RiverWare Qt4 Porting -- 2012 / RiverWare 6.3

Open Object Dialog -- Porting Lists from Qt3 to Qt4

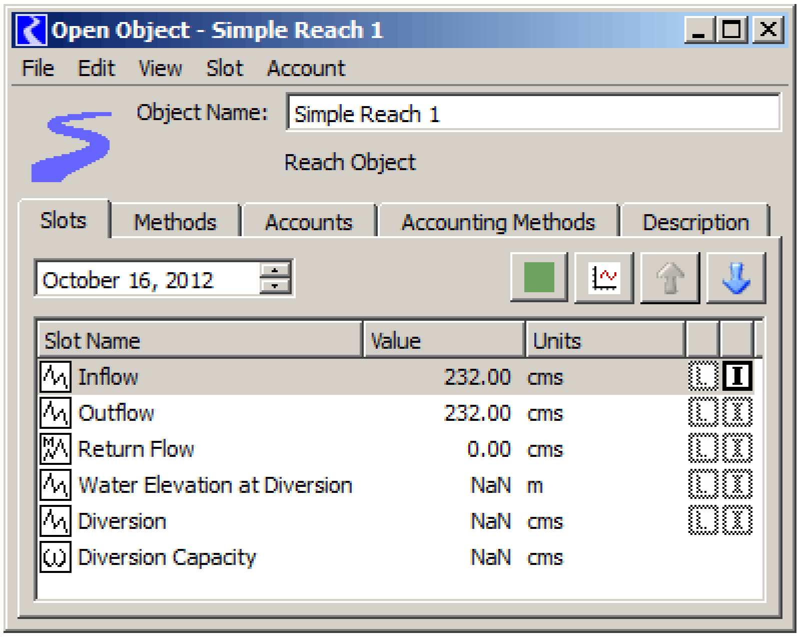
Phil Weinstein, 10-18-2012 -- Revised 10-19-2012 (A) -- PDF

Overview

The first four tabs of the Open Object Dialog have lists and trees sharing a common implementation based on Q3ListViews and Q3ListViewItems.

We will be able to port this fairly directly to analogous item-based Qt4 QTreeWidgets without much difficulty, as has been done for most of the other Qt3 lists and trees in RiverWare.

One significant change is being planned -- *replacement of drag-and-drop behavior* with *multiple item selection* -- and another is being considered -- *a redesign of the ordering controls.* These two areas are explored in this document.



Object Name: Simple Reach 1

Reach Object

October 16, 2012

Slot Name	Value	Units
Inflow	232.00	cms
Outflow	232.00	cms
Return Flow	0.00	cms
Water Elevation at Diversion	NaN	m
Diversion	NaN	cms
Diversion Capacity	NaN	cms

There is a good amount of complexity within the list implementation in this dialog. An example is the addition of top-level Simulation Object tree items for Aggregate Objects. However, these mechanisms are independent from the issues involved in replacing Q3ListViewItems with Qt4 QTreeWidgetItems. For this reason, I recommend initially porting these lists without design modifications except where both of these considerations apply:

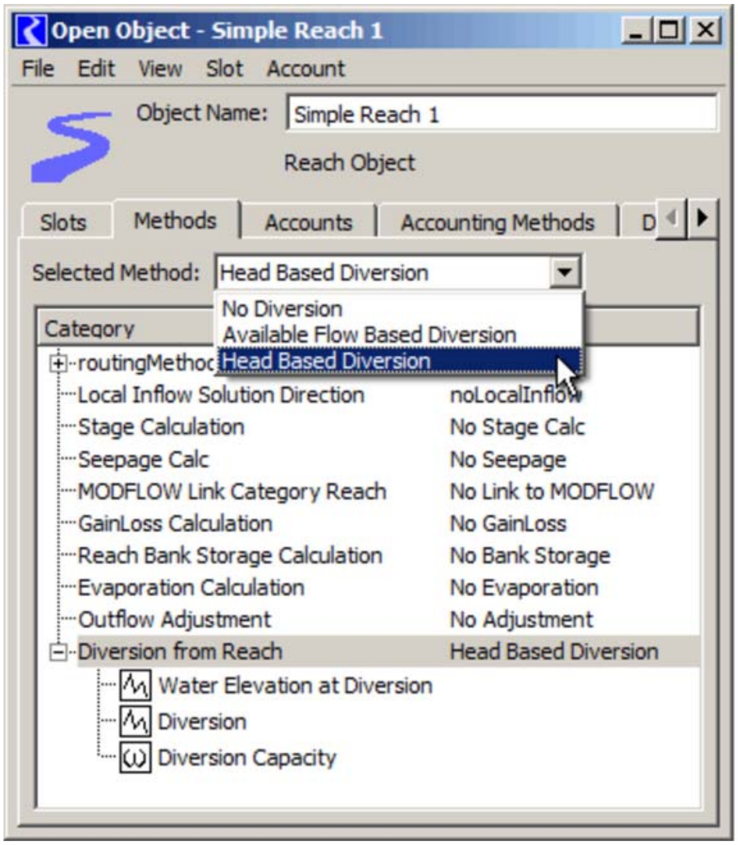
1. The aspect of the port is non-trivial. That is, the mechanism needs to be substantially rewritten for Qt4, and
2. The current functionality isn't particularly desirable.

