

## RiverWare 6.7 Model Report Enhancements: Image Items and Rich Text

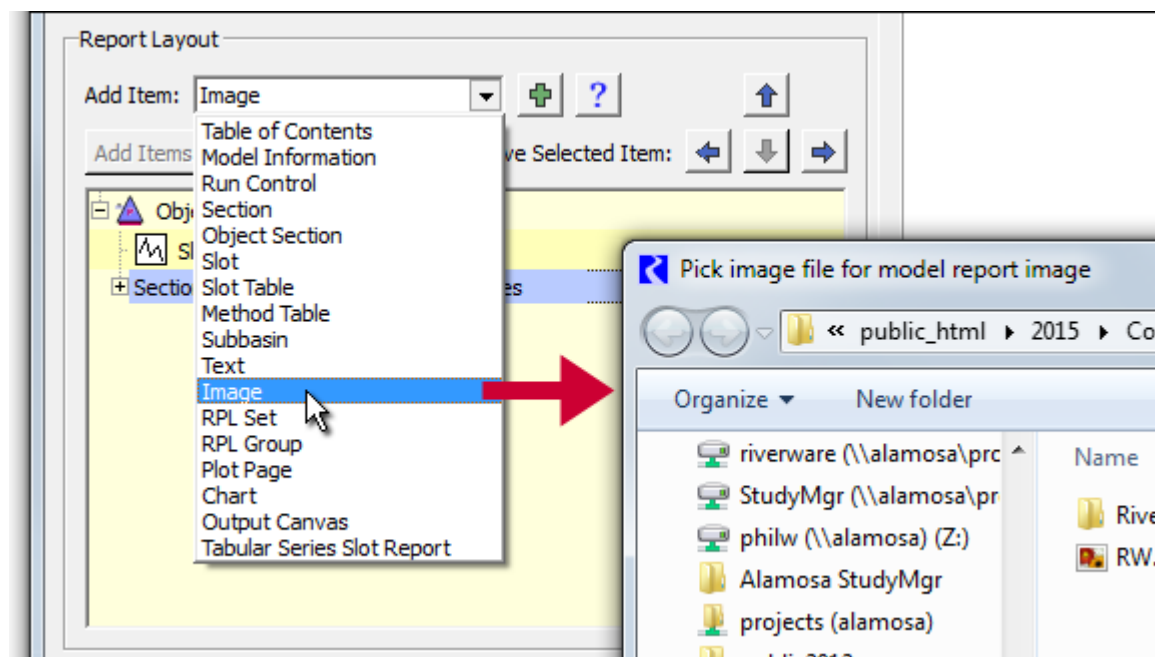
Phil Weinstein, David Neumann, Edie Zagana, CADSWES, 5-1-2015

The following two enhancements have been developed for RiverWare Model Reports:

1. Image Items. Individual images can be imported from an image file (JPEG, PNG, GIF).
2. "Rich Text" support for Text Items, with basic support from copying rich content from external applications, e.g. Word.

### (1) Image Items

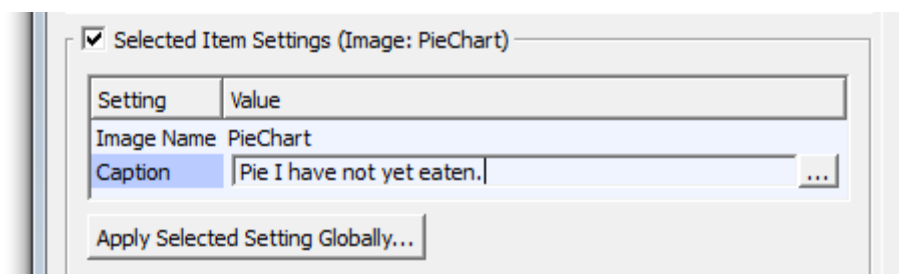
A new "Image" menu item appears in the Report Layout panel's "Add Item" combo box and in the context menu within the report configuration item tree. Selecting "Image" brings up a file selector to pick an image file ...



After picking an image file, a new Image item is created with the image data from that file. (The image data is maintained within the model report configuration).

