

RiverWare Output Canvas: Teacup Storage and Flow Animation: Development Plan
Phil Weinstein, David Neumann, Patrick Lynn, Edie Zagona, CADSWES, 8-27-2014 (b)

(0) Overview

This document presents a *development plan* for the RiverWare "**Output Canvas**" output device described in this design document:

- **RiverWare Output Canvas: Teacup Storage and Flow Animation: Design 3**
Word: R:\doc\Output\OutputCanvas\2014\OutputCanvDesign3-2014-08-25.docx

Note [8-27-2014]: We may want to consider overlaying a full-blown Wizard feature for Teacup creation -- including setup of markers and integrated text items within the Teacup Group. The "Create Multiple Operation (Wizard)" feature described in section 5.1 would be just one part of that. Development of the currently designed feature is being put off (moved from Phase 2 to Phase 5) until we more broadly consider what the high-level features should be.

(0.1) Document Status

- 8-27-2014: Development Estimate.
- 8-27-2014: Change: Moving the advanced Multiple-Teacup Creation feature to after completion of Text Items and Images. See note above.
- 8-26-2014: Ready for content review.

(0.2) Contents

1. Development Phases
 1. Basic Teacups, without integrated text items -- PROTOTYPE
 2. Basic Teacups, without integrated text items -- PRODUCTION FEATURE
 3. Text Items in Teacups and on canvas
 4. Canvas Images
 5. Create Multiple Teacups Operation
 6. Flow Lines
2. Advanced Features
 1. Inclusion of Output Canvases in Model Reports
 2. Animated File Generation
 3. Customized rwSetting ValueType support for timestep date/times.
3. Development Estimates

(1.0) Development Phases

This feature may be developed in the following phases, initially for RiverWare 6.6:

1. Basic Teacups, without integrated text items -- PROTOTYPE
2. Basic Teacups, without integrated text items -- PRODUCTION FEATURE
3. Text Items in Teacups and on canvas
4. Canvas images
5. Multiple Teacups Operation
6. Flow Lines

Advanced Features (can individually be done as part of prior phases).

1. Inclusion of Output Canvases in Model Reports
2. Animated File Generation
3. Customized rwSetting ValueType support for timestep date/times.

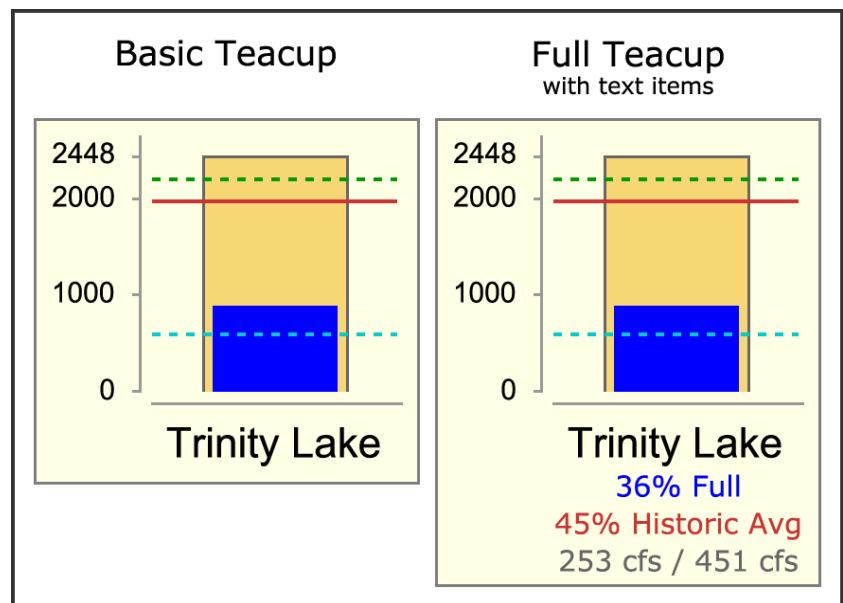
(1.1 and 1.2) Phases I and II: Basic Teacups, without integrated text items.

- Phase I -- PROTOTYPE
- Phase II -- PRODUCTION
FEATURE -- Development items are tagged with (**).

