

# Object Attributes - Requirements and Design

CADSWES: David Neumann, Patrick Lynn, Neil Wilson, Phil Weinstein and Edie Zagona

## 1.0 Introduction

FY14 Task 11 for Reclamation Upper Colorado lists the following as part of the scope of work:

“...enhancements to allow for attribute assignments to objects (e.g. use type, tribal/nontribal, state, basin, etc.) to be set through the GUI or via input DMI, to offer easy selection, summary, and visualization of model data;”

This document describes requirements, existing functionality and conceptual design for this task.

Reclamation is currently using the Demand Input Tool (DIT) to define demands as input to CRSS. Within the baseline configuration, they define Attributes and their possible legal values for Water Users and Agg Diversion sites. Within the scenario development, they can set the Attribute's value for each object. Examples are shown in the table:

Object	Attribute	Possible Values
Agg Diversion Site	Node	1.Glenwood, 2 Cameo, 3. Taylor Park, 5 Crystal Res, ...
	Tributary	Upper Colorado, Gunnison, Dolores, Yampa, White, ...
Water User	State	Arizona, California, Colorado, New Mexico, Nevada, Utah, Wyoming
	Sector	Agriculture, Energy, M&I, Minerals, Fish & Wildlife, ...
	Depletion Class	Anticipated, Current, Potential
	Tribe	Yes or No
	WQIP	Yes or N
	Planning Area	West Colorado, Southeast Colorado, Uintah Basin, Kanab, ...

Reclamation develops the demands based on these attributes/values. The demands are set in the Excel File and then imported into RiverWare via an Excel Database DMI.

Within RiverWare, there is no robust support for these attributes. There are Water User Groups and Subbasins, described below, but these are hard to use and have limitations.

Reclamation would like better support for attribute and values in RiverWare.

## 2.0 Requirements

RiverWare should be enhanced to have some formal attribute and value mechanism. Following is a bulleted list of requirements:

- There can be many attributes defined in a model, each has one or more possible values. A particular object can have zero or more attributes, but for each attribute, only ONE value is possible.

- 
- An Attribute can be specific to a particular class of objects (reach, water user, reservoir, power reservoir) or all objects.
  - Attributes and values are strings (i.e. not numbers or boolean).
  - Attributes and values need to be displayed on the Open Object dialog. The user should be able to select the value for each attribute directly from the Object. Possibly, Attribute/Values should show as tooltips or in the status bar of the workspace.
  - Attributes and Values need to be displayed and editable from some central location or manager.
    - The user should be able to create, edit, and rename the attributes from this manager.
    - The user should be able to define the possible values for each attribute and the object types for which an attribute is available.
    - The order of attributes and of values should be editable by the user. Wherever these strings are shown in other GUI components (e.g. in comboboxes), they will be listed in this order. In sortable displays, the user-defined order is the “default” order.
    - The user must be able to assign values to one or more objects from this manager.
  - Attributes need to be accessible from RPL so computations can be based on the attribute (i.e. Sum all CO M&I demands).
  - The Object Selector should have a filter to assist with selecting objects: HasAttributeValue.
  - The user should be able to export/import the attributes and values from one RiverWare model to another.
  - Attributes can be defined in the DIT and assigned to water user and agg diversion sites in the DIT. There must be some way to transfer them to RiverWare (likely via export/import). This is an infrequent activity that would occur during model/study setup. The transfer of values is not needed within RiverSMART, DMI or other automated connection.
  - Although not a requirement at this point, Attribute/Values may be used for workspace visualization or summary. Perhaps the user could change the color of all the water users that meet some attribute value criteria? Or easily sum and display a plot of all slots on objects that meet some Attribute/Value criteria.

