Rhapsody Element Creator

Andy Lapping

Technical Fellow

Sodius Willert

A person wearing a suit and tie

Description automatically generated

# Updates June 2024

If you have used the Element Creator before then please review the changelog at the end of this document – major updates have been made:

# Overview

The Rhapsody Element Creator is a customizable helper to provide a quick and easy way to add elements to a model by providing the user with a dashboard.

## How it Works

The helper starts by reading a csv file that it uses to populate its ‘dashboard’. The currently selected element in Rhapsody is then checked and the dashboard greys out unavailable buttons (based on owners and aggregates – see below).

If a button is pressed, then an element of that type is created and highlighted in the browser (and if it is a diagram the diagram is opened). The helper continuously monitors the current selection in Rhapsody and so after adding an element the available buttons may change based on the new selection. Automatically switching to a newly created element is an option in the Options menu however if you manually select something else in the browser this is still tracked by the helper (so you can keep the window open and add elements anywhere).

### Owners and Aggregates

If you have your own custom profile and you have defined the owners or aggregates properties, then these are checked by the helper to decide if an option should be available. If no such properties have been defined then the helper enables the button (this means that it will work for non-profiled elements such as basic UML). If the button is pressed and the element is illegal (such as adding a Use Case to a Statechart) then the error is trapped and a message presented in Rhapsody.

### Options:

The Options menu allows you to toggle Always on Top and also whether the Helper automatically navigates to an element upon creation (thereby changing the currently selected element)

Graphical user interface, text, application, chat or text message

Description automatically generated

These options are stored in a preferences file in your user directory (ElementCreator.ini)

## Example

Launching the Element Creator on a Project with a custom profile (in this case a Star Wars profile !) presents the following – all buttons are greyed out except for Equipment Configuration – the only element allowed at project level by this profile:

A picture containing graphical user interface

Description automatically generated

Clicking Equipment Configuration creates one and then switches to it – now further buttons are available:

Graphical user interface, application

Description automatically generated

Clicking AT-AT creates one and the buttons change again:

Graphical user interface

Description automatically generated

# Customization

There are two options for customization:

* Modify the provided HEP, PRP and CSV files
* Use your own profile.

To use your own profile:

1. Copy the Apps folder so it is next to your profile
2. (optional) Copy the Doc folder so it is next to your profile (this contains property help)
3. Copy the HEP file or copy the entries from it into your own HEP file.
4. An example is provided in the CustomProfileExample folder.

The supplied profile has a **PRP**, a **HEP** and a sample csv file.

The supplied example custom profile provides a PRP, a HEP and two sample csv files.

Screenshots below are from the custom profile

## HEP File

A screenshot of a computer program

Description automatically generated

Note that multiple CSV files may be used by specifying paths and file names, this allows multiple grids to be used and invoked from the browser rather than having one (potentially very large) grid.

## File Names

The name of the file is taken from the menu entry in the HEP file (only the menu **name** – the path to the menu name is discarded). For example, this menu entry would look for a file named ***Launch Grid 2.csv***:

name3=Element Creator\Launch Grid 2

## File Paths

The helper looks in the folder specified by the property:

General.ElementCreator.BasePathForCustomCSVFiles

The path may:

* Be a full path (e.g. C:/MyCSVFiles)
* Contain Rhapsody variables such as $OMROOT
* Be relative to a profile / package\*

\*To specify the package/profile the csv file path is relative to, set the following property:

General.ElementCreator.BasePathRelativeTo

## PRP File

The **PRP** file has the following properties:

### ElementCreatorName

This property allows you to name the helper (the name appears in the header bar of the dashboard – the same name will be used regardless of which csv file is in use).

## LockOnSelectedElementAtLaunch

If the *LockOnSelectedElementAtLaunch* property is set to true on the project, then new elements will always be added to the element that was selected when the helper was launched. If it is set to false, then new elements will always be added to the **current** selection (the current selection may change if the Navigate to Created Element option is selected in the UI)

## EnableLogging

Setting this property to true on the project enables a more detailed output – useful if you are specifying custom properties and want to check they are being read correctly.

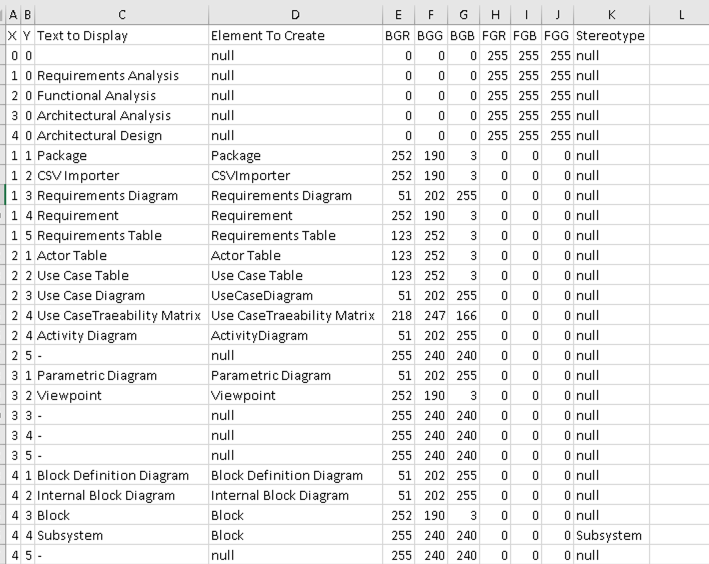
## BasePathRelativeTo

The property should contain a full path to the package / profile in the format Package::Package::Package

For example, to make the path relative to a profile then simply use the profile name (since it is at the root of the model). To make the path relative to a ‘Files” package in that profile then use *MyProfile::Files*

## Elements File

The CSV file(s) should be formatted as shown below to customize the content of the dashboard(s). The CSV file is read upon launch of the helper and so you can modify it freely without restarting Rhapsody or reloading the model – simply relaunch the helper after changes.



**X**: Row in Dashboard

**Y**: Column in Dashboard

**Text to Display**: The name that appears in the dashboard

**Element To Create**: The model element to create when the button is pushed (which can be base types or new terms). Note that Rhapsody sometimes uses different names in the API than in the display – for example to add a *Use Case Diagram* then the Element To Create field should read *UseCaseDiagram*

**BGR, BGG, BGB**: The RGB colour code for the background colour in the cell

**FGR, FGB, FGG**: The RGB colour code for the Foreground colour in the cell.

**Stereotype**: The stereotype to apply on the newly created element. All of the cells must have an entry - use the keyword *null* to specify no stereotype.

**Empty Cells**: You may either leave a cell empty (have no entry for it at all in the matrix) or you may add some text and set the Element to Create to null (this is a keyword that prevents the helper from creating a button at all). However, the table must be fully formed – that is if you have a row in the CSV file then each cell in the row must contain data.

# Conclusion

This is a flexible helper that I’ve used on many profiles including UAF:

A picture containing screenshot, cabinet, sign

Description automatically generated

# Changelog

## June 2024

In order to make the creator more flexible, the following changes have been made:

* Path to csv files may now be relative or contain any Rhapsody variable.
* Property PathToElementsFile is no longer used. File names are now determined by menu and properties as described above.
* Added property help to base project
* Added example of custom profile
* Added profile builder project