This phase provides a usable Output Canvas output device type with simplified "Basic Teacup" graphics (lacking integrated text items). The user will be able to create multiple Output Canvas output device instances and add Teacups for individual simulation objects (and related data objects) and a basic Teacup Legend.

This includes these Teacup graphical features:

- "Maximum" Outer Rectangle
- "Current" Inner Rectangle
- Multiple Marker Lines
- Axes, vertical axis has automatically placed, labeled tick marks.
- Teacup Label
- Bounding Box*



*The Bounding Box is included among the basic features because it is a graphical depiction of a fundamental property Qt graphics items used to implement the Teacup graphics. This will be useful in development diagnostics.

The Output Canvas "Add Item" Action Menu will support these actions:

1. Add Teacup Group
2. Add Teacup Legend ... *to Teacup Group*
3. Add Bounding Box ... *to Teacup Group*
4. Add Teacup Marker ... *to Teacup Group*
5. Add Teacup ... *to Teacup Group*

Phase I Tasks:

Note: Enhancements to rwSettings ValueTypes -- for property types not currently supported -- e.g. timestep date/time can be put off until later. Support for such types can initially be provided using plain text or enumerated types.

(A) Basic Data Model and Configuration GUI Support

1. Define rwSettings Identifiers for each of the "Phase I" Output Canvas Config classes.
2. Data Model: class definitions for (a) Output Canvas Output Device (b) Output Canvas Config, containing ... (c) Teacup Group, (e) Teacup, (e) Bounding Box, and (f) Teacup Marker, (g) Teacup Legend. This includes rwSetting list definitions for most of these classes.
3. Management of Output Canvas output devices, including device data *persistence*.
4. Mostly stubbed Output Canvas Configuration Dialog.
5. Object Tree (QTreeView), including Qt Abstract Item Model implementation based on an Output Canvas Config instance.
6. "Add Item" Action Menu with the five operations enumerated above.
7. Deployment and integration of an rwSettingsTree in the Settings property editor panel.
8. Application of rwSettingsTree changes to the Output Canvas Config instance data.

(B) Basic Canvas / Graphics View Implementation

1. Basic Qt Graphics View subclasses (of RwGraphicsScene and RwGraphicsView) to implement the canvas, deployed within the Output Canvas Config Dialog as the "preview" panel.
2. Stubbed development of the Teacup and Teacup Legend Graphics Items, with just dummy (icon) display.
3. Basic canvas operations: Teacup item selection, moving (by dragging).
4. Reference Timestep GUI support: Deploy Date/time spinner / animation controls.
5. (**) Heuristic initial Teacup placement algorithm, based on relative SimObj positions within the simulation view.

(C) Teacup Graphics Items

1. Data source slot reference support, resolution of data source slots for the various graphic entities.
2. Geometry computations and vertical scale composition.
3. Graphics rendering for each of the graphical features listed above.
4. (**) Teacup context menu, operations.

(D) (**) Teacup Legend Graphics Items

1. (**) Geometry computations: vertical positions determined by label text rows.
2. (**) Graphics rendering modifications for legend-specific geometries

(E) Output Canvas Viewer Dialog

1. Deployment of simplified version of canvas in viewer dialog.
2. Function: Export as Image File ...
3. (**) Slot Data Display Functions:
 1. Plot... (all entities of selected Teacups)
 2. Show slots in New SCT ...
 3. Show slots in Open SCT ...
 4. Open <object> ...
 5. Open <data object> ...
 6. Open Current Value Slot
 7. Open Marker Slot >> ...

(F) (**) Special Data and GUI Maintenance Issues

1. (**) Handling of item positions when reducing canvas dimensions.
2. (**) Responding to SimObj name changes.
3. (**) Other (overlooked issues).