---

### 3.0 Existing Functionality

In past efforts, subbasins have been used to group objects geographically, by a common attribute, or just out of convenience. There are standard subbasins and typed subbasins including **Computational**, **Smooth Energy**, **No Optimization** and **Allocated**. Within RPL, the user can get a list of the objects in a subbasin through the ListSubbasin(“subbasinName”) predefined function. The Subbasin Manager allows users to highlight the objects on the workspace and add/remove objects from the subbasin. There are both User-Defined and Automatic (based on object class) subbasins. The Object Membership tab in the subbasin manager allows the user to query which subbasin a particular object belongs to.

In RiverWare 6.1, the Water User Groups slot was added to Water Users. This List Slot shows the user-defined subbasins to which that water user belongs. They are not editable from the List Slot,

---

only from the subbasin manager. There is no way to import them from the DIT. The slot is only on Water Users, not on Agg Diversions (or any other object)

There are a few other existing areas in RiverWare that may or may not be applicable to this effort:

- Accounts and Supplies have attributes. Accounts have *account type*, *water owner*, *water type*, and *priority date*. Supplies have *supply type*, *release type*, and *destination*. The legal values are defined in the Account System Configuration. Both Accounts and Supplies have a manager where the attributes can be viewed and values set. These managers may serve as a template for a new Object (or Attribute Manager).
- The Multiple Object Method Selector has an interface to select objects and then view the selected method (Value) for a particular category (Attribute). The user can then select one or more object and apply a new method to those objects. (This is not the most user-friendly interface though).
- Object Display Groups are sets of objects on the workspace used to change the appearance on the workspace.
- Object Clusters are groupings of objects on the workspace that appear as only one icon.

---

## 4.0 Design

Following are the five main development areas and the structure for this design:

- Architecture, data structure, import/export: How are attributes represented, stored, exported/imported within the RiverWare code.
- Objects Attribute Manager: What is the user interface for the new Manager? (Objects Manager is analogous to the Accounts Manager and Supplies Manager)
- Open Object Dialog - What changes must be made to the Open Object Dialog and other interfaces?
- DIT export- How will information be exported from the DIT?
- RPL Functions - What are the new RPL functions, operators, etc...

The following sections describe the requirements and design. They are written in present tense so they can be easily incorporated into other documents

### 4.1 Architecture

#### 4.1.1 SIMOBJ ATTRIBUTES

The SimObj class will be extended to support representation of a set of attribute value pairs and supporting methods:

- Return a list of the existing attributes
- Return the value for a given attribute
- Set the value for a given attribute
- Serialize the attribute/value pairs (presumably as XML)
- Add a set of attribute/value pairs from a given serialization

Probably the best data structure for supporting these methods is a map.

---

The serialization should work for both the model files as well as import/export of attributes.

#### 4.1.2 OBJECT ATTRIBUTE MANAGER

Create a new singleton class **ObjAttributeMgr** which is responsible for model-wide management of attributes and values. This class will maintain the set of triplets which define the valid attributes for the current model, i.e., the set of <attribute, set of SimObj types, set of value> triplets. The **ObjAttributeMgr** will constitute the primary Sim library interface for the Object Attribute Manager dialog, so its methods will include:

- List the attributes supported by the current model
- For a given attribute, return the valid SimObj types
- For a given attribute, return the valid values
- For a given object and attribute, return the value
- Serialize the manager data (set of triplets supported by the model)
- Serialize all attribute value pairs
- Read a serialized set of attribute/value pairs, return the contents in an appropriate data structure.
- Check whether a given attribute is supported for a given object
- Check whether a given value is supported for a given attribute
- Add a set of attribute/value pairs
- Add an attribute to a set of SimObj types
- Add/Remove a value from the valid set of values for a given attribute
- Remove an attribute from a set of SimObj types
- For an object, set the default set of attributes/value pairs (called by SimObj constructor)

There are several analogous managers in RiverWare (e.g., UnitSchemeMgr) any of which could be used a software model. Serialization will be as XML, in a schema that is known also by the DIT (Demand Input Tool).

