

SYD366 Notes

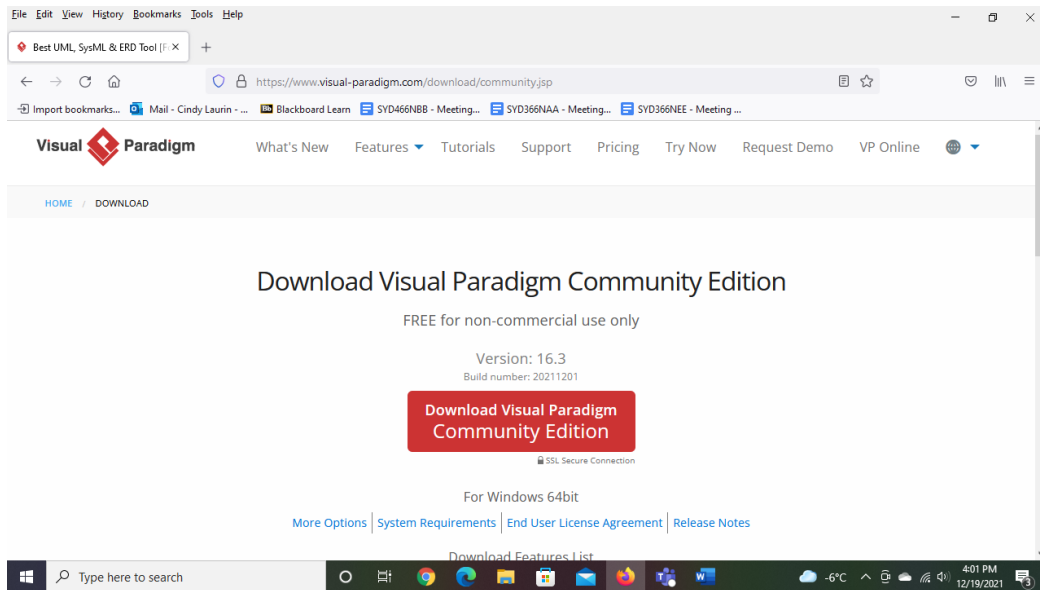
Contents

Installing Visual Paradigm	2
Creating a Class Diagram	2
Creating a Class	5
Drawing Associations	7
Reference Attributes.....	9
Creating a Controller.....	9
Creating a Sequence Diagram (including Actor, Controllers and Classes).....	13
How to change an object to a Multi-Object in a Sequence Diagram.....	16
Defining Messages in Sequence Diagrams.....	16
Defining Replies in Sequence Diagrams.....	17
Defining Create Messages in Sequence Diagrams.....	18
How to Retrieve a List of Objects from the Entity Manager	19
How to Create a New Object (Add) and Save it to the Entity Manager	19
How to Change an Object and Save it to the Entity Manager	20
How to Remove an Object and Delete from the Entity Manager.....	22
Loops in Sequence Diagrams	23
Exporting Diagrams and Copying to Word.....	23

Installing Visual Paradigm

Using the Search Engine of your choice, search for Visual Paradigm Community Edition.

<https://www.visual-paradigm.com/download/community.jsp>



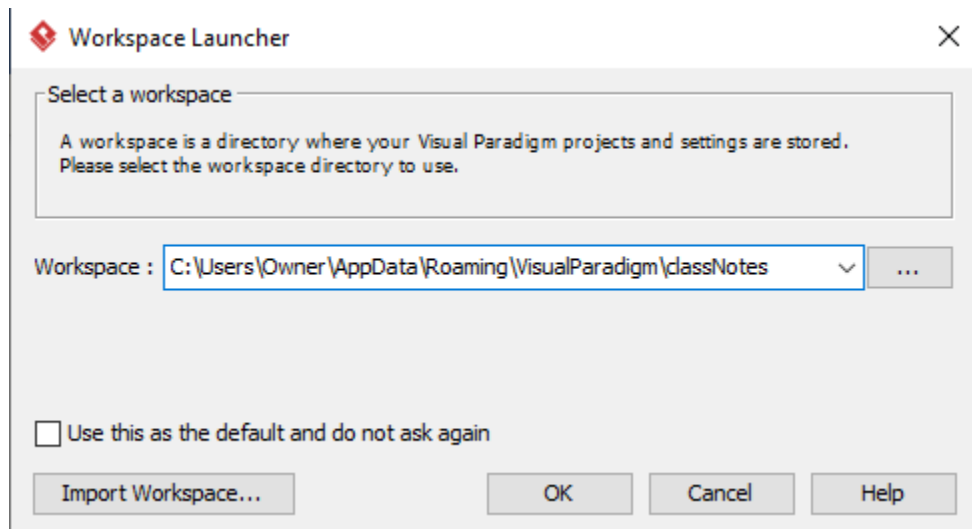
Click on the download button and install.

Creating a Class Diagram

Woodbine Entertainment Group operates Woodbine Horse Racing in Toronto. One of the features of their system is Maintain Horses which allows the Racing Secretary to keep track of the horses that are available to race at WEG. The real feature has screens of information for each horse, but for our example, we will assume that every horse has a horse code and horse name.

To define our Horse Class, we must first start by creating a Class Diagram.

Open Visual Paradigm.

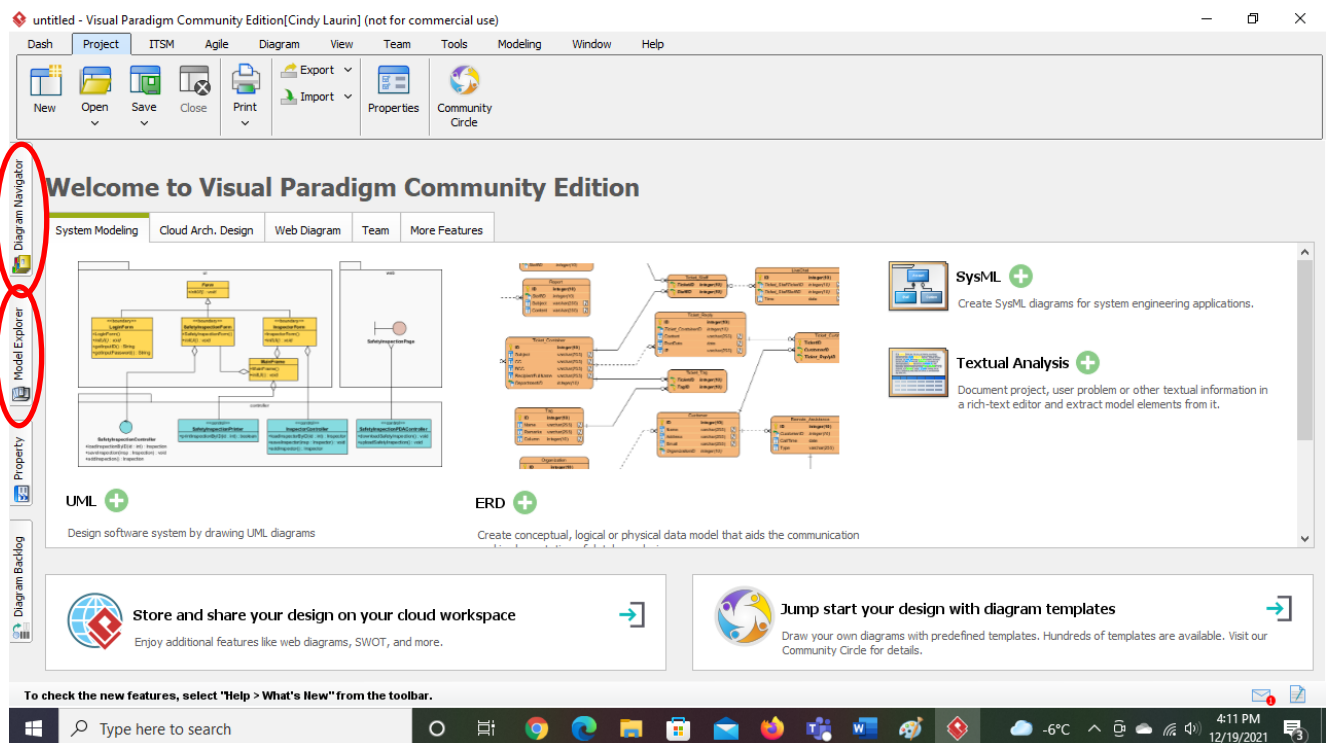


A workspace is a folder on your computer that stores all of your settings. Ours is name classNotes. Click Ok.

