Analysis: User configurable default supply name composition
Phil Weinstein -- 4-16-2010, revised 4-20-2010.

Original Request:

User defaults for supply names

Revisions:

* 4-20-2010: Added **"[x] Remove all spaces"** checkbox in the "Options" section.

Discussion:

In RiverWare 5.2 (and prior versions of RiverWare), default Supply names are composed from the names of the upstream and downstream Accounts, plus a standard suffix for Diversion/Return and Transfer Supplies, of the following form (with spaces between the components):

* <upstream object name> <upstream account name> to <downstream object name> <downstream account name>
* <upstream object name> <upstream account name> to <downstream object name> <downstream account name> Div
* <upstream object name> <upstream account name> to <downstream object name> <downstream account name> Tran

To support the current default supply names, it would be necessary to support sequences of "name parts" defined with these components:

* Upstream object name
* Upstream account name
* Downstream object name
* Downstream account name
* String: "to"
* Supply type strings: "Div" and "Tran"

It may also be reasonable to support name parts for these components:

* Account Water Type
* Account Water Owner

It may also be desirable to support name parts for these components -- but we would have difficulty supporting these because default names are constructed before these properties are assigned to the Supply. (In particular, these two Supply properties are defaulted to "None" when creating a Supply from the Accounting View using context menu operations). **I propose that we do not support name parts for these components:**

* Release Type
* (Supply-) Destination

Additionally, we could support condensing name parts which have internal spaces. That is, there could be an option to remove internal spaces and forcing capitalization of the characters that had been preceded by a space.

|  |  |
| --- | --- |
|       | http://cadswes2.colorado.edu/%7Ephilw/2010/Accounting/DefaultSplyName/Mockup1B.gif |

User Interface Proposal:

This mockup presents GUI controls for supporting the configuration capabilities discussed above.

With the use of the "More" and "Less" buttons, user would be able to grow or shrink the number of separate name parts used to compose the default name for a Supply. Each such name part is defined with one row of controls, including a ComboBox for the predefined components, and a LineEditor to specify a text string for the part.

The ComboBox would have these items:



The user can specify the text strings used for the Supply Type Code of each of the three types of Supplies.

The user can choose to have spaces removed from name parts (e.g. an Object Name) and force capitalization of the characters that had been preceded by a space (also known as "Camel Casing").

The use can choose to have all spaces removed (i.e. including in between name parts).

The user can choose a particular existing Supply instance to be used as a *sample* to illustrate the configured default naming convention.

The user can restore the default Supply name convention to the currently implemented default (which is illustrated in the accompanying mockup screenshot).

**Handling Name Conflicts:** An integer will be appended to the default supply name if there is already an existing Supply having that name. This will occur if the number of, and types of name parts are not sufficient to distinguish a newly created Supply from existing Supplies. Of course, such integer suffixes with be generated in the required way, i.e. being incremented to one above the largest existing suffix for a particular Supply name.

Development Tasks and Estimate

|  |  |  |
| --- | --- | --- |
| **DevDays** | **...(optional)**  | **Description** |
| 1.5 |   | Data model (C++ classes representing a specification), including functions to compose a default name from a Supply instance, and to build a "factory default" state. |
| 1.5 |   | Dialog development: UI-based construction, dynamic Supply Name Part "widget row" mechanism, populating dialog with instance data. Note: I recommend doing this as a QMainWindow (with a menubar) rather than just a QDialog especially if the optional features are planned (see below). [Here is the Mockup UI File](http://cadswes2.colorado.edu/~philw/2010/Accounting/DefaultSplyName/DefaultSupplyNameMgrDlg.ui) (Qt4 Designer). |
| 0.5 |   | Dialog development: Saving widget states to instance data. |
| 1.0 |   | Data model serialization (saving in RiverWare file). |
| 0.5 |   | Addessing Supply-name conflicts |
|   | + 1.5 | **Optional**: User preferences serialization, to be applied to newly created models and possibly, explicitly saved/loaded ("Save/Load to/from user settings"). It would be best to use a single-string serialization so as not to overly "pollute" the registry with a variable number of multiple records. |
|   | + 1.0 | **Optional:** Import / Export Supply Name Specification (for conveyance between existing models) via a data file. This could use the multiple-line Tcl serialization used for the model file serialization. |
| **5.0** | **+ 2.5** | **TOTALS** |