What is Package Diagram? How to Draw Package Diagram?

Written Date : July 29, 2014

At the beginning of the project, you only have a limited number of diagrams and everything is simple and beautiful. However, when time flies, more and more diagrams have been created and they start to become unmanageable. As a result, your project becomes hard to navigate and diagrams become difficult to locate when you want to review or make changes.

How can we fix it up? We can make use of the Package Diagram to organize your diagrams into different packages. This helps you in categorizing your diagrams according to their natures, making them easier to be navigated and located. The Package Diagram also serves as a catalog for you to jump to the diagram that you want to look at. In this tutorial, we will show you how this can be done.

Create Packages for your diagrams

First, we need to have our packages ready. To create packages:

1. To create a Package Diagram, select **Diagram > New** from the toolbar.
2. In the **New Diagram** window, select **Package Diagram** and click **Next**.

3. Enter *Racing Game Packages* as diagram name and click **OK** to confirm.

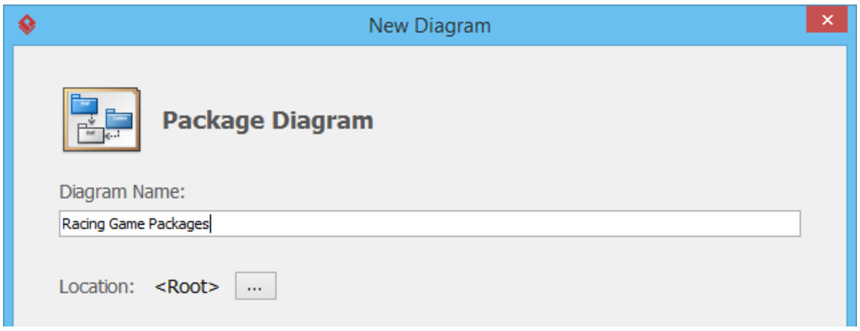
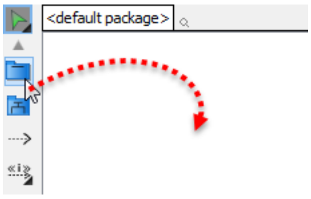
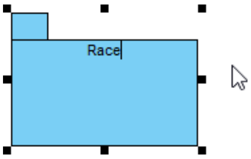


Diagram Name: Racing Game Packages

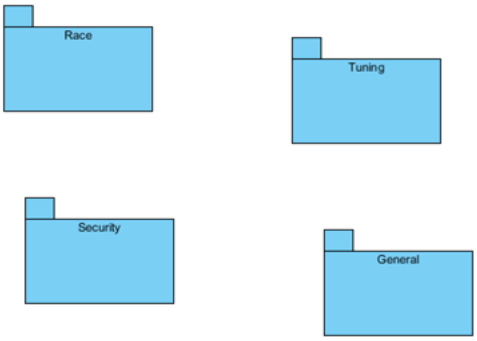
4. Click the **Package** button in diagram tool bar, then click on the blank area of the diagram to create the package.



5. Name the package as *Race*.



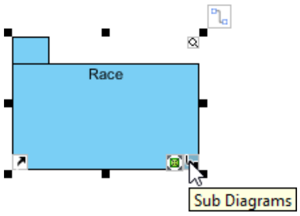
6. Repeat the steps to create other packages, namely, *Tuning*, *Security* and *General* respectively.



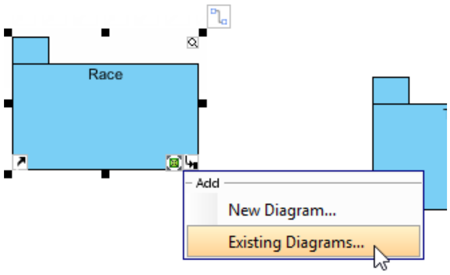
Organize diagrams with Packages

Once the packages are ready, we can then start to organize our diagrams.