Visual Paradigm will initialize your environment and prompt you to take a look at the new features. We suggest that you do, otherwise, you'll have to do this every time that you open your workspace!

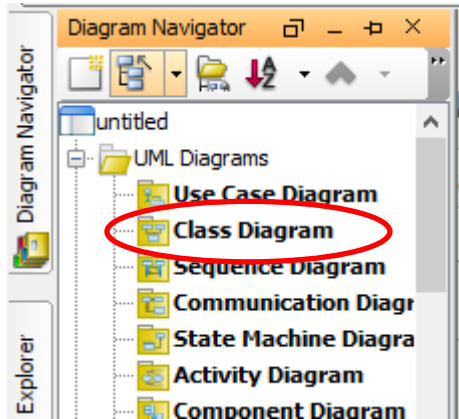
Click Yes to continue in the next startup.

You should end up with a screen that looks like this.

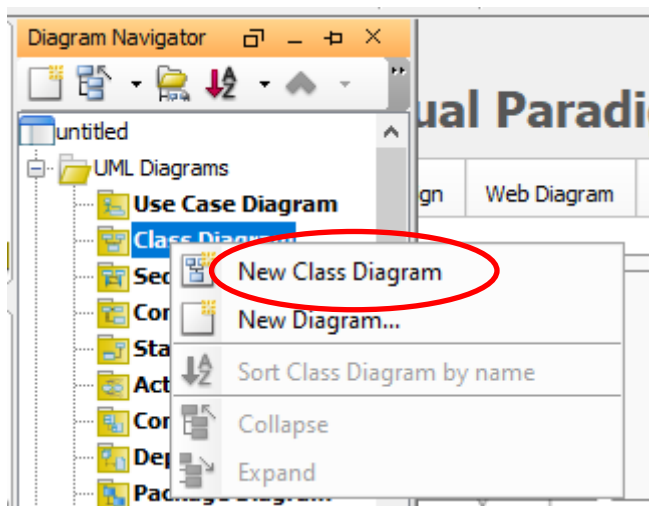


Make sure that you see Diagram Navigator and Model Explorer down the left hand side of your screen. We'll use these panes often.

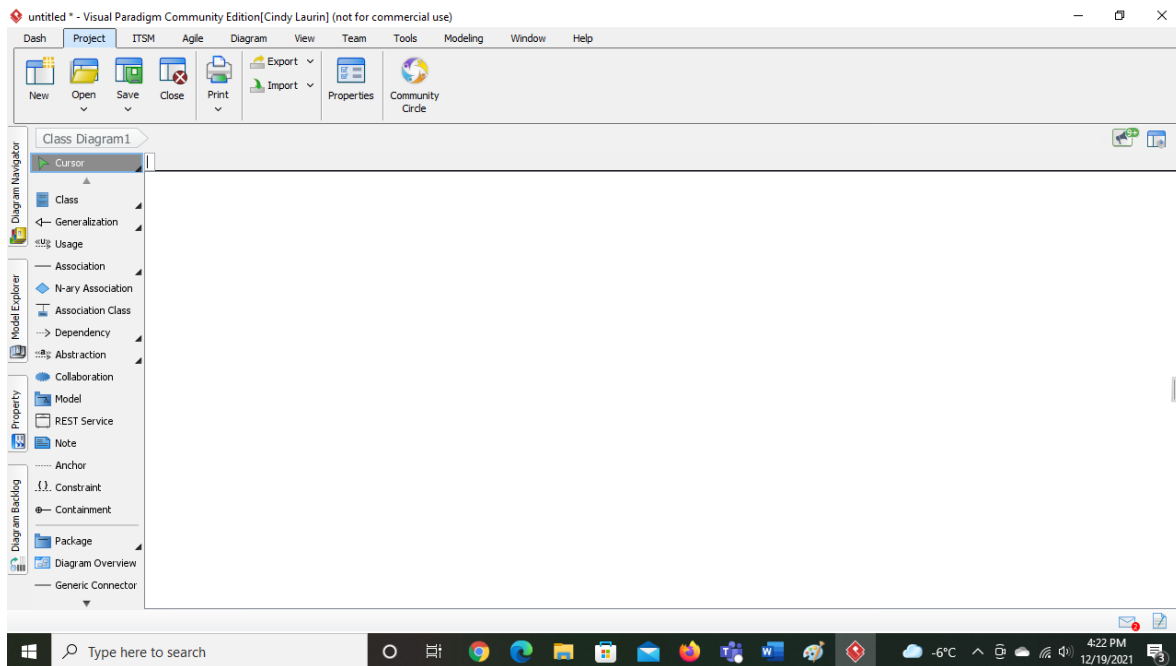
To create a Class Diagram, click on Diagram Navigator



Then Right Click on Class Diagram

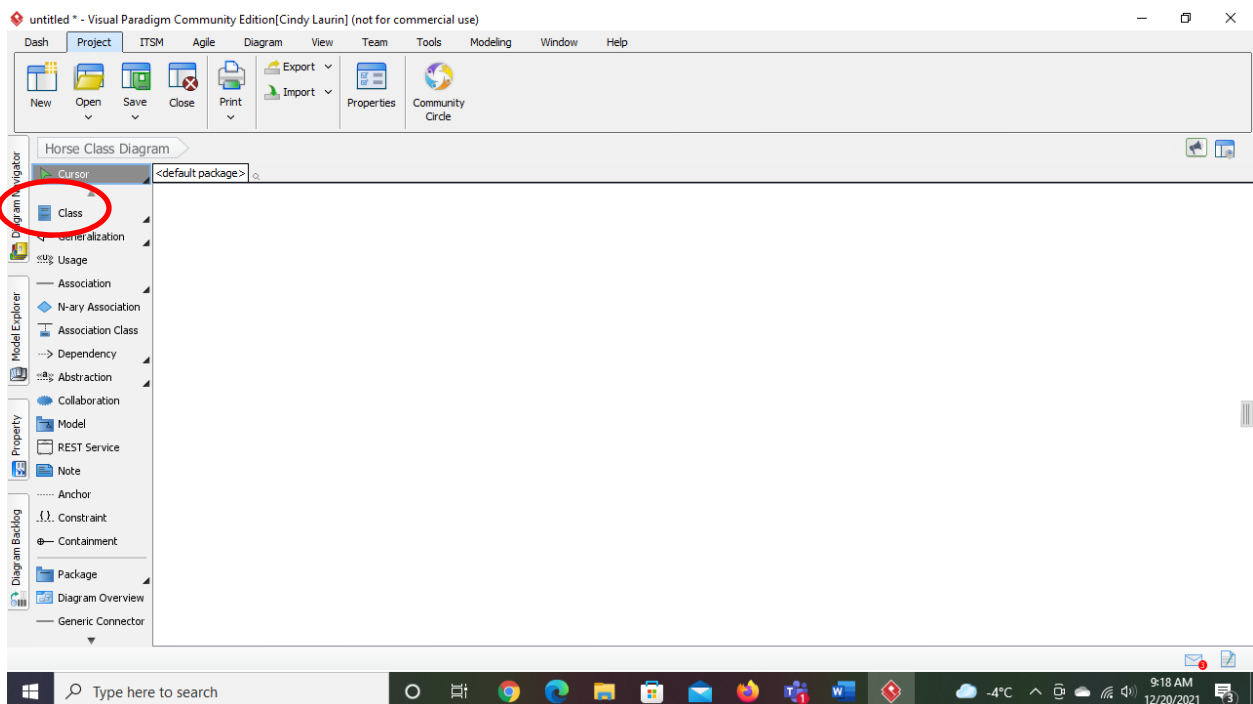


Then click on New Class Diagram. You should see the following screen. Be sure to name your class Diagram!

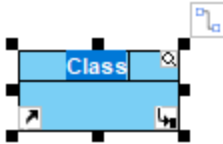


Creating a Class

Let's create our Horse Class. Our class has 2 attributes, horse code which is the unique identifier and horse name. Please note the class icon on the tool bar!