The primary decisions to be made are policy decisions involving validity, e.g.:

When an attribute is removed, what happens to

#### 4.1.3 SERIALIZATION

Two types of object attribute information will be written and read, what I'll call metadata and values:

- Object attribute metadata:
  - ordered list of valid attributes
  - for each attribute: ordered list of valid values, the default value, and the set of SimObj types to which the attribute applies
- Object attribute values:
  - For each object: the set of attribute value pairs for that object

The metadata are owned by the ObjAttributeMgr; the values are owned by the individual SimObjs. The model file will contain both types of information, presumably the owner of the data will be

---

responsible for writing it. On the other hand, it might make sense for reading to depend on the context. Exporting from the Object Attribute Manager dialog will write both types of information to a single file, which can later be imported by RiverWare. Export/import of a SimObj will include its values. The DIT will export only values, so import of those alone will be supported by the manager dialog. So while the object attribute metadata will be exported and imported by the manager, the object attribute values will be exported by the DIT and by SimObjs but imported by SimObjs and the manager.

## 4.2 Object Attribute Manager (dialog)

A new **Object Attribute Manager** is accessible from each **Open Object** dialog and from the **Workspace>>Objects** menu. There are two main purposes in this dialog, managing the attributes and applying values to objects. Perhaps these are two tabs on the dialog with the relevant tab opening depending on how it is accessed. Alternatively, it could be one big dialog with two regions or two completely separate dialogs.

### 4.2.1 MENUS

The file menu has options to

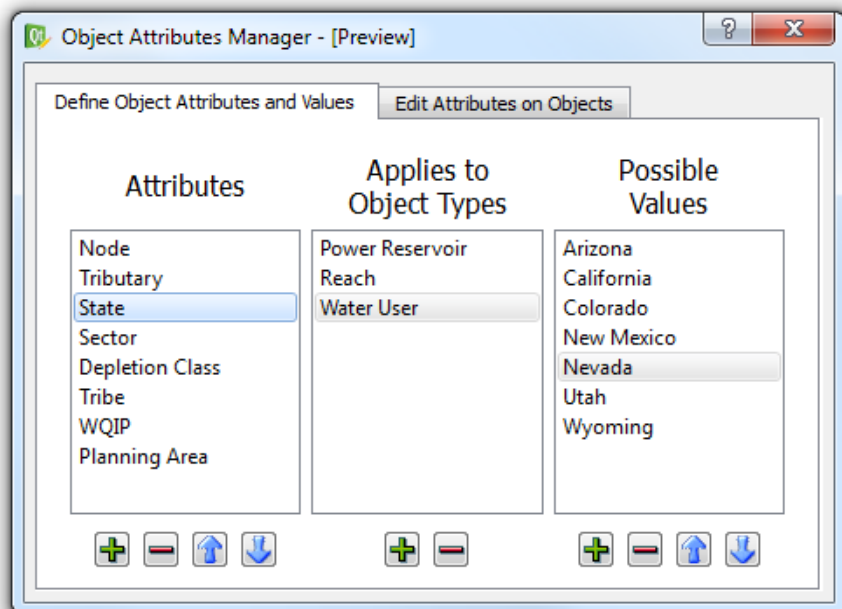
- Export and import attributes and values as described above.
- Show Workspace
- Close Window

### 4.2.2 DEFINE ATTRIBUTES AND VALUES

This portion of the dialog is used to define attributes and their legal values. There is no limit to the number of attributes or number of values for a given attribute.

This dialog image illustrates a possible design of the “Define Object Attributes and Values” tab within a tabbed layout of the Object Attribute Manager dialog. (Its menubar is not shown).