It is for these reasons that I'm recommending that **drag-and-drop** behavior within the Slots list (first tab) be eliminated in favor of supporting **multiple-item selection**. These two features are effectively mutually exclusive: there is no standard convention for (e.g. with a modifier key) distinguishing starting a multiple item selection by dragging from initiating an item-move operation by dragging.

These considerations do also apply to the **list ordering control** design. The custom drop-down menus of the column headers (described below) do have to be explicitly coded -- though that's not as significant a consideration. In earlier discussions, I had concerns about the list ordering controls with respect to the provided capabilities and the effectiveness of the user interface implementation. However, I'm now feeling that the current functionality is reasonably coherent, and also not too difficult to implement in Qt4. This decision should depend mostly on whether we prefer some variation of the new "ordering controls" design proposed below.

Method Category Method Selection

David has proposed changing the method-category method selection controls. Currently, there is a single "Select Method" combo box which operates on the single selected category (*left image*, *below*). We could instead show a combo box on every category category row -- similar to what has recently been done for the RPL Function Argument List editor (*right image, below*).



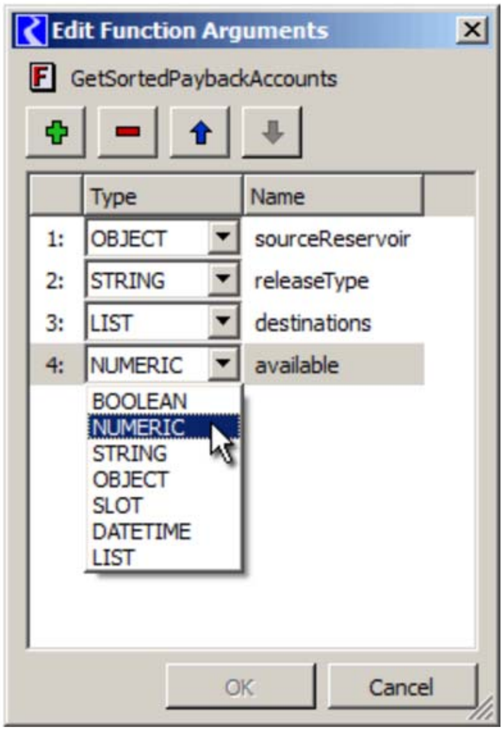
Object Name: Simple Reach 1

Reach Object

Selected Method: Head Based Diversion

No Diversion
Available Flow Based Diversion
Head Based Diversion

Category	
routingMethoc	
Local Inflow Solution Direction	noLocalInflow
Stage Calculation	No Stage Calc
Seepage Calc	No Seepage
MODFLOW Link Category Reach	No Link to MODFLOW
GainLoss Calculation	No GainLoss
Reach Bank Storage Calculation	No Bank Storage
Evaporation Calculation	No Evaporation
Outflow Adjustment	No Adjustment
Diversion from Reach	Head Based Diversion
Water Elevation at Diversion	
Diversion	
Diversion Capacity	



GetSortedPaybackAccounts

	Type	Name
1:	OBJECT	sourceReservoir
2:	STRING	releaseType
3:	LIST	destinations
4:	NUMERIC	available

BOOLEAN
NUMERIC
STRING
OBJECT
SLOT
DATETIME
LIST

I recommend that this be considered for a subsequent enhancement -- after the Qt4 port of the Qt3 lists in this dialog.

Technical note: QTreeWidget does have a provision for "persistent widgets" -- though they are supposedly intended only for non-interactive use. But I have reason to be confident that we could get that to work well. We wouldn't have to use a model-based QTree*View*.