Image items support two settings:

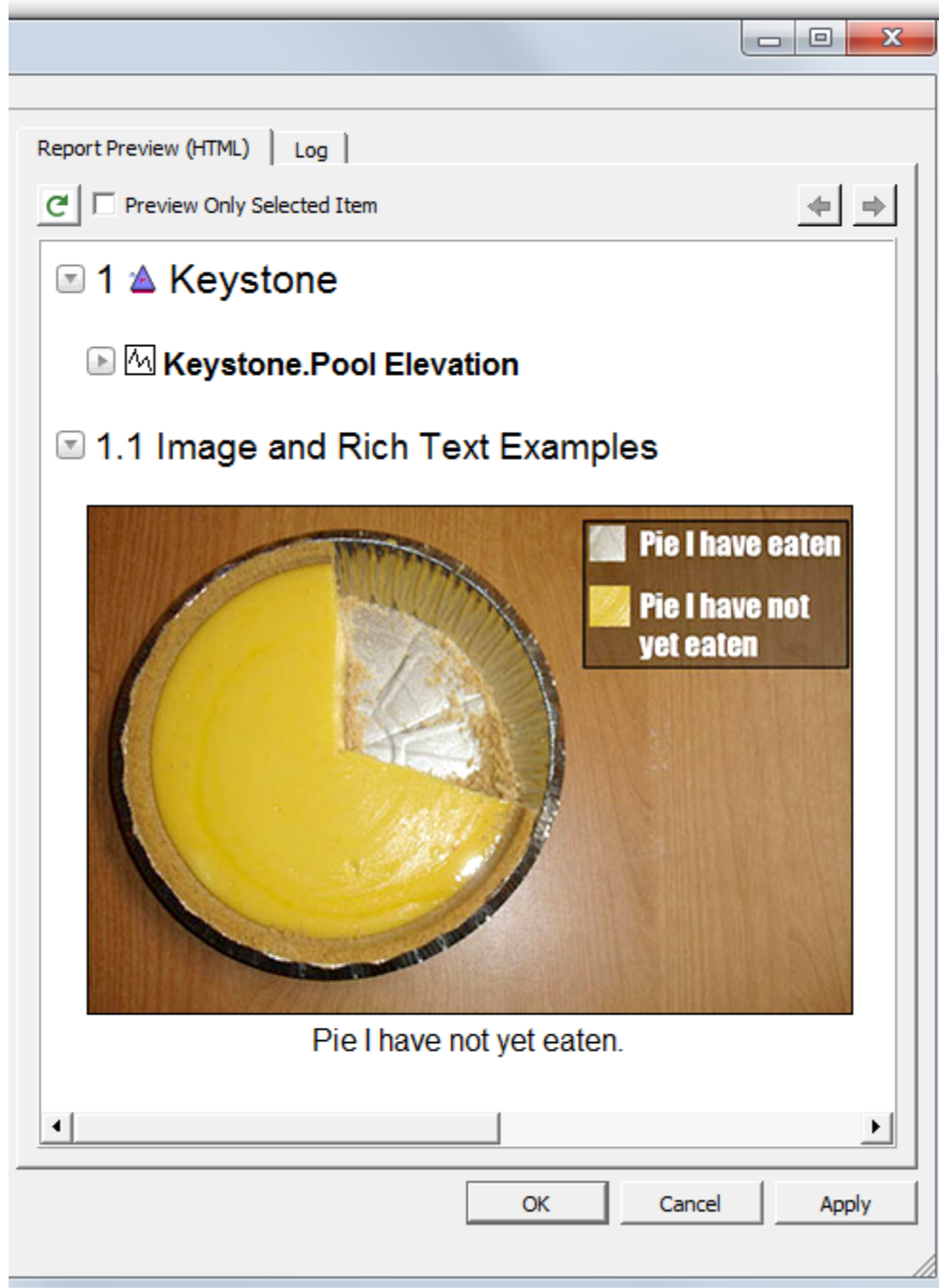
- Image Name
- Caption



The Image Name is initialized from the picked image file. Its only use is identifying the image item within the Report Layout item tree.

Optional caption text can be specified with each image item. If specified, it is shown under the image in the model report. For sufficiently wide images (at least 300 pixels wide), the caption is horizontally centered underneath the image, and wrapped to the width of the image. Captions for narrower images are not constrained in this way; they appear under the image, left aligned, and flowing to the full display width of the generated model report page.