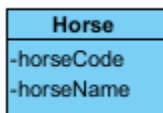
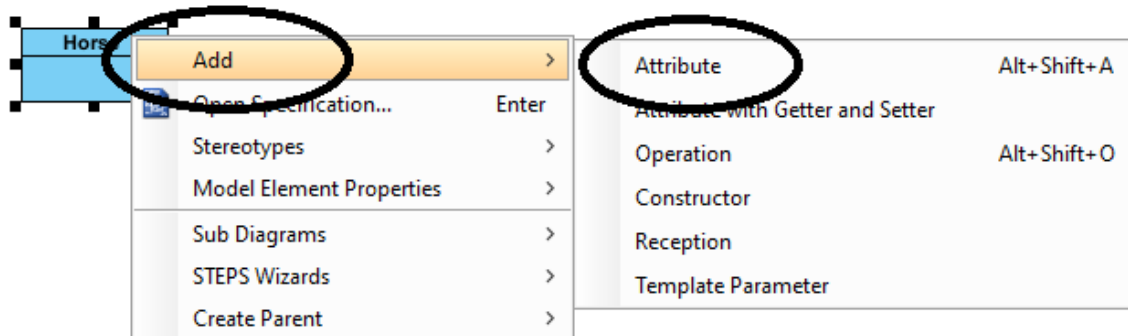


Click on the class icon and click inside your diagram. You'll see a new class in your class diagram.

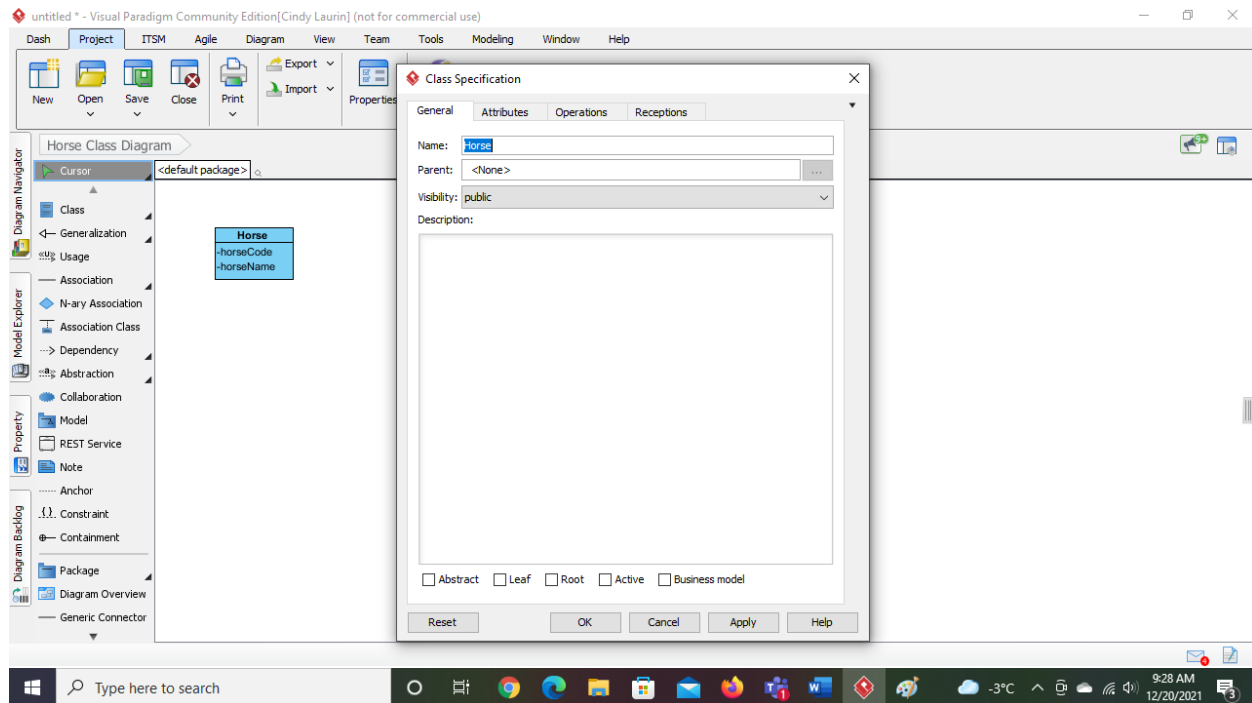


Change the class name. Remember to use CapitalizedHumpBack notation. Use singular noun form for class names.

To add attributes to the class, right click on the class, click on Add and click on Attribute.



If you run into problems, right click on the class and click on Open Specifications. You can solve most problems from there.

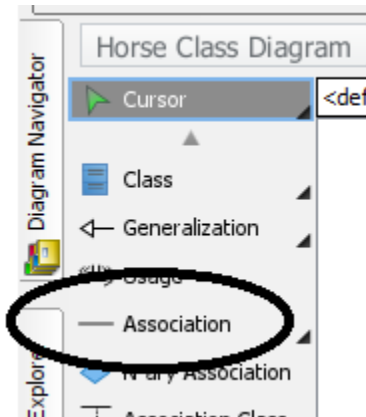


Drawing Associations

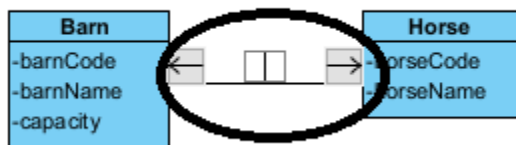
Horses are assigned to barns while they reside on the property. Each barn holds at most 50 horses. Our Barn class has as attributes, barn code (unique identifier), barn name and capacity. Our class diagram now looks like this:



To note that horses reside in barns, we use an association. Please find the association icon on your tool bar.



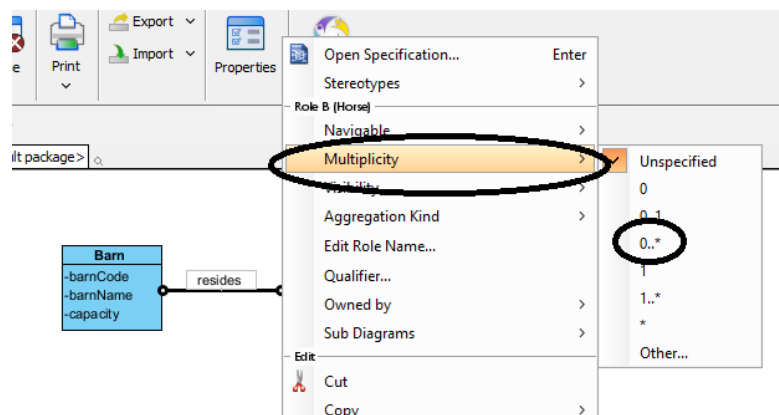
Click on Association. Click on Barn. Keeping your mouse click down, draw a line to Horse. This connects the two horses.

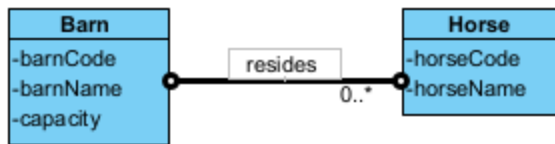


Name the association. Never use 'has'!



To add multiplicities, click on the association. In our situation, a horse must reside in a barn, but a barn could be empty or hold up to the capacity limit. Barn would have a multiplicity of 0 or more with regards to horse and horse would have a multiplicity of 1 with regards to barn. For the barn to horse multiplicity, right click on the right-hand small circle. Click on multiplicity and then 0..*.





Click on the association again.

To add the horse to barn multiplicity, right click on the left-hand small circle. Choose multiplicity and then 1.



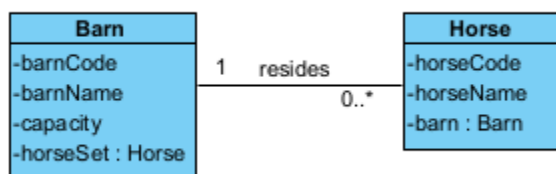
Reference Attributes

A Barn would have a set of Horses residing in it and a Horse resides at a Barn, so we must add the reference attributes to further define the relationships.

To add the reference attributes horseSet to barn, right click on Barn, choose Add and then attribute. Add the attribute, horseSet: Horse to the class.

The attribute horseSet defines that we have a collection of horses and the : Horse defines that horseSet is a collection of Horse objects.

We repeat the same exercise for the horse class and add barn: Barn to Horse. This defines that Horse has an attribute of barn and barn is a Barn object.