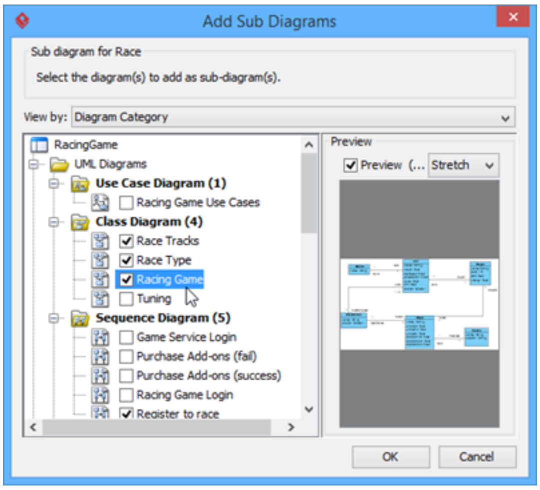
1. Click on the *Race* package to bring up the sub-diagram resource icon.



2. Click on the resource icon, then select **Existing Diagrams...** from the popup menu.



3. Select the following diagrams in the **Add Sub Diagrams** dialog, then Press **OK** to commit.
    1. [Class Diagram] Race Tracks
    2. [Class Diagram] Race Type
    3. [Class Diagram] Racing Game
    4. [Sequence Diagram] Register to race
    5. [Communication Diagram] Register to race - Communications
    6. [State Machine Diagram] Race Series State
    7. [Activity Diagram] Register to race
    8. [Entity Relationship Diagram] Player Race
    9. [Entity Relationship Diagram] Race Series
    10. [Entity Relationship Diagram] Racing Game



Add Sub Diagrams

Sub diagram for Race

Select the diagram(s) to add as sub-diagram(s).

View by: Diagram Category

RacingGame
UML Diagrams
Use Case Diagram (1)
□ Racing Game Use Cases
Class Diagram (4)
☑ Race Tracks
☑ Race Type
☑ Racing Game
□ Tuning
Sequence Diagram (5)
□ Game Service Login
□ Purchase Add-ons (fail)
□ Purchase Add-ons (success)
□ Racing Game Login
☑ Register to race

Preview

☑ Preview (... Stretch

4. Repeat the steps for the following packages and diagrams

Package	Diagram
Tuning	[Class Diagram] Tuning [State Machine Diagram] Car State [Entity Relationship Diagram] Cars
Security	[Sequence Diagram] Game Service Login [Sequence Diagram] Purchase Add-ons (fail) [Sequence Diagram] Purchase Add-ons (success) [Sequence Diagram] Racing Game Login [Business Process Diagram] Purchase Add-ons
General	[Use Case Diagram] Racing Game Use Cases

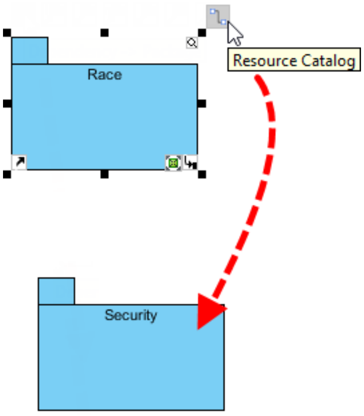
	[Component Diagram] Game Components [Deployment Diagram] Game Deployment [Requirement Diagram] General Requirements [Matrix Diagram] Classes Matrix [Matrix Diagram] Entity Matrix [Matrix Diagram] Process Matrix [Analysis Diagram] Player Analysis Diagram

Define relationships between packages

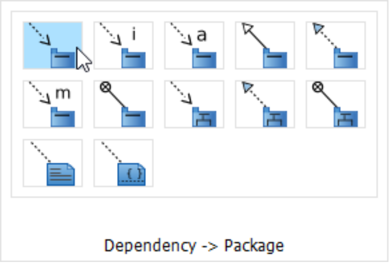
Now, since our diagrams are being structured with packages, we can define the relationships among them. To create relationships between packages:

1. Click on the *Race* package to show up its resource icon.

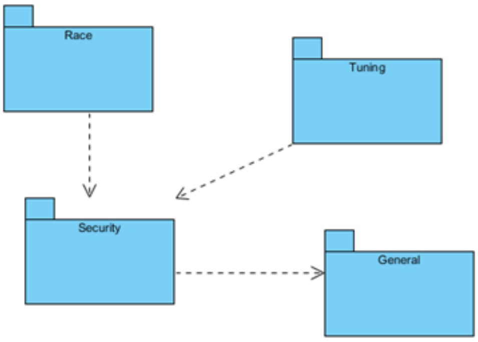
2. Drag out the **Resource Catalog** button and drop it at the *Security* package.