(1.3) Phase III: Text Items in Teacups and on canvas

This phase provides addition of Text Items to the canvas (in Text Groups) and to Teacups (as integrated text items). The Output Canvas "Add Item" Action Menu will support these additional actions:

1. Add Text Group
2. Add Text Item: Plain ... *to Text Group*
3. Add Text Item: Timestep ... *to Text Group*
4. Add Text Item: One Slot Value ... *to Text or Teacup Group*
5. Add Text Item: Two Slot Values ... *to Text or Teacup Group*
6. Add Text Item: Computed Fraction ... *to Text or Teacup Group*

Phase III Tasks:

(A) Basic Data Model and Configuration GUI Support

1. Define rwSettings Identifiers for Text Groups and Items
2. Data Model: class definitions for (a) Text Group and Text Item, including rwSetting list definitions.
3. Object Tree (QTreeView) and Settings support for Text Groups and Text Items.
4. "Add Item" Action Menu: implement the six operations enumerated above.

(B) Canvas Text Items (in Text Groups)

1. Text composition (of the non-Plain text item types)
2. Geometry computations
3. Text Graphics Items / Text drawing
4. Text Item context menu, operations: delete, configure.

(C) Canvas Text Items in Teacups

1. Text composition enhancements (using the Teacup's context)
2. Geometry computations (including revisions to Teacup geometry)
3. Text drawing (as part of the Teacup graphics)

(D) Special Data and GUI Maintenance Issues

1. Handling of text item positions when reducing canvas dimensions.
2. Responding to SimObj name changes.
3. Other (overlooked issues).

(1.4) Phase IV: Canvas Images

This phase provides addition of Images to the canvas. The Output Canvas "Add Item" Action Menu will support this additional action: Add Image.

Phase IV Tasks:

(A) Basic Data Model and Configuration GUI Support

1. Define rwSettings Identifiers for Image Groups and Items
2. Data Model: class definitions for (a) Image Group and Image Item, including rwSetting list definitions.
3. Object Tree (QTreeView) and Settings support for Image Groups (one instance) and Images.
4. "Add Image" action implementation, including identification and loading of an image file.
5. Image data serialization and persistence in the Image Item.

(B) Image Graphics Item

1. Creation, maintenance, and rendering
2. Image context menu, operations: delete, configure.
3. Other (overlooked issues).

(1.5) Phase V: Create Multiple-Teacups Operation

See section 5.1 in the design document.

Phase V Tasks:

1. Basic dialog creation and layout.
2. Object list composition (Add / Remove Objects)
3. Associated Data Object identification (automatic initialization)
4. Object and Data Object name editing
5. Slot location algorithm; "Slots Found" indication
6. Create Multiple-Teacups Operation

(1.6) Phase VI: Flow Lines

This phase provides the addition of Flow Line Groups and Flow Lines to the canvas. The Output Canvas "Add Item" Action Menu will support these additional actions:

1. Add Flow Line Group
2. Add Flow Line Interval Specification ... *to Flow Line Group*
3. Add Flow Line ... *to Flow Line Group*

Phase VI Tasks:

(A) Basic Data Model and Configuration GUI Support

1. Define rwSettings Identifiers for Flow Line Groups, Interval Specifications, Flow Lines and Threshold items.
2. Data Model: class definitions for those four item types.
3. Object Tree (QTreeView) and Settings support.
4. "Add Item" Action Menu: implement the four operations enumerated above.
5. Coordination of Threshold item lifecycle (on Flow Line instances) with the number of Interval Specifications (in Flow Line Groups).
6. Flow Interval Threshold editing logic.

(B) Flow Line Graphics Items

The position of a Flow Line graphics item is based on other graphics items. The end of a Flow Line which is not *attached* to a Teacup will be attached to an invisible Flow Line Anchor Point graphics item. This "anchor point" thing is used for both "free" (unattached) Flow Line ends and Flow Line ends attached to other Flow Line ends. Flow Lines (as such) will not themselves be movable by dragging. Rather, each of a Flow Lines end dynamically "tracks" either the Teacup or the Flow Line Anchor Point to which it is attached.

1. Flow Line Anchor Point (invisible graphics item) development.
2. Flow Line graphics item, basic implementation.
3. Custom rendering, for line thickness, color and style (pattern).
4. "Attach" action, initiated by dragging and dropping a "free" Flow Line Anchor Point onto something to which the end can be attached (see above).
5. "Detach" action, initiated by a context menu operation on Teacups and Flow Line Anchor Points (having more than one Flow Line).