Replacing Slot Item Drag-and-Drop with Multiple Selection

The slot list (first tab) uses special intermediate C++ classes to implement "drag-and-drop" for individually-selected slot items -- for purposes of reordering the slot items within the list. In Qt4, this would need to be rewritten, as the drag-and-drop architecture has changed. Also, however, we have decided that supporting multiple-slot item selection would be more useful.

With this port, the following operations will be enhanced to support *multiple* selection:

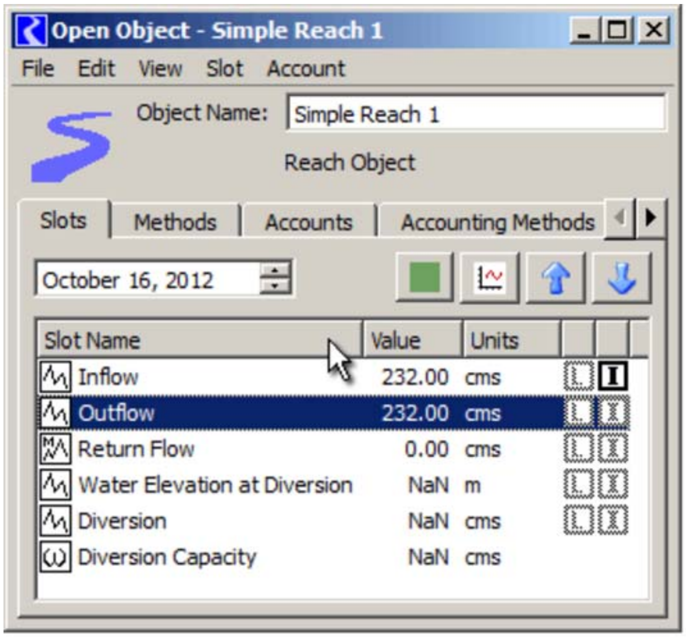
- Plot slots
- Copy slots to slot clipboard, e.g. for pasting into an output device, snapshot manager, data object, etc.
- Delete slots from a data object (with user confirmation)
- Add slots to an open or new SCT
- Move slots up or down within the custom order (for the object instance).

It may also be useful to enable multiple selection on the Methods tabs so that the slots introduced by (dependent on) a method can be copied or plotted together. Operations within the methods lists for which a multiple selection is ambiguous will be enabled only when a single item is selected, e.g. changing the method for a method category.

Existing Ordering Features (RW 6.2)

The column headers of the lists on the first four tabs of the Open Object Dialog use a non-standard convention for ordering controls.

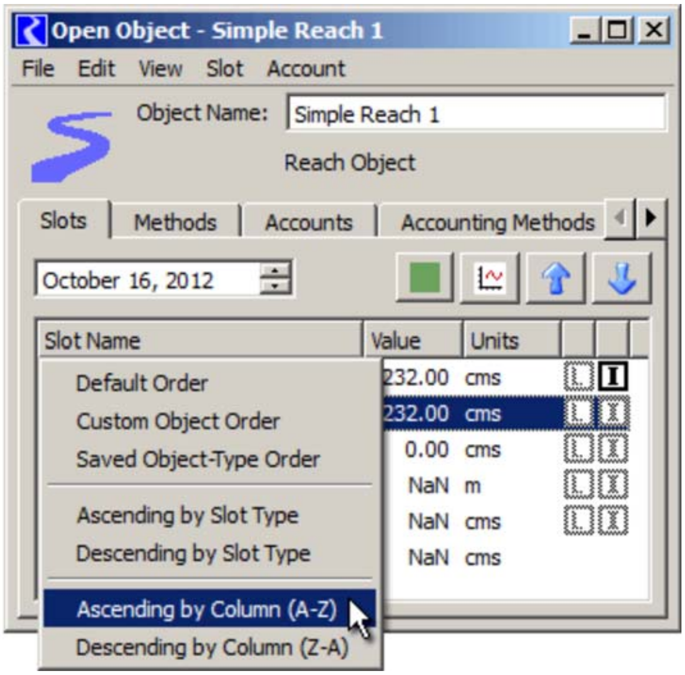
Normally, clicking on a column header sorts by that column, and clicking on it again reverses that order -- with an optionally shown triangle-arrow icon indicating the direction of the order (ascending or descending). In the Open Object Dialog, clicking on a column header instead shows a drop-down menu:


Open Object - Simple Reach 1
File Edit View Slot Account
Reach Object
Slots Methods Accounts Accounting Methods