3. Select **Dependency -> Package** in Resource Catalog.



4. Repeat the steps to create dependency from *Tuning* to *Security* and also from *Security* to *General*.

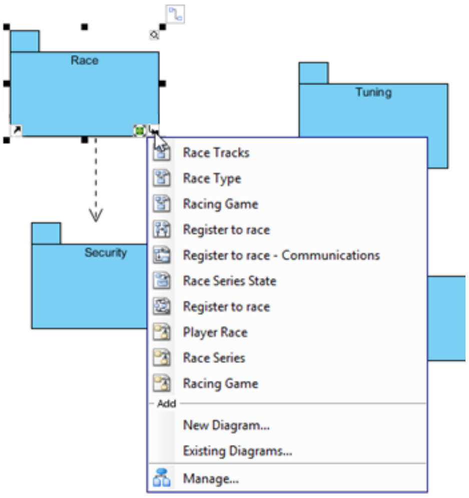


Navigate between packages and contained diagrams

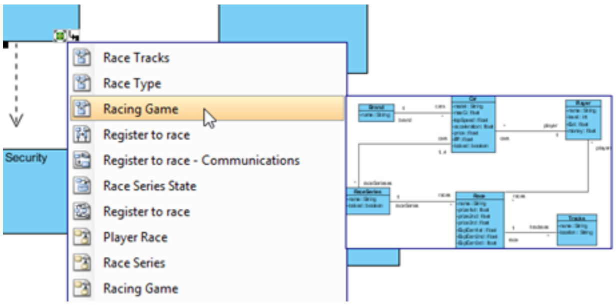
The Package Diagram can act as a starting point for you to navigate within your project. To navigate from one diagram to another:

1. Click on the package to bring up the resource icons, say the *Race*.

2. Click on the sub-diagram resource icons. This will bring up a list of sub-diagrams under the Race package.

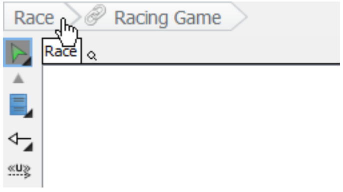


3. Locate the diagram from the popup menu, say the *Racing Game* class diagram and then click on it.



4. You are now being brought to the *Racing Game* class diagram.

5. You can go back to the package by clicking the link at the top left corner of the diagram.



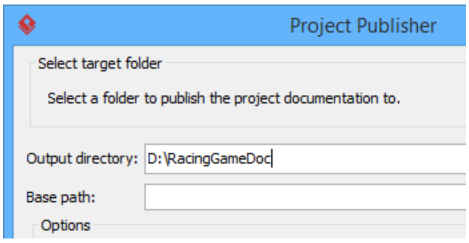
Publish a project to HTML with Package Diagram as the navigation map

This kind of handy navigation is not limited to within the application. You can also have it inside the published HTML website. To publish your project into a HTML website:

1. Save your project by selecting **Project > Save** from the toolbar.

2. Select **Tools > Doc > Publish Project...** from the toolbar.

3. Enter the output location for the publisher in the **Output directory** field.


Project Publisher

Select target folder

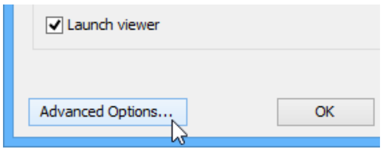
Select a folder to publish the project documentation to.