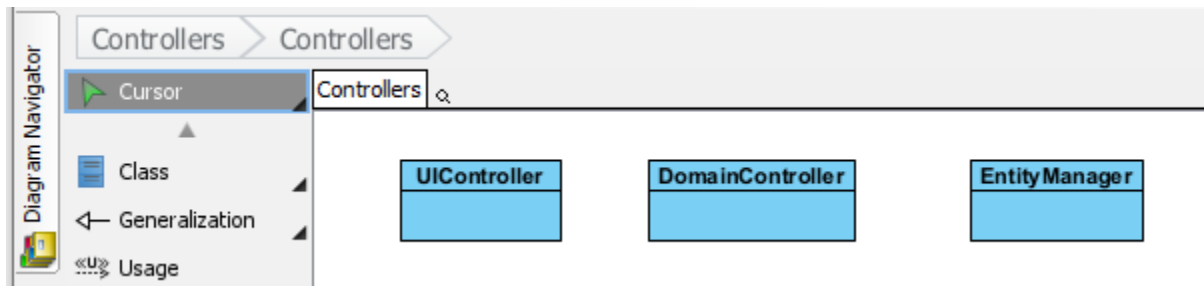


Creating a Controller

Controllers are classes that manage functionality instead of data. In SYD366, we use 3 controllers. UIController is the class that manages the user interface. If we are building a web application, this class would contain our html, css and javascript code. DomainController is the class that manages the business domain logic. In a web application, this class would contain the server side code (C++, Perl, C#, PHP...). The Entity Manager is the class that manages the data source. If your using an SQL datasource, then the Entity Manager would be contain SQL statements.

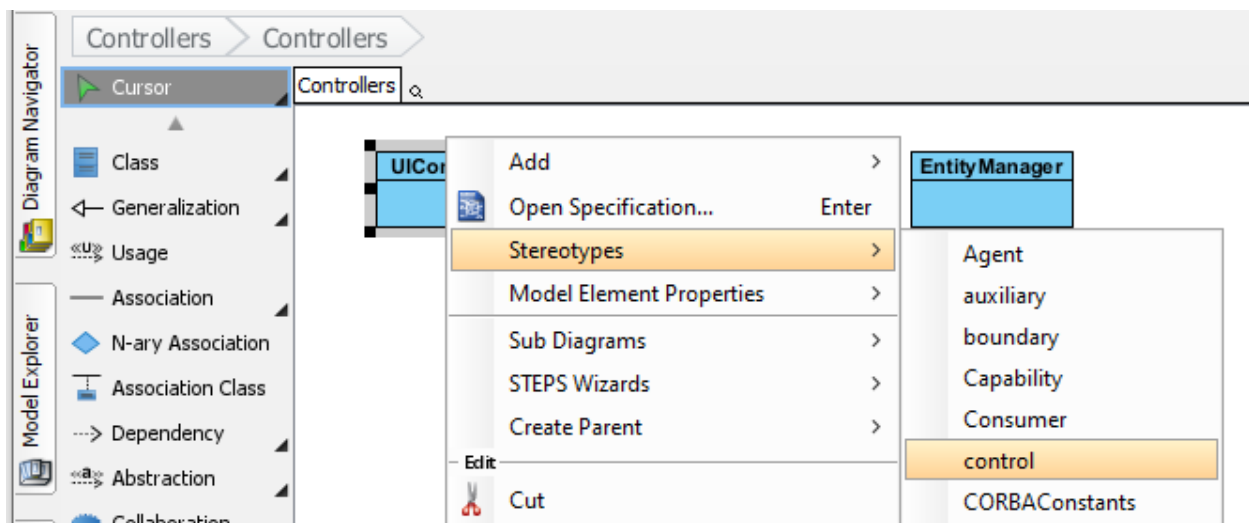
Every model that you create must have a Controllers Class Diagram.

To create this class diagram, start with the same instructions as creating a class diagram and add the UIController, DomainController, EntityManager classes.

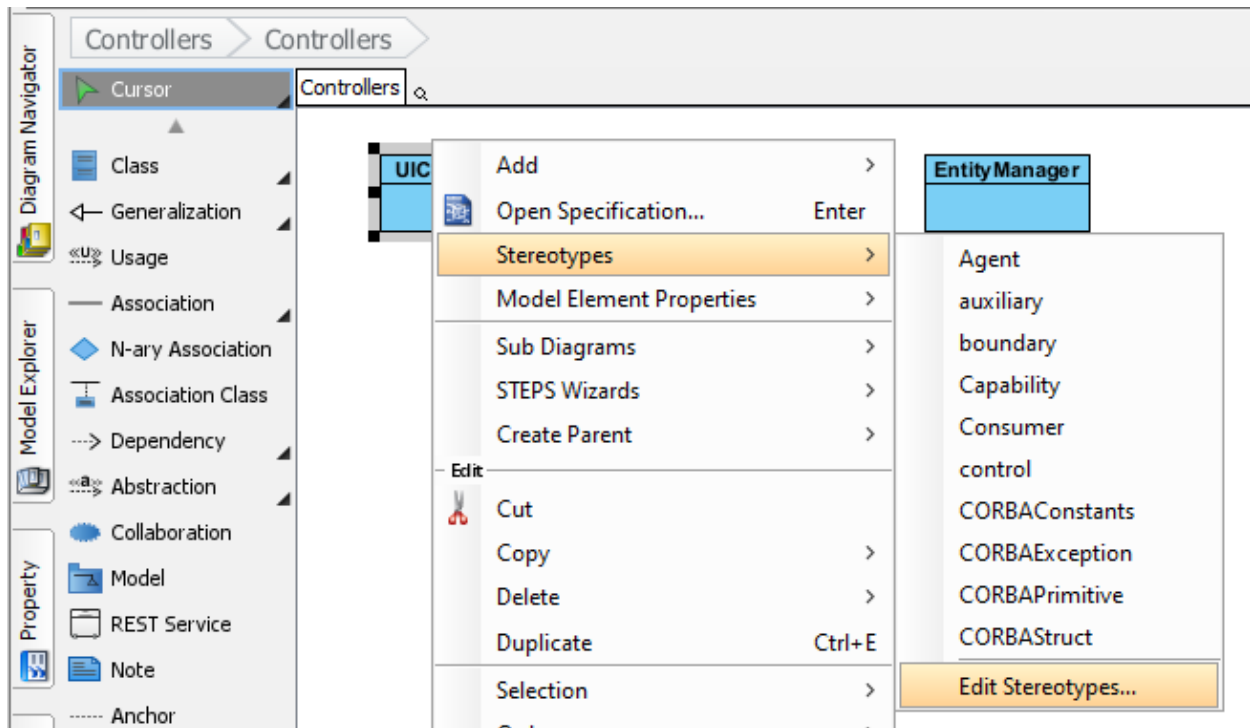


Next, change the Stereotype to control.

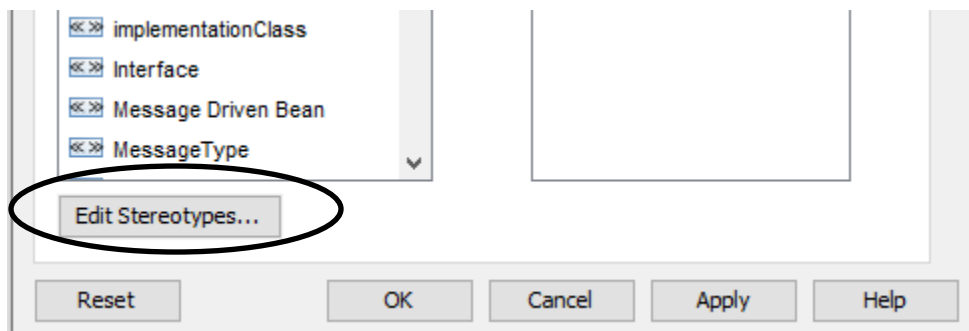
Right click on the UIController, click on Stereotypes and then control.