The full model report content -- or optionally just a single selected report item -- can be previewed by clicking the green-circle-arrow preview refresh button at the top of the Report Preview panel.



It is *not currently possible* to switch out a different image for a previously created image item. If a different image is desired, a new image item must be created. (The user may wish to retain the old image item during the creation of the new image item so that the old image caption text can be copied and pasted to the new image item).

When the Model Report HTML document is generated, new image files are created for model report image items in a similar way as images for slot plots, but using the binary image data loaded from the original source image file.

Note that, unlike slot plots which are dynamically constructed as needed, *since image item image data is actually stored in the model file* (in the model report output device configuration record), including many large images in a model report configuration could potentially significantly increase the size of the model file and could result in slower model file loading. If this becomes a problem, e.g. with a production model which needs to be regularly run, the user could choose to export the model report output device configuration, delete that output device from the model, and then reload the model report output device configuration (and not re-save the model) when the model report is to be generated.

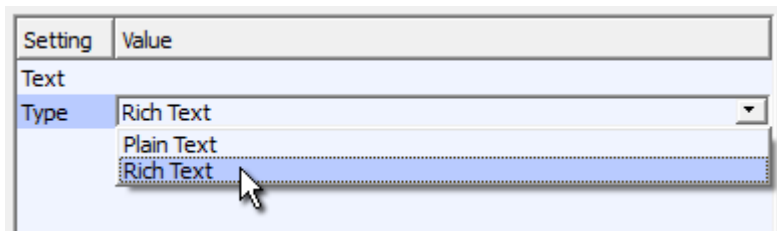
Image data size statistics are not currently displayed within the RiverWare user interface. But those statistics can be seen in uncompressed copies of the model file or model report output device export files. Each image item's basic statistics are available on a single text line; search or "grep" for "BinaryData". Here is an example of such a line for the example shown above:

```
<BinaryData appl="Image" name="PieChart" width="400" height="265" sizeRaw="186818" sizeB64="252985">
```

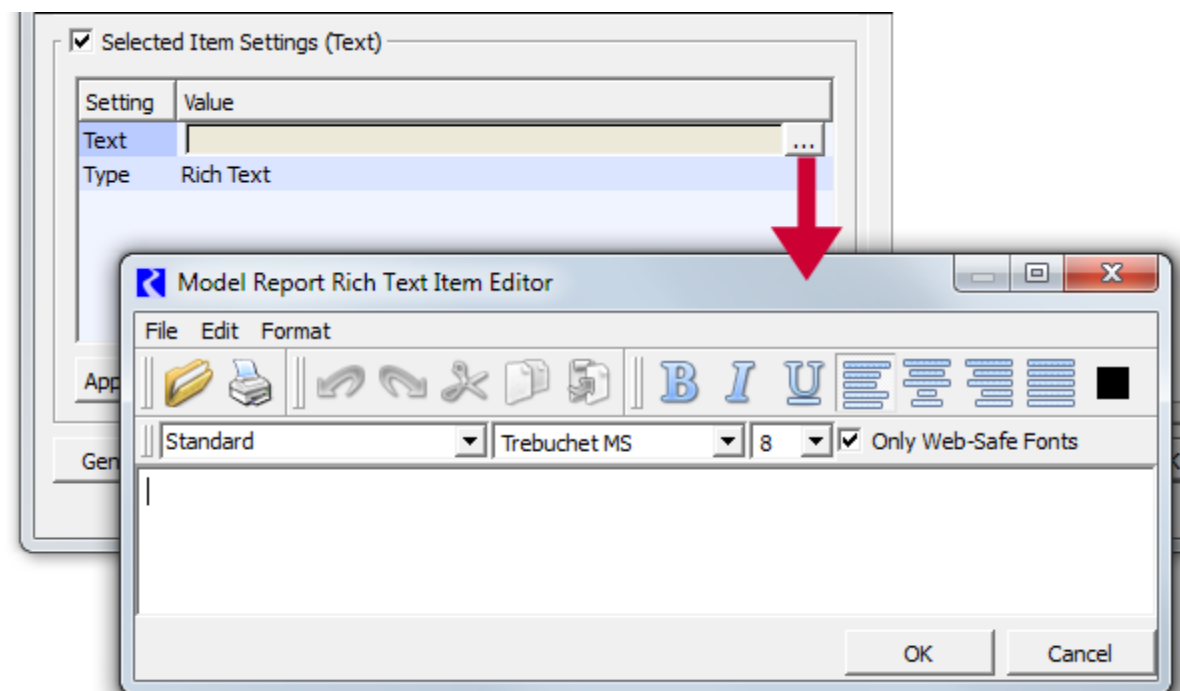
- "width" and "height" are in pixels.
- "sizeRaw" shows the number of bytes of actual binary image data, as loaded from the original model file.
- "sizeB64" shows the number of bytes of the image data, as encoded within the model file or output device export file. This is approximately 33% larger than the binary data. This is encoded in a standard way using ASCII text characters representing base-64 values (i.e. 6 bits of actual binary data in each 8-bit character; nominally 75% efficient).

