- display units
- display scale (floating point value)
- display format (integer, float, or scientific notation)
- precision (integer: fractional decimal digit count)
- minimum value (floating point value)
- maximum value (floating point value)
- convergence type (non, absolute, percent, unit percent)
- convergence limit (double)
- max iterations

Other than through the external "riverwareDB" file, there is currently no way to change any of the properties of the prototype simulation objects -- i.e. for initializing properties of slots on subsequently created simulation objects. (This is different from the treatment of properties of the prototype *accounts*, described in the next section).

Note that the RiverWare Resource Database mechanism doesn't support initial properties for *accounting system* slots (i.e. slots on accounts, supplies or exchanges).

2.2 Prototype Account Slot Display Properties

When a new account is created, the account's slots' display properties are initialized from those set on one of the internal, hidden prototype accounts (which reside on the prototype simulation objects). The prototype account display properties (e.g. units, scale and precision) are not initialized through the RiverWare Resource Database (see prior section). Instead, they can be set by the user through the Accounting System Configuration GUI -- and such property edit operations can be applied to also existing user account instances (i.e. other than the internal prototype accounts).

2.3 Account Series Slot Display Properties: Primary and Alternate

Series slots on accounts support both a "primary" and "alternate" set of display properties (units, scale, precision, etc.) -- invariably for two different unit types related by a time factor (e.g. flow and volume, or power and energy). Internal accounting system computations are always done in the standard units of the "primary" unit type. For slots representing "flow" entities, generally the primary unit type is FLOW and the alternate unit type is VOLUME. But in cases where the natural way of computing the flow entity's values is a volume-based computation, the primary unit type is VOLUME and the alternate unit type is FLOW (for example, with the "Gain Loss" slot on storage accounts). However, where the GUI supports alternate units for slots, these two options are presented as "Flow" or

"Volume" (e.g. in the Edit Account dialog). Because of this, even though alternate units (on accounts) could support other *pairs* of units related by a time factor (e.g. power and energy), the GUI currently supports only alternate units involving flow and volume.

Naturally, not all account series slots support an "alternate" unit type. For example, entities that fundamentally represent a volume (e.g. the "Storage" series slot on a reservoir) cannot be displayed as a flow.

There is no fundamental reason why the ability to show flow entity values in series data as either "flow" or "volume" is limited to *accounting* slots. Extending that capability to physical slots is desirable, but is probably outside of the scope of the initial implementation of the Unit Scheme architecture. However, the architecture design will anticipate that implementation.

2.3.1 Implicit Flow / Volume Numeric Equivalence

Note that flow / volume conversions are implicitly automatic (even with physical slots) if the user chooses flow display units having a "time denominator" equal to the series timestep size, e.g. "acre-feet per day" in a daily timestep series (generally appropriate, in a daily timestep model). "Summing" several such flow values in multiple timesteps within the series does correctly represent the aggregate volume of water.

2.4 Ability to "correct" data entered or imported with the wrong units

The dialogs which allow the user to change the display units of slots and slot columns provide this checkbox:

[X] Convert slot values to new User Units and Scale

When applying a unit change with this checkbox *off*, the underlying internal "standard unit" values are rescaled such that the apparent numeric values in the new display units remain the same (e.g. changing 100 cms to 100 cfs -- i.e. "no conversion"). Internal standard unit values remain unchanged with the "convert" checkbox turned *on* (e.g. changing 100 cms to 3531.47 cfs). This feature is available in the following slot unit-related dialogs:

- Configure Slot dialog (accessible from Open Slot dialogs)
- Configure Existing Slots dialog (accessible from the Workspace's menubar: Workspace >> Slots >> Configure Slots; This dialog currently supports only physical slots).