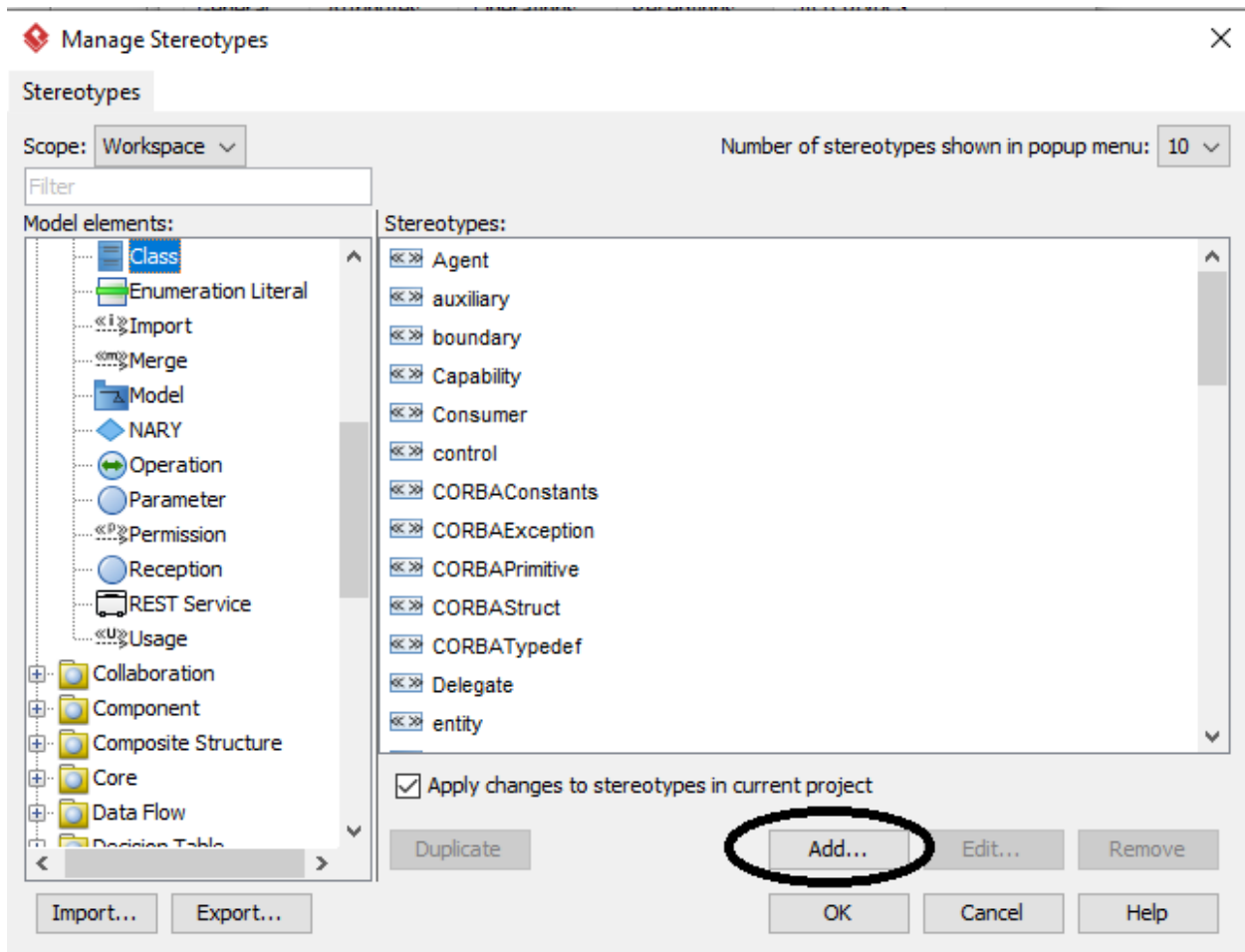
If control doesn't show up on your list then click Stereotypes and click Edit Stereotypes.



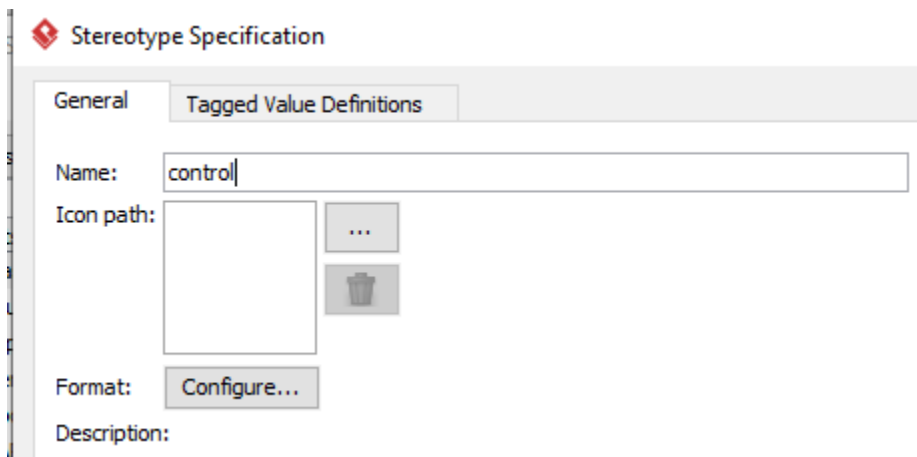
At the bottom of the Class specification window, click on Edit Stereotypes.



Click on Add at the bottom of the Manage StereoTypes Window

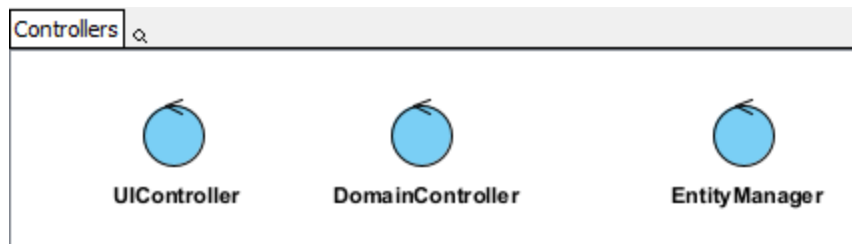


Type in control as the stereotype name



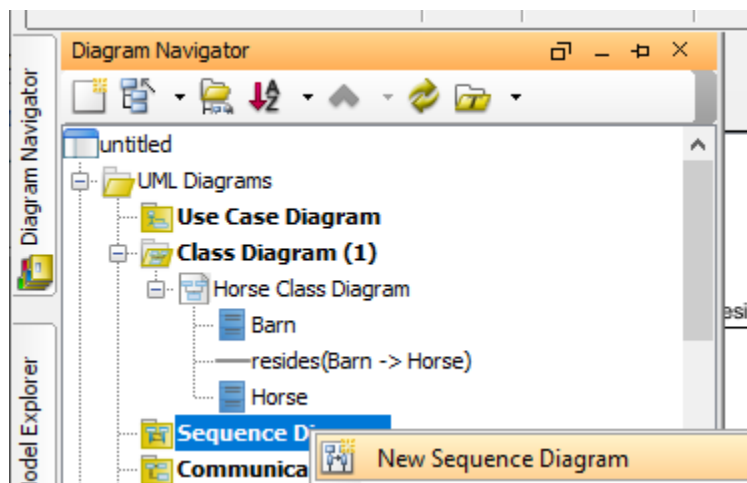
Click OK and then OK and you should see control on the stereotype list. Select control.

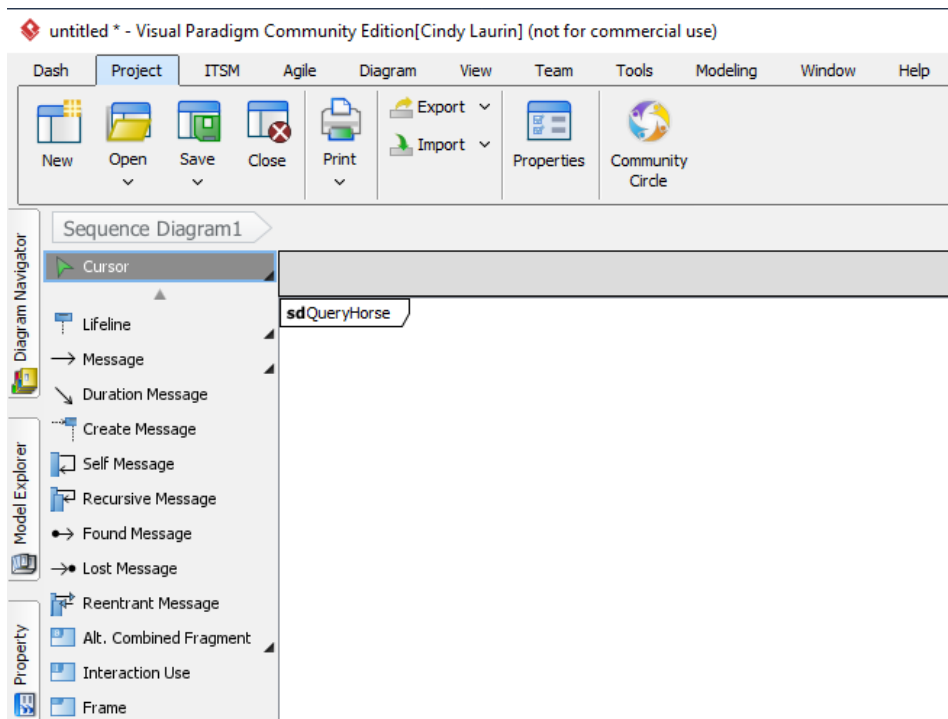
Add the stereotype control to the DomainController and Entity Manager classes. You should end up with this class diagram.