Output directory: D:\RacingGameDoc

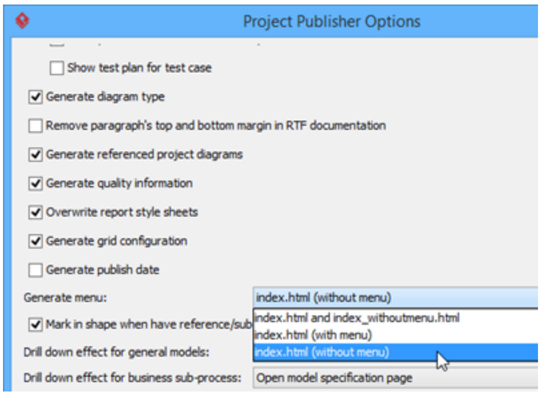
Base path:

Options

4. Press the **Advanced Options...** button to bring up the advanced settings.


☑ Launch viewer

5. Select **index.html (without menu)** in the **Generate menu** combo box.

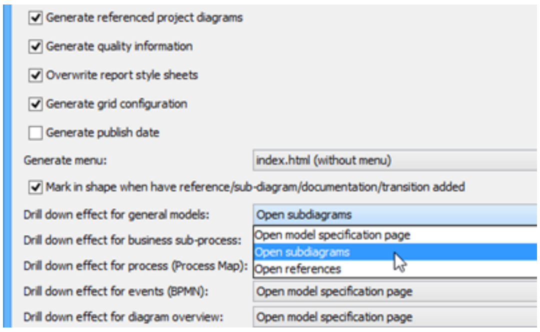


Project Publisher Options

□ Show test plan for test case
☑ Generate diagram type
□ Remove paragraph's top and bottom margin in RTF documentation
☑ Generate referenced project diagrams
☑ Generate quality information
☑ Overwrite report style sheets
☑ Generate grid configuration
□ Generate publish date

Generate menu: index.html (without menu)
index.html and index_withoutmenu.html
index.html (with menu)
index.html (without menu)
☑ Mark in shape when have reference/sub
Drill down effect for general models:
Drill down effect for business sub-process: Open model specification page

6. Select **Open subdiagrams** in **Drill down effect for general models**.

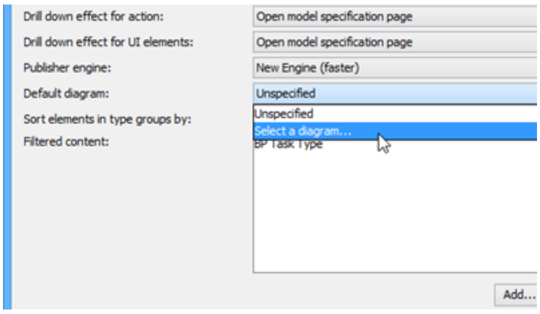


☑ Generate referenced project diagrams
☑ Generate quality information
☑ Overwrite report style sheets
☑ Generate grid configuration
□ Generate publish date

Generate menu: index.html (without menu)
☑ Mark in shape when have reference/sub-diagram/documentation/transition added
Drill down effect for general models: Open subdiagrams

Drill down effect for business sub-process:
Drill down effect for process (Process Map):
Drill down effect for events (BPMN): Open model specification page
Drill down effect for diagram overview: Open model specification page

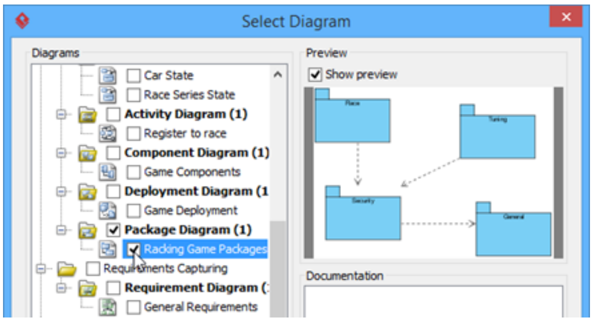
7. Choose **Select a diagram** in the **Default diagram** combo box.