Object Name: Simple Reach 1

October 16, 2012

Slot Name	Value	Units
Inflow	232.00	cms
Outflow	232.00	cms
Return Flow	0.00	cms
Water Elevation at Diversion	NaN	m
Diversion	NaN	cms
Diversion Capacity	NaN	cms


Open Object - Simple Reach 1
File Edit View Slot Account
Reach Object
Slots Methods Accounts Accounting Methods
Slot Name Value Units
Default Order
Custom Object Order
Saved Object-Type Order
Ascending by Slot Type
Descending by Slot Type
Ascending by Column (A-Z)
Descending by Column (Z-A)

Object Name: Simple Reach 1

October 16, 2012

The drop-down menus on the "Methods", "Accounts" and "Accounting Methods" tabs include only these items:

- Default Order
- Ascending by Column (A-Z)
- Ascending by Column (Z-A)

Clicking on any of the items in this drop-down menu changes the displayed order of slot items, as follows:

> **Default Order** is either the internally defined (hard-coded) order, or -- *in the case of the Accounts tab* -- the order in which the Accounts on the object were created.
>
> **Custom Object Order** (*Slots tab only*) is a user-defined slot order associated with the particular Simulation Object instance. This order is set each time the slot order is modified by the user -- either by dragging or using the Up and Down arrows. So, note that if you sort by a column, and click an up or down arrow, the custom order is completely changed.
>
> **Saved Object-Type Order** (*Slots tab only*) is a somewhat loosely user-defined custom order shared by all instances of a given Simulation Object type. Different instances of that type will have different slots, but each time this order is set (from the **View >> Save Object-Type Slot Order** main-menu operation) the order of the slots represented within that object is saved, and the order of the slots that

are not in that object (i.e. only in other objects) is preserved in a relative sense.

Ascending / Descending by Slot Type (*Slots tab only*) is needed because there is no column dedicated to the slot type icon. (*I recommend adding a Slot Type column to the slot list in most cases, see below*).

Ascending / Descending by Column sorts the column by the clicked-column's content.

Some problems with this current design:

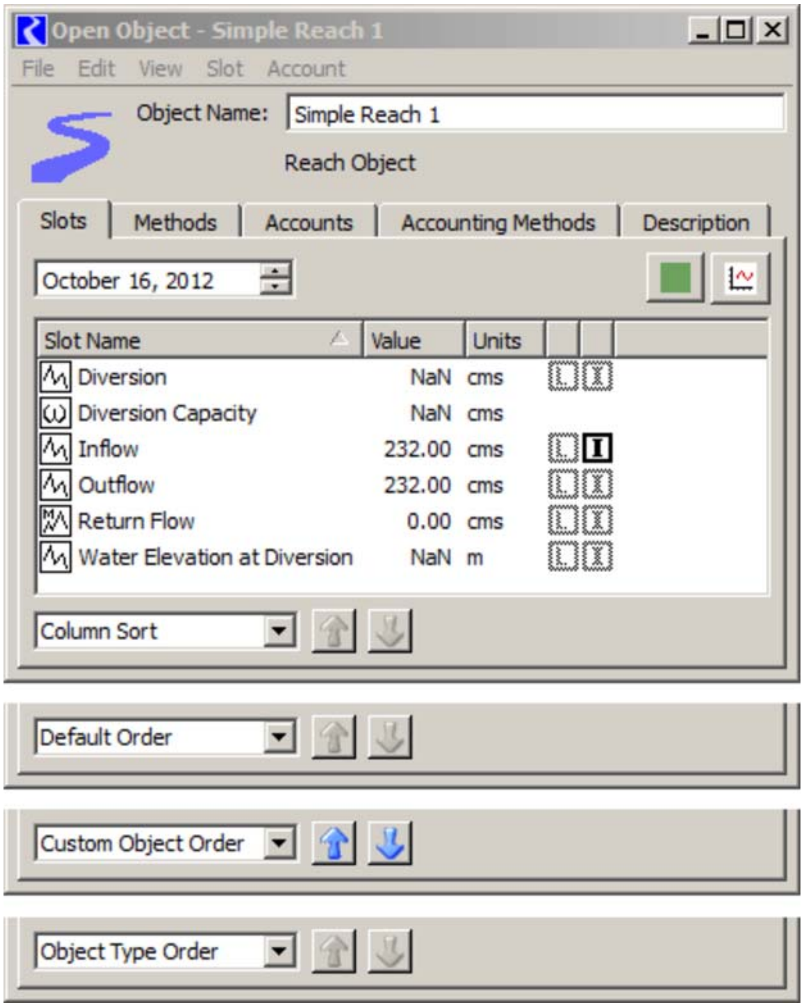
1. There is no visible indication that the list is currently displayed in one of the special orders: Default, Custom Object, or Saved Object-Type orders.
2. It is too easy to change the Custom Order, as mentioned above.
3. The "Save Object-Type Slot Order" menu item is in a very different place than the (Select-) "Saved Object-Type Order". This makes this feature a little difficult to understand.