This part of the dialog has three list views supporting multiple item selection. These listviews are described left to right:



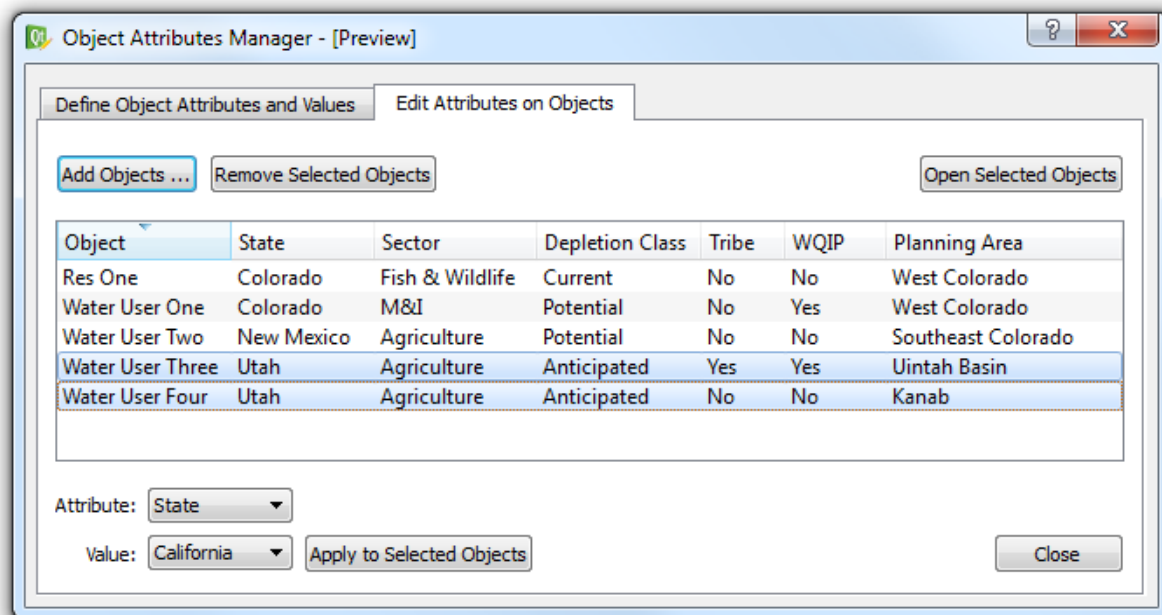
- **Attributes:** Add or delete an attribute using + and - buttons. Rename by double clicking the name. Move the selected items up or down within the list with the arrow buttons.
- **Applies to Object Types:** For the selected Attribute, add or remove the object types to which this attribute applies. Add or remove a type of object using + and - buttons. Object Types are always listed in alphabetical order.
- **Possible Values:** Define the legal values for each attribute. For the selected attribute, this right-most list shows the legal values. Add or delete a value using + and - buttons. Rename by double clicking the name. Move the selected items up or down within the list with the arrow buttons.

Context menus also provide add (or insert) and delete (or remove) functions, as well as copy and paste (for text to and from the system clipboard).

### 4.2.3 VIEW AND SET ATTRIBUTE VALUES

This section of the Object Attributes Manager dialog displays the attributes and value for a selection of objects, and supports applying new values to all or some of those objects.

The dialog image below illustrates a possible design of the “Edit Attributes on Objects” tab within a tabbed layout of the Object Attribute Manager dialog. (The dialog’s menubar is not shown).



The following features are supported:

- Add objects to the list of objects by clicking on the **Add Objects** button. Each object is added as a row in the table.

- 
- Remove the selected objects in the list by clicking the **Remove Selected Objects** button.
  - Use the **Open Selected Objects** button to open the selected set of objects. If there are more than 5, a confirmation is presented before opening them. When showing Open Object dialogs in this way, their Attributes tab is selected.
  - Optional: There could be menu operations to highlight objects on the workspace (from the selected object items in the list) and take a workspace selection and add those objects to this list.
  - Each column in the list represents an attribute. The values are then shown on each row.
  - If an attribute does not apply to a selected object, the cell is left blank.
  - Columns are sortable and rearrange-able. Optional: There could be a way to filter or hide columns that the user does not wish to see.
  - Use the right-click context menu to copy a list of objects, attributes, values to clipboard.
  - To change the value of a particular attribute for one or more objects, select the objects and then select an attribute and a value from the two pulldown menus (comboboxes) at the bottom of the dialog. Click **Apply to Selected Objects** to apply that value to those selected objects. This button is enabled only if all of the selected object items (rows) support the currently selected attribute.