## (2) "Rich Text" support for Text Items

RiverWare Model Report Text Items now support a "Rich Text" mode for basic HTML-based formatting. Newly created Model Report Text Items are in the "Rich Text" mode by default. This is controlled with an "enum" combobox "Type" setting: Plain Text or Rich Text:



When in Rich Text mode, a new, more advanced editor is shown for editing the text item content. This is initiated by clicking the ellipsis button after double-clicking in the Text setting's value field ...



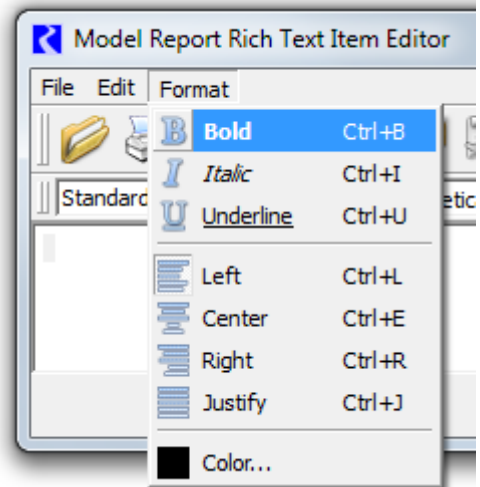
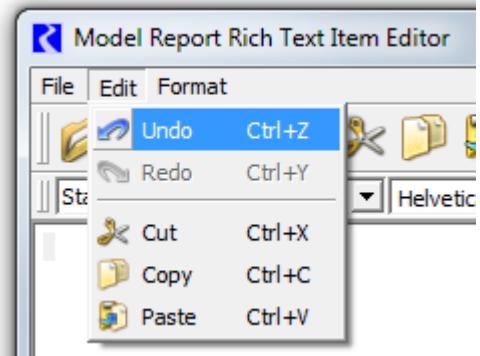
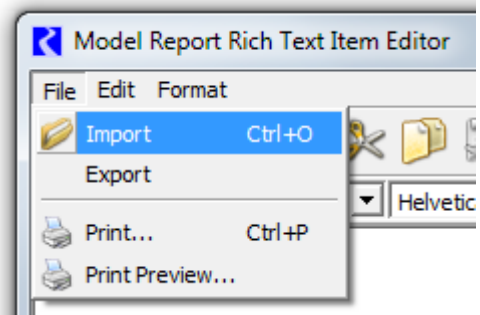
In Rich Text mode, content can be edited only in a new editor dialog -- the line editor in the settings panel showing the first part of a plain text version of the content is read-only (not editable).