Minimal Recommended Change to the Ordering Controls

In the course of the Qt4 port, I do recommend making at least this minor change.

Except in the case of the Open Object Dialog for **Aggregate Objects** (which shows top-level tree items for the overall AggObj and each of the Element objects), in the **Slot list** (on the Slots tab) we could show a **distinct initial column for the Slot Type icon,** and remove the "Ascending / Descending by Slot Type" column header drop-down menu. Clicking on the column header for that column would sort the slot list by slot type, or reverse that order.

Ordering Control Redesign



Object Name: Simple Reach 1

Reach Object

October 16, 2012

Slot Name	Value	Units
Diversion	NaN	cms
Diversion Capacity	NaN	cms
Inflow	232.00	cms
Outflow	232.00	cms
Return Flow	0.00	cms
Water Elevation at Diversion	NaN	m

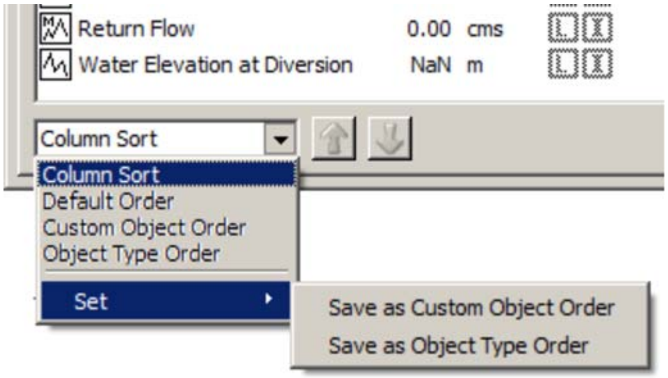
We could revert the column headers within these lists to the standard "click" behavior: sorting by the column content, and reversing that order. Additional GUI controls would be introduced to choose a special order, and to assign the settable custom orders. **This would address the problems with the current design cited above.**

In the Slot tab, a combo box would be shown below the list, and the Move Up and Move Down arrow buttons would be placed next to that. The arrows would be enabled only when "Custom Object order" was selected within the combo box (and also when they are individually valid depending on the current slot item selection).

The combo box would have a custom-implemented popup menu supporting a "Set" submenu to set the two settable orders from the currently displayed order.

- Column Sort
- Default Order
- Custom Object Order
- Object Type Order
- ---------------------
- Set (*submenu* ...)
  - Save as Custom Object Order
  - Save as Object Type Order

(Note that the following image is a "Photoshopped" mockup) ...



Return Flow	0.00	cms
Water Elevation at Diversion	NaN	m

Clicking in a slot list column header would force the combo box to the "Column Sort" item.

In the **tabs other than the Slot tab,** all that is needed is a **"Default Order" button** (*not illustrated here*). I recommend that this be deployed in the same place as the more complicated controls needed for the Slot tab's list -- i.e. below the list. The "Default Order" button would be enabled only if the displayed list order was not currently in that order. So, clicking on the button immediately disables it. It becomes enabled again as soon as the user sorts by a different order -- i.e. by clicking on a column header (assuming that that column sort does not match the default order).

Note about "Non-destructive" sorting supported by Qt4

Qt4 QTreeViews and QTreeWidgets now support, by default, *non-destructive* sorting. This means that the relative order of items is not changed for items having the same value in the clicked column. This allows the user to sort by a sequence of chosen columns to perform a multiple-level sort.

We haven't yet decided, in general, to adopt the "non-destructive" sorting convention. Our convention is still to defer to particular other columns when two items have equal values in the clicked column. (Qt4 made this "fully deterministic" sorting somewhat more difficult to implement -- *I will add, without explaining in the documentation what was going on here*).

I do remember that some visiting water engineers recommended that we provide non-destructive sorting for our lists (a few years back). We should consider making a decision about adopting and consistently implementing that convention.

--- (end) ---