### 4.3 Open Object Dialog

The **Open Object** dialog displays attributes and values in a new **Attributes** tab. (Alternative idea: change the Description tab to a General tab which then contains both the description and the attributes).

- The tab has one row per attribute defined for that class of object.
- Sort by columns using either the attribute or value. This is similar to the Methods tab..
- The row lists the Attribute name and the value. A pulldown menu (combobox) ON THE ROW allows the user to select a different value for that attribute. No apply is necessary.
- Use the **Open Attribute Manager** button, tool bar **View >> Object Attributes Manager** menu, or right-click context menu to open the **Object Attribute Manager**. This adds the object (and its element object, in the case of an Aggregate Object) to the manager's object list, and causes (only) those items to be selected.

Question: Is there any kind of hierarchy or organization to attributes. For example, are some attributes “internal” or “automatic”?

The dialog images to the right and below illustrate this design (using an Attributes tab) for a Water User object and an Agg Diversion Site (with two Water User element objects).

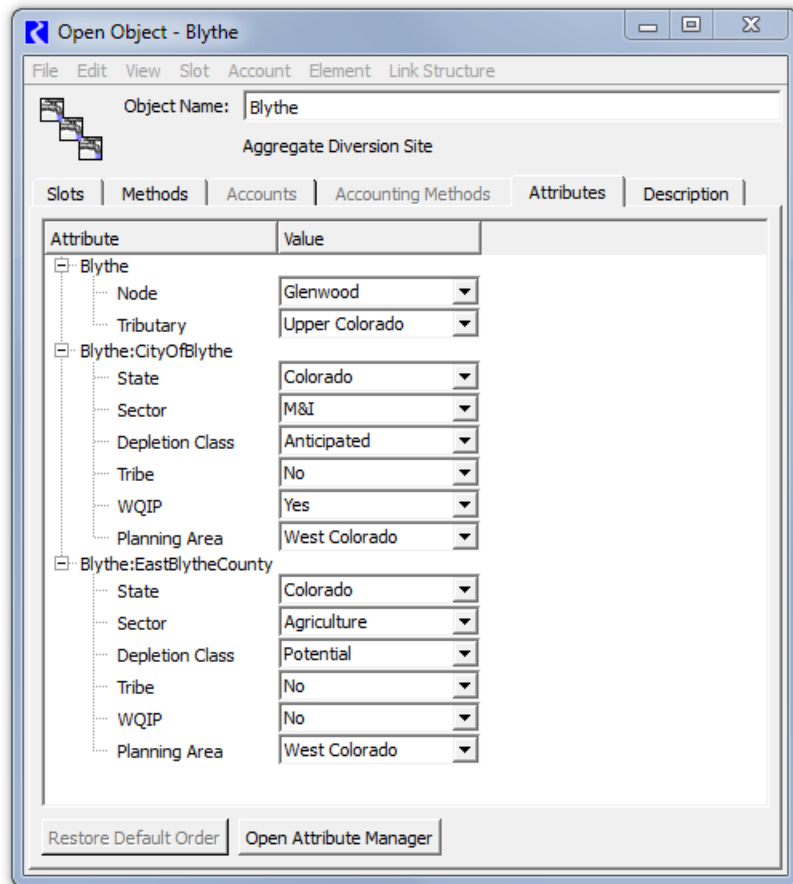
In the case of an Aggregate Object (e.g. the image below), Attribute / Value pairs are organized in a treeview for the overall Aggregate Object and each of the element objects -- similar to the presentation of those objects on the other tabs in this dialog.