Creating a Sequence Diagram (including Actor, Controllers and Classes)

To create a sequence diagram, return to the Diagram Navigator, right click on Sequence Diagram and then click on New Sequence Diagram.

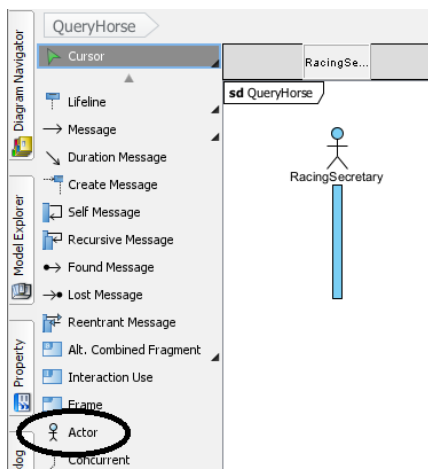




Be sure to name your diagram!

Next step, we add our lifelines. The first lifeline belongs to our Actor. The name of the Actor is the business title assigned to the person doing the work. In our example, that person is called the Racing Secretary.

Find the Actor icon on the tool bar. Click on the Actor icon and then click on the diagram. Change the name from Actor to the actual actor's title.



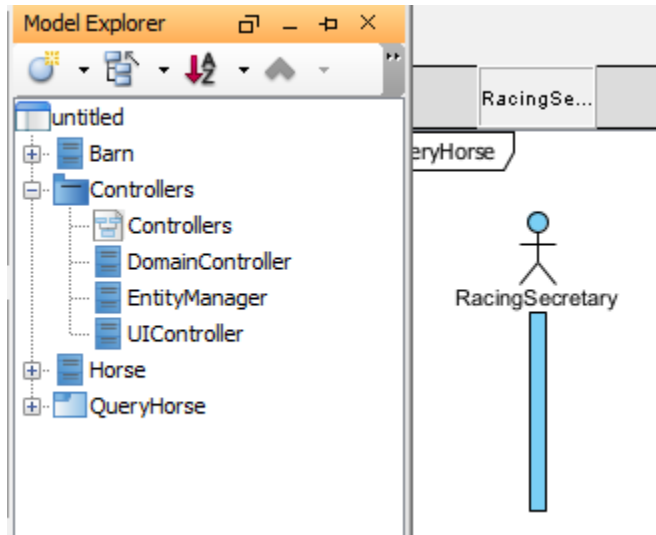
Next step, add the controllers in the following order, using the Model Explorer:

UIViewController

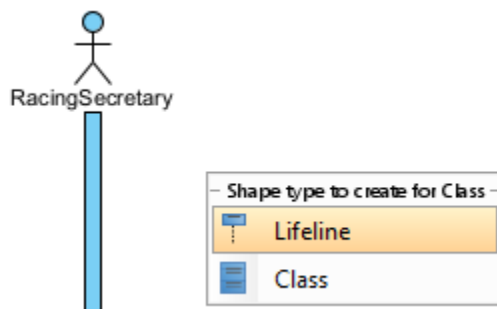
DomainController

EntityManager

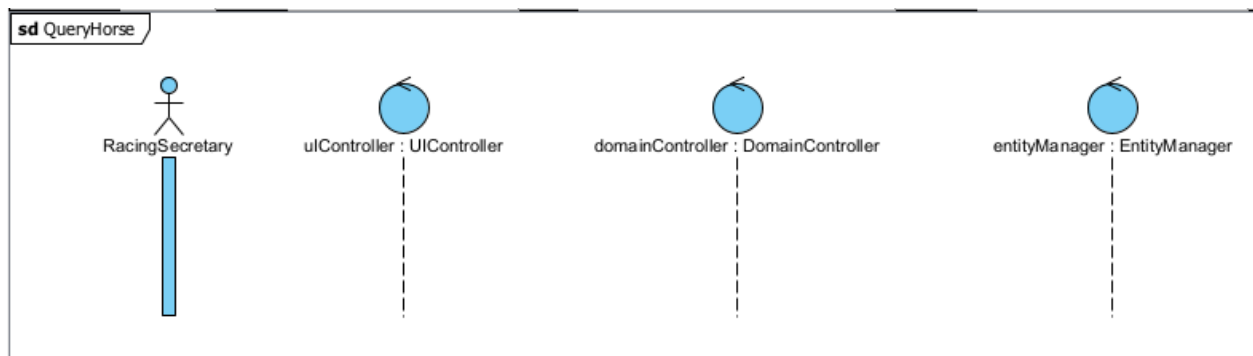
Click on Model Explorer on the left-hand side of the screen. Expand the Controllers class diagram.



Click on UIController and drag the class to your diagram. Choose Lifeline for the class.



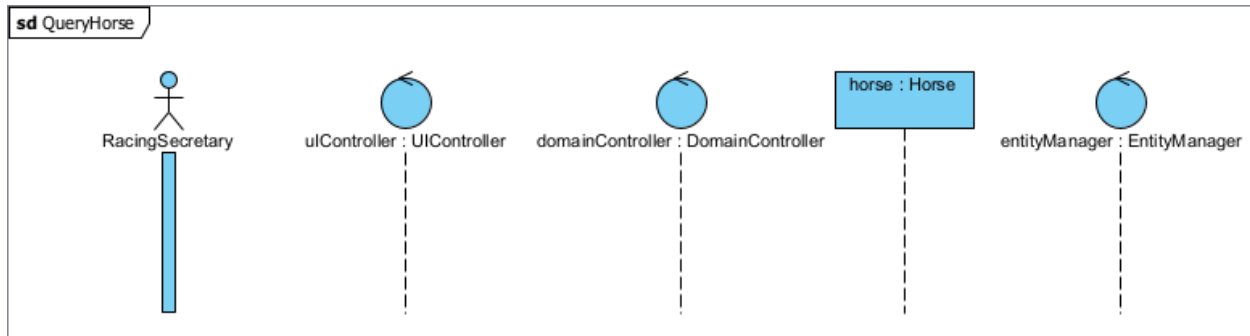
Repeat for DomainController and EntityManager.



We will be working with our Horse class, so we'll add that class to our diagram using the same method.

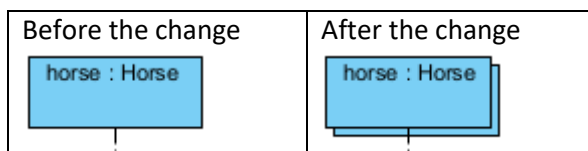
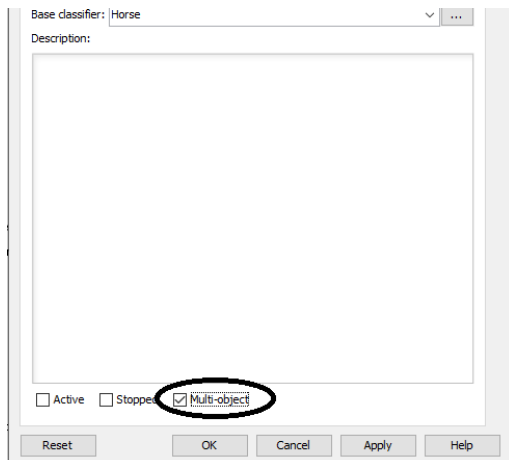
Click on Model Explorer, then click and drag Horse on to the diagram. Choose lifeline.