Even though the information presented for making a correction to a unit-related "mistake" (e.g. having imported numeric data with the wrong units configured on a slot) is similar to that used for changing display units, it may be a liability to present the former capability in a new architecture supporting a more "fluid" (easily switchable) alternate display unit capability. The user should be able to switch between Unit Schemes without being concerned about unintentionally changing the underlying slot values. Furthermore, with the introduction of alternate display units not actually stored on slots and slot columns, the potential for confusion is greater. These two design scenarios need to be considered -- (described in more detail in subsequent sections):

- When "Unit Schemes" are used in conjunction with (sort of, as an "override" to) units configured on slots and slot columns (tentatively anticipated for the initial implementation of "Unit Schemes") the unit "correction" capability could simply remain with the current slot unit configuration mechanisms.
- When "Unit Schemes" replace units configured on slots and slot columns, a distinct "unit correction" utility should be developed apart from the display unit selection capability implemented with "Unit Schemes". A distinct "correction" utility might allow the user to view a subset of the slot data with a selectable ("current", but wrong) unit specification, and allow the user to specify a distinct ("correct") unit specification (which actually is correct for the specific displayed numeric values).

2.5 Other Considerations

2.5.1 Scale and Units are optionally indicated in the SCT

It is currently possible for the user to turn off the display of scale and units in row or column headers in the SCT's "Series Slot" tab. Not indicating scale and units within this numeric slot data display may be more of a liability with the introduction of a more flexible way switching between different unit schemes. The following possibilities can be considered:

- No longer providing the option of *hiding* scale and unit indications in row and column headers.
- Always showing the *name* of the currently active Unit Scheme in the SCT (at least on the "Series Slots" tab). This could take the form of a "combo box" (option menu) with which the user would switch between the various available Unit Schemes.

[This concern was raised by David].

3.0 Unit Schemes

3.1 The Structure of a Unit Scheme Instance

Generally speaking, a Unit Scheme is a mapping from slots, or -- in the case of multiple-column slots, slot columns -- to at least these two unit-related display properties:

1. Display Unit Scale (a floating point value)
2. Display Unit

For numeric values displayed with the "float" display format -- which is almost always used -- the choice of precision will often depend on the Display Unit Scale and Unit. For example, a precision of "2" (fractional decimal digits) may be desirable for only the first of these two approximately equivalent quantities:

- 12.34 cms
- 25930.89 acre-feet/month

So, it may be a good idea to include also at least the first of the following two properties in the Unit Scheme mapping from slot/slot columns. And since they are related, it might be a good idea to keep them together.

3. Display Precision (number of fractional decimal digits)
4. Display Format: *Integer, Float,* or *Scientific Notation.* (Note: We should consider removing the "Integer" display type. That is redundant with configuring a "Precision" of zero, and is a source of confusion).

If all four of the above properties are included, it may be more accurate to refer to "Unit Schemes" as "Numeric Display Schemes" or just "Display Schemes".

The mapping represented by a Display Scheme -- from Slot/Slot Column to the two or four display properties defined above -- could be defined as a prioritized list of individual matching rules. To provide at least the capabilities of the current RiverWare Resource Database, the following types of matching rules are needed -- this supports only physical slots (i.e. slots on simulation objects):

- <object type> .. <slot name>
- <object type> .. <slot name> .. <column name>

- <slot name>
- <slot name> .. <column name>
- <unit type>

To provide at least the capabilities supported for account slots on the prototype account slots, the additional matching rules would be needed. This includes the possibility of multiple column account slots, which may not actually be used yet, but the Unit Schemes mechanism design shouldn't assume that.

- <object type> .. <account type> .. <slot name> .. [unit type]
- <object type> .. <account type> .. <slot name> .. <column name> .. [unit type]

... the "unit type" value would be used only for slots supporting alternate units; and by convention, only two such rules would be defined for any account slot or account slot column entity. (That is, that assumption could be made in the implementation of any supporting internal or GUI module).

The only other type of accounting system object which would need display property support is "exchange"

- [exchange] .. <slot name> ... [unit type]

Note: See the "Other Alternatives to Support time-based Alternate Units" section, below. [TODO, explore this possibility].