The screenshot shows a window titled "Open Object - Userville" with a menu bar (File, Edit, View, Slot, Account). Below the menu bar is a toolbar with a water tap icon. The main area displays "Object Name: Userville" and "Water User Object". There are several tabs: Slots, Methods, Accounts, Accounting Methods, Attributes (selected), and Description. The Attributes tab contains a table with two columns: Attribute and Value. The table lists six attributes with their corresponding values in dropdown menus.

Attribute	Value
State	Arizona
Sector	M&I
Depletion Class	Potential
Tribe	No
WQIP	Yes
Planning Area	West Colorado

At the bottom of the dialog, there are two buttons: "Restore Default Order" and "Open Attribute Manager".





#### 4.4 DIT export

#### 4.5 RPL Functions

Following are the initial set of RPL Predefined functions that will be implemented to access Object Attribute Values.

---

#### 4.5.1 OBJECTS FROM ATTRIBUTE VALUE

<b>Description</b>	Given a string representing an attribute and a string representing a value, return a list of all of the objects that have that value for the attribute	
<b>Type</b>	LIST {OBJECT, OBJECT, ...}	
<b>Arguments</b>		
<b>1</b>	STRING	The name of the Attribute
<b>2</b>	STRING	The Value of the that Attribute.
<b>Evaluation</b>	When evaluated, the function looks throughout the model and finds all objects that have the Attribute Value pair. The set of matching objects is returned in a list.	
<b>Comments</b>	If the attribute or value is not found in the model, an error will be issued.	

Syntax Example:

```
ObjectsFromAttributeValue("State", "Colorado")
```

Return Example:

```
{"Arkansas", "RioGrande", "SanJuan", "Dolores", "Gunnison"}
```

#### 4.5.2 OBJECT HAS ATTRIBUTE VALUE

<b>Description</b>	Return whether the particular Object has the specified Attribute Value.	
<b>Type</b>	BOOLEAN	
<b>Arguments</b>		
<b>1</b>	OBJECT	The Object
<b>2</b>	STRING	The name of the Attribute
<b>3</b>	STRING	The Value of the Attribute.
<b>Evaluation</b>	When evaluated, the function looks at the particular object and checks to see if it has the given Attribute and Value.	
<b>Comments</b>	If the attribute or value is not found on the object, FALSE is returned	

Syntax Example:

```
ObjectHasAttributeValue("Dolores", "State", "Colorado")
```

Return Example:

```
TRUE
```

---

#### 4.5.3 OBJECTATTRIBUTEVALUE

<b>Description</b>	For the specified Object and a string representing an attribute, return the value for that particular Object, as a string	
<b>Type</b>	STRING	
<b>Arguments</b>		
<b>1</b>	OBJECT	The Object
<b>2</b>	STRING	The name of the Attribute
<b>Evaluation</b>		
<b>Comments</b>	If the attribute is not found on the object, an error is issued.	

Syntax Example:

```
ObjectAttributeValue("%Dolores", "State")
```

Return Example:

```
"Colorado"
```

#### 4.6 Misc Enhancements

Attributes should show up in a model report in a new Object item. It would show the attributes and values. We should also show the Object's description for this item which is not shown elsewhere.

---

### 5.0 Additional Ideas

Attributes could be expanded to include predefined or internal attributes. Following is a table of some ideas of RiverWare items that could fit into an attribute / values framework:

Attribute	Legal Values
Object Type	Reservoir, Level Power Reservoir, Reach, ...
Category (E.g. Routing)	Method (E.g. No Routing, Time Lag ...)
Subbasin (E.g. Upper Basin)	In, Out

To meet the current needs, attributes map to a set of strings, but it is easy to imagine other value types such as integer or unconstrained string (i.e., values are strings but are not limited to a given set of possibilities). The design should be extensible to other value types.