Most of work will be done between the Domain Controller and the Entity Manager, so drag the Horse class between the Domain Controller and Entity Manager.



How to change an object to a Multi-Object in a Sequence Diagram

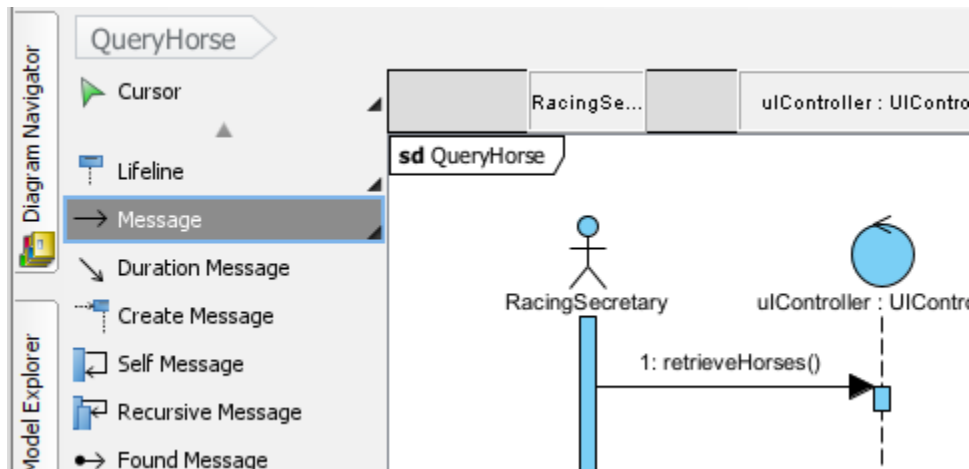
Often, we retrieve lists of data and create multiple instances of objects. To denote that we have a collection of objects, we define a class as a multi-object. To do this in your sequence diagram, select the class lifeline. Right click and select Open Specification. Click the Multi-object check box.



Defining Messages in Sequence Diagrams

Messages are eventually coded as methods in our applications. Message names usually follow verbNoun() sequence.

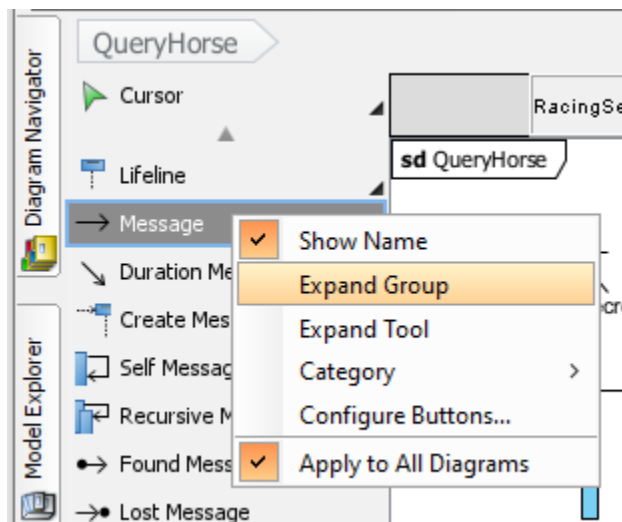
To define a message, use the Message icon on the tool bar. Click and drag from the starting lifeline to the end lifeline. Name the message.



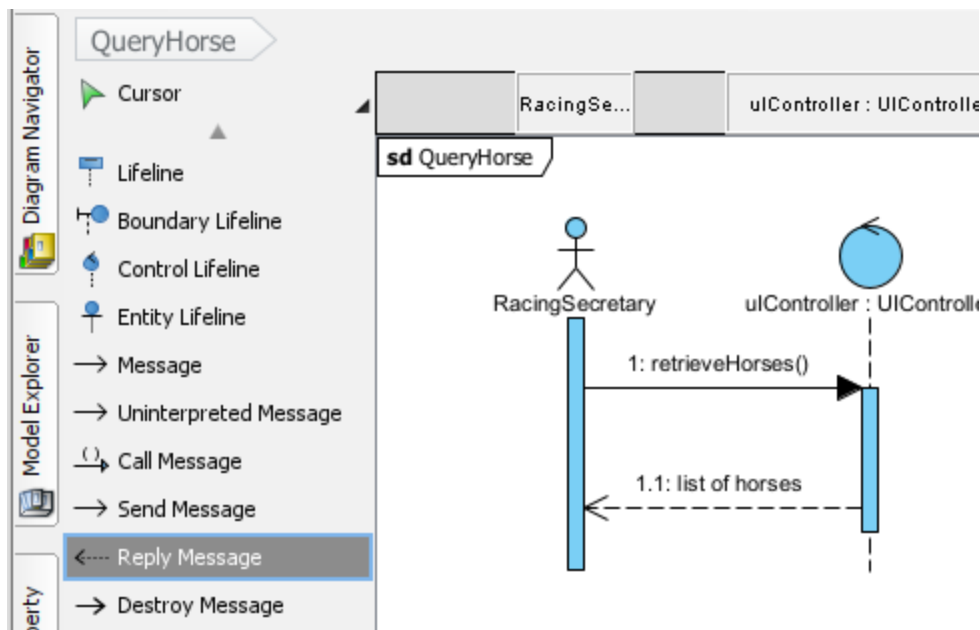
Defining Replies in Sequence Diagrams

Reply messages contain data only. We **NEVER use verbs** in reply messages.

To create a reply message, expand the message icon by right clicking on it and selecting expand group.

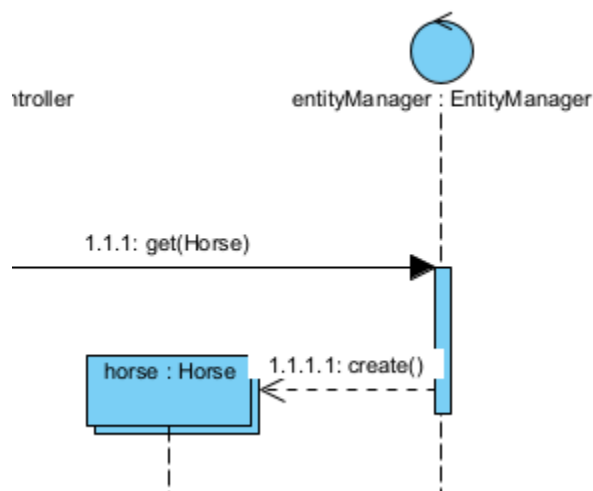


Click Reply message and drag the reply message from the starting life line to the end life line



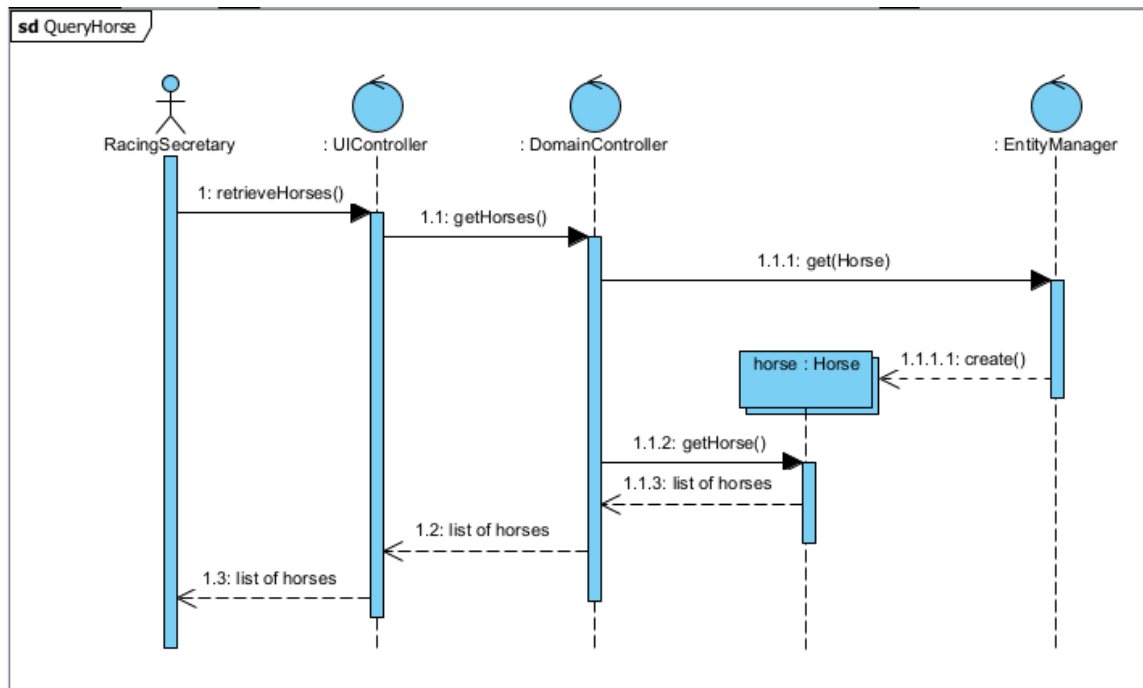
Defining Create Messages in Sequence Diagrams

Create messages are usually sent from the Entity Manager to instantiate objects created from data retrieved from the data source, or from the Domain Controller to instantiate a newly created object. To add a create message to your diagram, click on Create Message and then drag from the starting lifeline to the class lifeline. Change the name of the message to create(). The message will appear as a dotted message and the ending lifeline will lower accordingly in your diagram.