(C) Special Data and GUI Maintenance Issues

1. Handling of Flow Line Anchor Point positions when reducing canvas dimensions.
2. Responding to SimObj name changes.
3. Other (overlooked issues).

(2.0) Advanced Features

The following advanced features can be completed as a final phase, or can be done as part of an earlier phase.

(2.1) Inclusion of Output Canvases in Model Reports

The integration of Output Canvases into Model Reports will be virtually identical to Model Report support for Pie Charts. See occurrences of "ChartItem" in the "ModelReportItem" class.

(2.2) Animated File Generation

Animated-image file export will be supported with the MPEG, AVI, MNG or MOV file format, or something similar. None of the utility libraries we are currently using (including Qt) support any sort of movie-file generation. I believe that we will be able to make use of the **FFmpeg**, **Libav**, **MEncoder**, or **GStreamer** libraries, but this requires more investigation and some integration work.

Here are a few relevant FFmpeg-related references:

- [FFmpeg Compilation Guide](https://trac.ffmpeg.org/wiki/CompilationGuide/MSVC) / MS Visual Studio
... <https://trac.ffmpeg.org/wiki/CompilationGuide/MSVC>
- [Using FFMPEG in C++ with Qt Creator](http://infernusweb.altervista.org/wp/?p=553)
... <http://infernusweb.altervista.org/wp/?p=553>
- [Qt FFmpeg Wrapper](https://code.google.com/p/qtffmpegwrapper/) ... <https://code.google.com/p/qtffmpegwrapper/>

I had also posted a question on the Qt Project Forum (8-5-2014). There was a response, but no direct Qt-based solution (for exactly what I was looking for) was suggested.

- [Generating MPEG animations from Qt? \(from sequence of QImages or QPixmaps\)](http://qt-project.org/forums/viewthread/45897/)
... <http://qt-project.org/forums/viewthread/45897/>

(2.3) Customized rwSetting ValueType support for timestep date/times.

The rwSettings property editor infrastructure currently supports about 25 "value types" assigned to individual settings. (See Sim/rwSettings.hpp). These determine the type of editing provided within -- and from -- property editor cells. Examples include: Multiline Text, Single Line Text, Filename, Simobj name, Font, Slot value -- to name just a few.

There is currently no support for timestep date/times. This is a minor setting in the Canvas Configuration (See section 4.4). The primary way of setting that value will actually be in a timestep spinner deployed within the canvas (both preview canvas and viewer canvas) along with the timestep animation controls. So initially, we will just use a text field for that property.

But it will be nice to explicitly support a timestep date/time ValueTime (either with a program-settable timestep size, OR perhaps in combination with a new Step Size ValueType, OR just supporting the run controller's timestep size) as its own ValueType. This needs more discussion.

(3.0) Development Estimates

Estimates Summary -- *See details in the subsequent sections.*

Note that the major phases include time for "post-development review changes". This ranges from 4 hours to 12 hours for each of the six phases.

Hours (est)	Description
124	Phase I: Basic Teacups, without integrated text items -- Prototype.
48	Phase II: Basic Teacups, without integrated text items -- Production Feature
60	Phase III: Text Items in Teacups and on canvas
38	Phase IV: Canvas Images
30	Phase V: Create Multiple-Teacups Operation
102	Phase VI: Flow Lines
	Advanced Features:
10	Inclusion of Output Canvases in Model Reports
48	Animated File Generation (<i>new library research and integration</i>)*
8	Customized rwSetting ValueType support for timestep date/times.
16	Finished-Feature Documentation
484	TOTAL (Hours) -- Phase IV: Create Multiple-Teacups Operation

*The "Animated File Generation" item estimate is very approximate -- this item involves some research and assessment of 3rd party libraries.