3.1.1 Series Timestep Size-Dependent Matching Rules

Additionally, for the purpose of supporting implicit Implicit Flow / Volume Numeric Equivalence (see the section above), we may want to support the following advanced "unit type" matching rules for use with series slots and table series slot columns, typically for the "FLOW" (volume per time) and "POWER" (energy per time) unit types:

- <unit type in an hourly timestep series>
- <unit type in a 6-hour timestep series>
- <unit type in a 12-hour timestep series>
- <unit type in a weekly timestep series>
- <unit type in a daily timestep series>
- <unit type in a monthly timestep series>
- <unit type in an annual timestep series>

For example, it would be natural to show flows in a daily series as "acre-feet/day", but as "acre-feet/year" in an annual series (e.g. in an annual aggregation series). Note that the standard RiverWare "units" file does not support time denominators for all of these timestep sizes (e.g. m3/6-hours). Those strange units could be added by the user-- or the mapped display scale and unit scale can be set appropriately (e.g. in this case to 4.0 m3/day) -- possibly too confusing for actual use.

Alternatively, this sort of support could be presented as an option in the ordinary <unit type> matching rule. For example, if this matching rule as defined:

```
VOLUME  >> [1] m3
```

then all slots or slot columns having both Flow represented in their primary and alternate units, would automatically be displayed as "m3 / series-timestep-size". In fact, this ability would only require a slot or slot column to

have a flag indicating that the entity represents a “flow” (or having a per-time factor, e.g. “power”) even if its unit type is a “volume” (not having a per-time factor, e.g. “energy”). This would perhaps be a more elegant solution to supporting alternate units in physical slots, and could also be applied to account slots in place of the alternate unit configuration.

3.2 Other Alternatives for Support of Time-Based Alternate Units

The architecture described above makes use of multiple (two) “matching rules” to support alternate units (paired with primary units related by a time factor) -- currently supported only for account slots. Making use of *two* matching rules is the most similar to mechanisms in the current slot-based unit system (on accounts). But the following alternatives can be considered:

3.2.1 Additional level of mapping between related time-related unit types.

We could support an additional distinct user-specifiable mapping to define the equivalence between particular “flow” units and particular “volume” units (and similar support for other supported time-related pairs of unit types). This would change the architeture defined above in the following ways:

- only one “matching rule” would be possible for any slot-entity or slot-column-entity rule,
- the [unit type] fields in the matching rules specified above would not be used, and
- the series timestep size-dependent matching rules would be moved to this mapping.

The mapping from *volums to flows* could be dependent on the series timestep-size if the series slot. For example, m3 could be converted to m3/day in a daily series, but to m3/month in a monthly series. It would be reasonable to have the user specifiy this sort of conversion as “auto” (automatic), instead of requiring the user to pick the obvious flow units.

3.2.2 Allow either volume and flow units to be specified for either unit type

The current support for switching between “Flow” and “Volume” could itself be implemented as a switch between two Unit Schemes. One particular Unit Scheme could associate certain flow entities (or flow entities in general) with, for example, *acre-feet per month*, and another could represent those flow entities (or flow entities in general) with *m3* (a volume!).

This would be a radical departure from the underlying RiverWare unit system, and would require a larger development effort. Also, as described above, this could be an opportunity to replace the “altnernate units” configuration on account series slots with a single flag on all series slots indicating that the slot allows flow/volume conversions (i.e. that the entity is inherently a flow, or flow-like unit, even if its unit type is not).

While this alternative may be technically elegant and coherent, it may not be better from the user’s point of view in terms of operation, depending on the design of the GUI controls. For example, it is a lot easier to click a two-state push button (bistable multivibrator) than it is to switch between two items within a combo box. Having distinct controls for switching between Unit Schemes and Flow/Volume display (for “flow-like” entities) may be preferable.

3.3 Supporting Multiple Named Unit Schemes

A set of multiple Unit Schemes would be stored in a RiverWare model file.There could be provisions for importing and exporting Unit Schemes to copy them between RiverWare models.

The following general designs would be possible for allowing the user to switch between multiple Unit Schemes:

1. A Unit Scheme selection control (probably a “combo box”, sometimes referred to as an “option menu”) could be displayed in every slot data display. The state of that control could either be:
    - Global -- changing it’s value would effect all slot value displays. (This is recommended).
    - Local -- changing its value would effect only the local slot value display, so different slot value displays could have different Unit Schemes. If this option is chosen, probably all slot value displays should always indicate units. (See the “Other Considerations: Scale and Units are optionally indicated in the SCT” section above).
2. A single Unit Scheme selection control would be made available from the RiverWare Workspace, and would, of course, have a global effect on slot data displays.