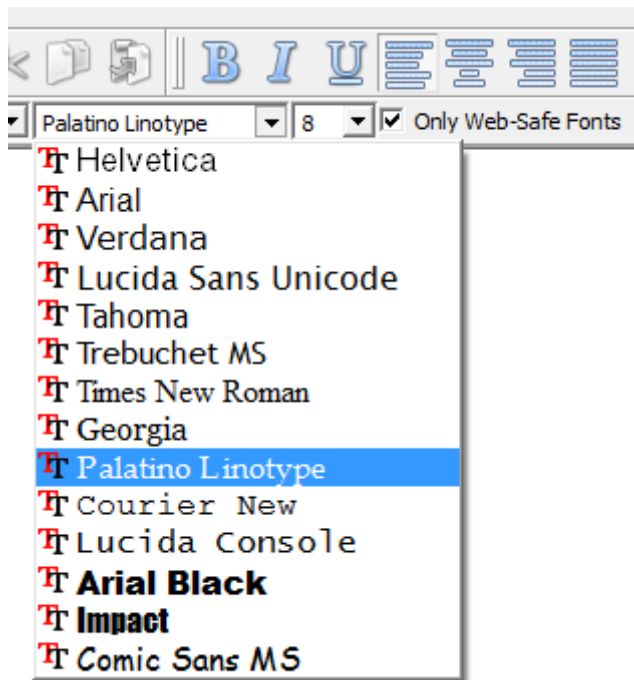
The Rich Text editor supports many common formatting operations; these are generally applied to the selected portion of the text content. Copying and pasting formatted text content from an external application (e.g. Word) -- via the system clipboard -- is basically supported. Also, it is possible to **"Import"** an existing HTML or text document; doing so replaces the editor's prior content.

The editor supports "Undo" and "Redo" operations, so it is very reasonable to experiment with the available formatting operations. Note however that the rendering by this editor doesn't precisely match the appearance of content in the generated HTML document. Formatting should be checked in the Model Report configuration dialog's preview panel after applying changes to rich text Text items.

Note: The Rich Text editor is not a full-blown HTML editor and will not work well for complex formatting. As more advanced editing tools become available (e.g. after porting RiverWare from Qt4 to Qt5), we might choose to replace this editor with something better.

Any font on the user's system can potentially be applied to rich text content. (A typical Windows system supports over 300 different font faces). However, since the generated RiverWare Model Report is an HTML document, and since web browsers generally rely on the fonts available on the viewer's system, we recommend that users make use of the "Only Web-Safe Fonts" option which limits the font selection to fourteen (14) of the most common fonts:





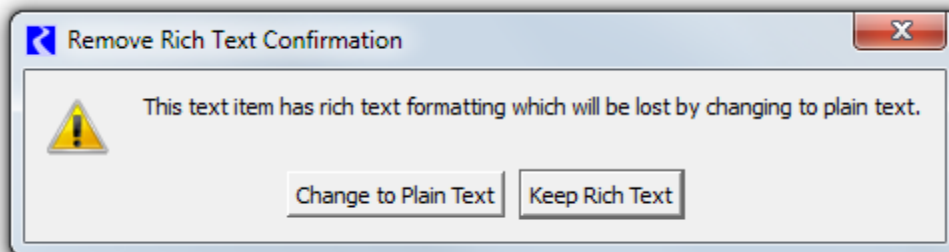
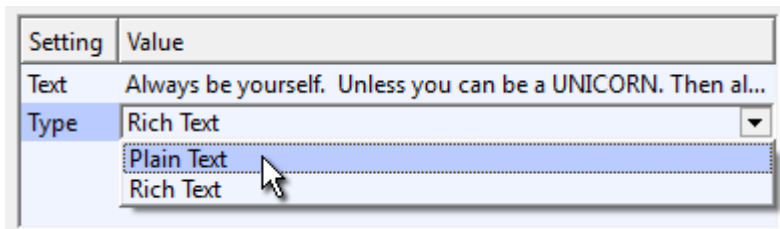
Note: The current editor implementation doesn't correctly handle "fallback" fonts, i.e. a list of alternative font faces to be used in case the first choice is not available on the user's system. Not being able to provide those increases the importance of using common ("web safe") fonts in rich text Text items. This editor limitation could potentially also cause problems with content imported from an HTML document, as the editor actually *corrupts* CSS font-family style specifications having multiple fonts; the result is that none of the specified fonts work, and the web browser uses just its default font.

- Technical: The low-level Qt4 QTextEdit widget implementation places single quotation marks around the full list of font face names in a CSS font-family style specification. Such a thing can be done for a single font face name, but not for a comma separated list of font face names. That is not valid CSS syntax. For example, the following CSS style is *invalid*:

```
<span style="font-family:'Verdana,Algerian,sans-serif'; font-size:10pt;">
```



Once rich text has been entered, switching back to "Plain Text" mode causes the loss of any formatting and text attributes which had been applied. For this reason, the following confirmation dialog is presented to the user when selecting "Plain Text" ...



--- (end) ---