Phase I: Basic Teacups, without integrated text items -- PROTOTYPE

Hours (est)	Hours (est)	Task	Description
56.0			(I.A) Basic Data Model and Configuration GUI Support
	16.0	#1	Define rwSettings Identifiers for each of the "Phase I" Output Canvas Config classes.
		#2	Data Model: class definitions for (a) Output Canvas Output Device (b) Output Canvas Config, containing ... (c) Teacup Group, (e) Teacup, (e) Bounding Box, and (f) Teacup Marker, (g) Teacup Legend. This includes rwSetting list definitions for most of these classes.
	8.0	#3	Management of Output Canvas output devices, including device data <i>persistence</i> .
	20.0	#4	Mostly stubbed Output Canvas Configuration Dialog.
		#5	Object Tree (QTreeView), including Qt Abstract Item Model implementation based on an Output Canvas Config instance.
		#6	"Add Item" Action Menu with the five operations enumerated above.
	8.0	#7	Deployment and integration of an rwSettingsTree in the Settings property editor panel.
	4.0	#8	Application of rwSettingsTree changes to the Output Canvas Config instance data.
28.0			(I.B) Basic Canvas / Graphics View Implementation
	8.0	#1	Basic Qt Graphics View subclasses (of RwGraphicsScene and RwGraphicsView) to implement the canvas, deployed within the Output Canvas Config Dialog as the "preview" panel.
	8.0	#2	Stubbed development of the Teacup and Teacup Legend Graphics Items, with just dummy (icon) display.
	8.0	#3	Basic canvas operations: Teacup item selection, moving (by dragging).
	4.0	#4	Reference Timestep GUI support: Deploy Date/time spinner / animation controls.
20.0			(I.C) Teacup Graphics Items
	4.0	#1	Data source slot reference support, resolution of data source slots for the various graphic entities.
	4.0	#2	Geometry computations and vertical scale composition.
	12.0	#3	Graphics rendering for each of the graphical features listed above.
12.0			(I.E) Output Canvas Viewer Dialog
	8.0	#1	Deployment of simplified version of canvas in viewer dialog.
	4.0	#2	Function: Export as Image File ...
8.0			Post-Development Review Changes
124.0			TOTAL (Hours) -- Phase I: Basic Teacups, without integrated text items -- PROTOTYPE

Phase II: Basic Teacups, without integrated text items -- PRODUCTION FEATURE

Hours (est)	Hours (est)	Task	Description
4.0			(I.B) Basic Canvas / Graphics View Implementation
	4.0	#5	Heuristic initial Teacup placement algorithm, based on relative SimObj positions within the simulation view.
4.0			(I.C) Teacup Graphics Items
	4.0	#4	Teacup context menu, operations.
8.0			(I.D) Teacup Legend Graphics Items
	4.0	#1	Geometry computations: vertical positions determined by label text rows.
	4.0	#2	Graphics rendering modifications for legend-specific geometries
8.0			(I.E) Output Canvas Viewer Dialog
	8.0	#3	Slot Data Display Functions: plot, show in SCT, open object, open slots.
16.0			(I.F) Special Data and GUI Maintenance Issues
	6.0	#1	Handling of item positions when reducing canvas dimensions.
	2.0	#2	Responding to SimObj name changes.
	8.0	#3	Other (overlooked issues).
8.0			Post-Development Review Changes
48.0			TOTAL (Hours) -- Phase I: Basic Teacups, without integrated text items -- PRODUCTION FEATURE

Phase III: Text Items in Teacups and on canvas

Hours (est)	Hours (est)	Task	Description
16.0			(III.A) Basic Data Model and Configuration GUI Support
	8.0	#1	Define rwSettings Identifiers for Text Groups and Items
		#2	Data Model: class definitions for (a) Text Group and Text Item, including rwSetting list definitions, persistence.
	8.0	#3	Object Tree (QTreeView) and Settings support for Text Groups and Text Items.
		#4	"Add Item" Action Menu: implement the six operations enumerated above.
16.0			(III.B) Canvas Text Items (in Text Groups)
	4.0	#1	Text composition (of the non-Plain text item types)
	4.0	#2	Geometry computations
	4.0	#3	Text Graphics Items / Text drawing
	4.0	#4	Text Item context menu, operations: delete, configure.
12.0			(III.C) Canvas Text Items in Teacups
	4.0	#1	Text composition enhancements (using the Teacup's context)
	4.0	#2	Geometry computations (including revisions to Teacup geometry)
	4.0	#3	Text drawing (as part of the Teacup graphics)
8.0			(III.D) Special Data and GUI Maintenance Issues
	2.0	#1	Handling of text item positions when reducing canvas dimensions.
	2.0	#2	Responding to SimObj name changes.
	4.0	#3	Other (overlooked issues).
8.0			Post-Development Review Changes
60.0			TOTAL (Hours) -- Phase III: Text Items in Teacups and on canvas