Drill down effect for action: Open model specification page
Drill down effect for UI elements: Open model specification page
Publisher engine: New Engine (faster)
Default diagram: Unspecified

Sort elements in type groups by:
Filtered content:

8. Pick the *Racing Game Packages* and press **OK**.



Select Diagram

Diagrams

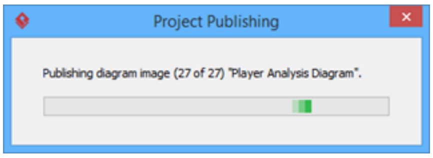
- □ Car State
- □ Race Series State
- □ **Activity Diagram (1)**
- □ Register to race
- □ **Component Diagram (1)**
- □ Game Components
- □ **Deployment Diagram (1**
- □ Game Deployment
- ☑ **Package Diagram (1)**
- ☑ Racing Game Packages
- □ Requirements Capturing
- □ **Requirement Diagram (**
- □ General Requirements

Preview

☑ Show preview

Documentation

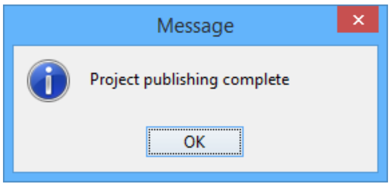
9. Press **OK** to close the **Advanced Options** dialog and press **OK** again in the **Project Publisher** dialog to start publishing.



Project Publishing

Publishing diagram image (27 of 27) "Player Analysis Diagram".

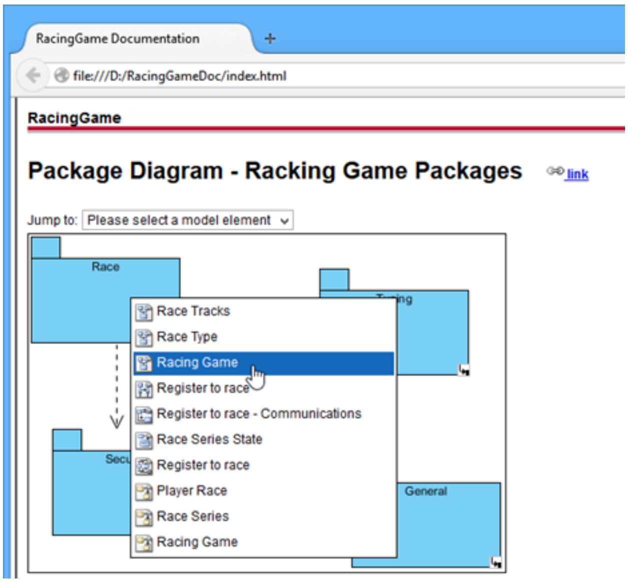
10. Press **OK** when publishing is completed. This will launch the browser and with the Racing Game Packages diagram showing.



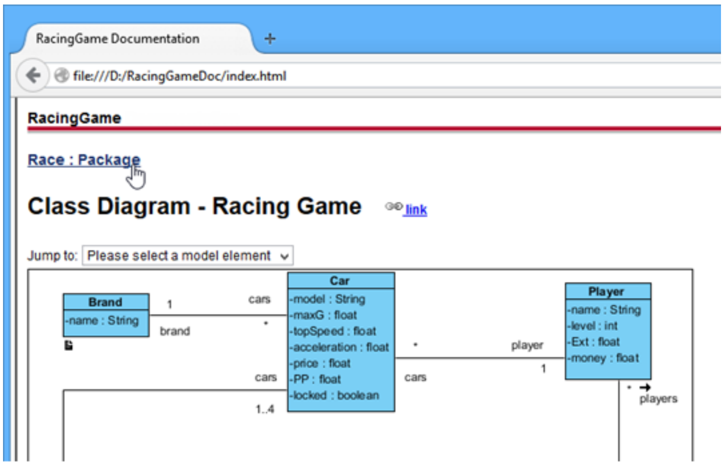
Message

Project publishing complete

11. You can click on the package to bring up the popup menu to jump to the sub-diagram, say the *Racing Game* class diagram.



12. And in the sub-diagram you can always go back to the parent package by clicking the link at the top of the page.



Download Sample Project

You can download the sample project to walk through this tutorial.

Watch this Tutorial on YouTube

Here is the video version of this tutorial.

Related Links

- What ist Package Diagram?

- Publish project as web content
- Sub-diagram vs Reference diagram
- Full set of UML tools and UML diagrams

Visual Paradigm home page
(https://www.visual-paradigm.com/)

Visual Paradigm tutorials
(https://www.visual-paradigm.com/tutorials/)