The dialog box for editing a Unit Scheme instance could also have a Unit Scheme selection control to switch between the currently edited scheme. That control would be disabled if there were any unapplied edits to the scheme, and it should be possible to cancel unapplied edits.

There would be at least one hard-coded, non-editable Unit Scheme -- the **“Standard (RiverWare) Unit Scheme”** which shows all slot values in standard units. A new user-defined Unit Scheme could be created by duplicated the Standard Unit Scheme or any other existing user-defined Unit Scheme.

3.4 Relationships between Legacy Slot-Based Unit Configurations and Unit Schemes

The initial implementation of Unit Schemes will work in conjunction with the current Slot- and Slot-Column based unit and display property configurations. These relationships are possible:

1. Unit Scheme selection controls (probably, combo boxes) could include the following item:

    “Slot-Configured Units”

2. Within any given matching rule in a particular Unit Scheme, the “range value” of a matching rule could be “Slot-Configured Units” -- that is, in place of specific values for unit scale, unit, display precision and display format.

In fact, both of these relationships could be implemented. We could support the “Slot-Configured Units” “pseudo-Unit Scheme”, and any user-defined Unit Scheme matching *rule* could defer to “Slot-Configured Units”.

3.5 Other Effects of Unit Schemes

The currently effected unit scheme will determine how slot data is **imported and exported** to and from RiverWare in certain contexts.

Where these operations make use of the currently configured display units (using slot-based configurations), they will need to use the active Unit Scheme.

3.5.1 Open Slot Dialogs: Import and Export operations

The “Export” operation from the Open Slot Dialogs for numeric slots (e.g. not for list slots) results in the creation of a text file having the numeric values of the slot in the currently configured display units. The units of that data are indicated only in “comment” text (lines starting with ‘#’), which are ignored by the “Import” operation. When importing data into a slot, the numeric data in the text file is interpreted using the currently configured display units for that slot.

3.5.2 Output Devices

Files generated from RiverWare Output devices (and GUI displays, in the case of a standard “Plotter” output device) present numeric slot values in the slots’ currently configured display units. The value of the unit scale and unit are also encoded in the output -- in a format particular to the type of output device (e.g. RDF, Comma Separated, or Excel). These are output-only encodings; with a minor low-level exception not related to setting slot values in RiverWare, RiverWare does not read the unit scale and unit encodings from these generated files.

3.5.3 Excel Database DMI Datasets

Since Excel doesn’t associate units with cell values, settings within the RiverWare Database DMI Dataset dialog referring to “database units” aren’t actually implemented as such. Instead, the currently configured display units on the respective RiverWare slots are used for generating or interpreting Excel cell values.

4.0 Special Issues

These issues need to be considered in our development of a design for Unit Schemes.

4.1 Display Precision and Display Type

(Note: Discussion of this issue was incorporated into the “Unit Schemes” section, above).

Our initial discussions have focused on display “units” configurations. Display precision and display type (“Float”. “Integer”, or “Scientific”) configurations are currently maintained in parallel with units configurations on slots. We need to clarify the extent to which the new display configuration architecture addresses all of these settings. In particular:

- Display Precision (fractional decimal digits) is very closely related to “unit scale” and “unit” when the “Float” display type is selected (which is the usual case).
- We should consider removing the “Integer” display type. That is redundant with configuring a “Precision” of zero, and is a source of confusion.

4.2 Special support for model analysis and diagnostics

Tim and Bill have noted that it is useful to momentarily change the units or especially **display precision** (generally to “full” precision) for a particular slot being examined in the course of analyzing the behavior of a model. These perceived “requirements” need to be evaluated in the context of the new capabilities being provided. That is, in some cases, the new general way of working with units in RiverWare may change the relevance or importance of these features.

--- (end) ---