Phase IV: Canvas Images

Hours (est)	Hours (est)	Task	Description
22.0			(IV.A) Basic Data Model and Configuration GUI Support
	8.0	#1	Define rwSettings Identifiers for Image Groups and Items
		#2	Data Model: class definitions for (a) Image Group and Image Item, including rwSetting list definitions.
	8.0	#3	Object Tree (QTreeView) and Settings support for Image Groups (one instance) and Images.
		#4	"Add Image" action implementation, including identification and loading of an image file.
	6.0	#5	Image data serialization and persistence in the Image Item.
12.0			(IV.B) Image Graphics Item
	4.0	#1	Creation, maintenance, and rendering
	4.0	#2	Image context menu, operations: delete, configure.
	4.0	#3	Other (overlooked issues).
4.0			Post-Development Review Changes
38.0			TOTAL (Hours) -- Phase IV: Canvas Images

Phase V: Create Multiple-Teacups Operation

Hours (est)	Hours (est)	Task	Description
24.0			(V.A) Dialog and operation development
	4.0	#1	Basic dialog creation and layout.
	4.0	#2	Object list composition (Add / Remove Objects)
	4.0	#3	Associated Data Object identification (automatic initialization)
	4.0	#4	Object and Data Object name editing
	4.0	#5	Slot location algorithm; "Slots Found" indication
	4.0	#6	Create Multiple-Teacups Operation
6.0			Post-Development Review Changes
30.0			TOTAL (Hours) -- Phase V: Create Multiple-Teacups Operation

Phase VI: Flow Lines

Hours (est)	Hours (est)	Task	Description
30.0	(VI.A) Basic Data Model and Configuration GUI Support		
	12.0	#1	Define rwSettings Identifiers for Flow Line Groups, Interval Specifications, Flow Lines and Threshold items.
		#2	Data Model: class definitions for those four item types, persistence.
	8.0	#3	Object Tree (QTreeView) and Settings support.
		#4	"Add Item" Action Menu: implement the four operations enumerated above.
	6.0	#5	Coordination of Threshold item lifecycle (on Flow Line instances) with the number of Interval Specifications (in Flow Line Groups).
	4.0	#6	Flow Interval Threshold editing logic.
48.0	(VI.B) Flow Line Graphics Items		
	12.0	#1	Flow Line Anchor Point (invisible graphics item) development.
	6.0	#2	Flow Line graphics item, basic implementation.
	6.0	#3	Custom rendering, for line thickness, color and style (pattern).
	12.0	#4	"Attach" action, initiated by dragging and dropping a "free" Flow Line Anchor Point onto something to which the end can be attached (see above).
	12.0	#5	"Detach" action, initiated by a context menu operation on Teacups and Flow Line Anchor Points (having more than one Flow Line).
12.0	(VI.C) Special Data and GUI Maintenance Issues		
	2.0	#1	Handling of Flow Line Anchor Point positions when reducing canvas dimensions.
	2.0	#2	Responding to SimObj name changes.
	8.0	#3	Other (overlooked issues).
12.0	Post-Development Review Changes		
102.0	TOTAL (Hours) -- Phase VI: Flow Lines		

Advanced Features (can individually be done as part of prior phases).

Hours (est)	Hours (est)	Task	Description
10.0	#1 -- Inclusion of Output Canvases in Model Reports		
48.0	#2 -- Animated File Generation		
	24.0	#1	Research and acquiring, building and experimenting with 3rd party libraries.
	16.0	#2	Integration with RiverWare Build and Build System / Development System Setup
	8.0	#3	Application Feature Development
8.0	#3 -- Customized rwSetting ValueType support for timestep date/times.		
66.0	TOTAL (Hours) -- Advanced Features (can individually be done as part of prior phases).		

--- (end) ---