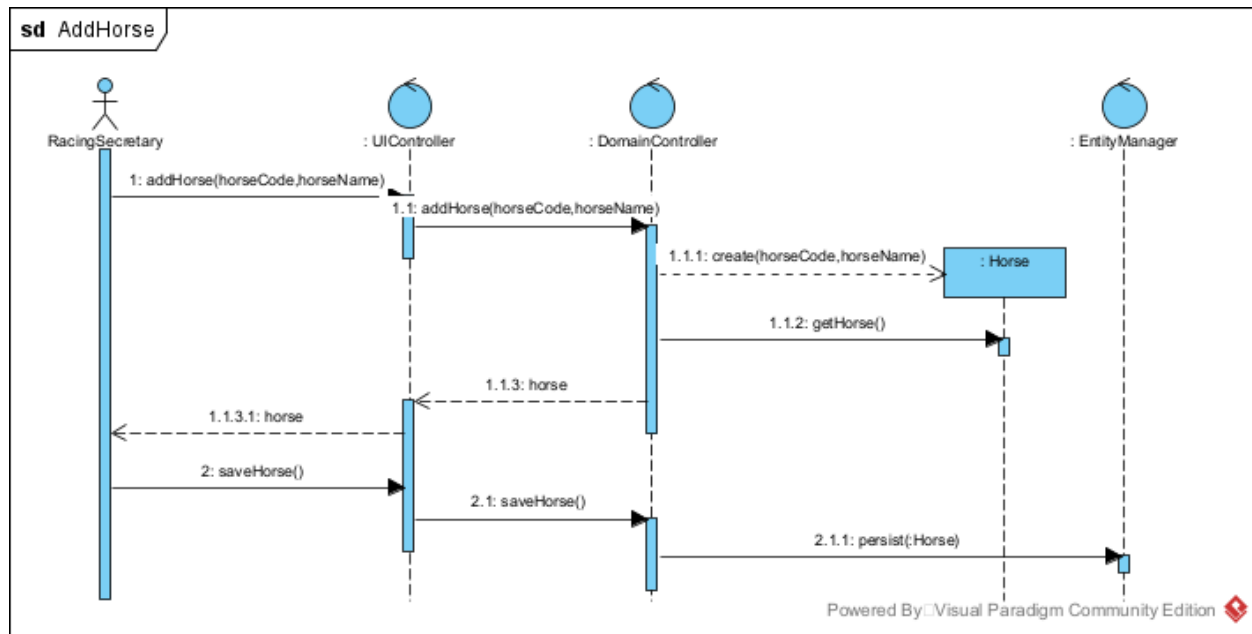
How to Retrieve a List of Objects from the Entity Manager

The following sequence diagram demonstrates how to retrieve a list of data from the Entity Manager and display the list to the actor.



Please note that the Entity Manager can only receive get, persist and remove messages. The syntax for a get messages is `get(ClassName, selectCriteria)`. If the `selectCriteria` is omitted, then all data is retrieved.

How to Create a New Object (Add) and Save it to the Entity Manager

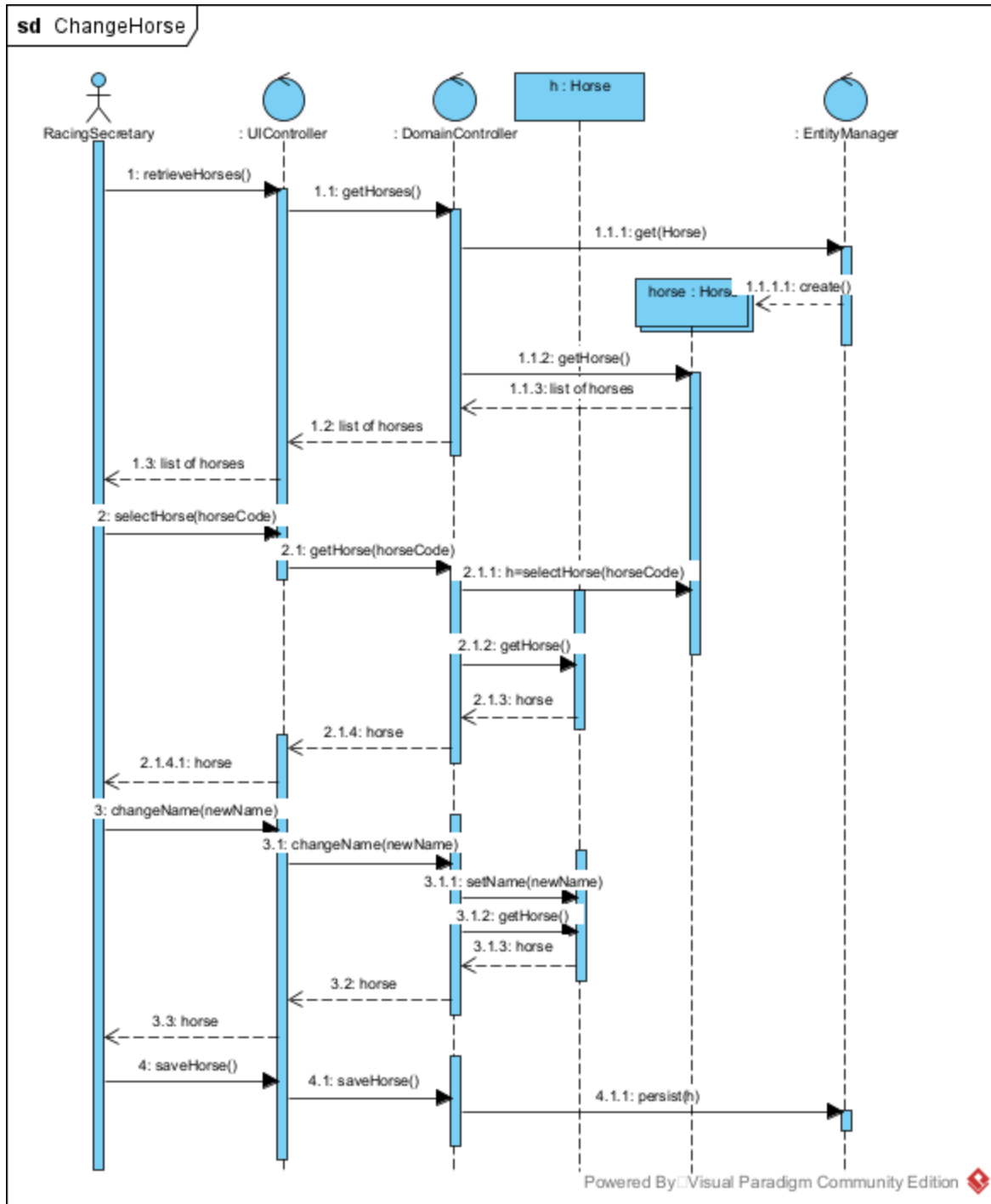


In this example, the Domain Controller starts the create message as we are adding a horse that doesn't exist in the data source.

The Domain Controller sends a persist message to the Entity Manager to save the Horse. The syntax for a persist message is `persist(Object,Object..)`. Be sure to add 'Set' to the object name if a collection of the same type of objects is saved.

How to Change an Object and Save it to the Entity Manager

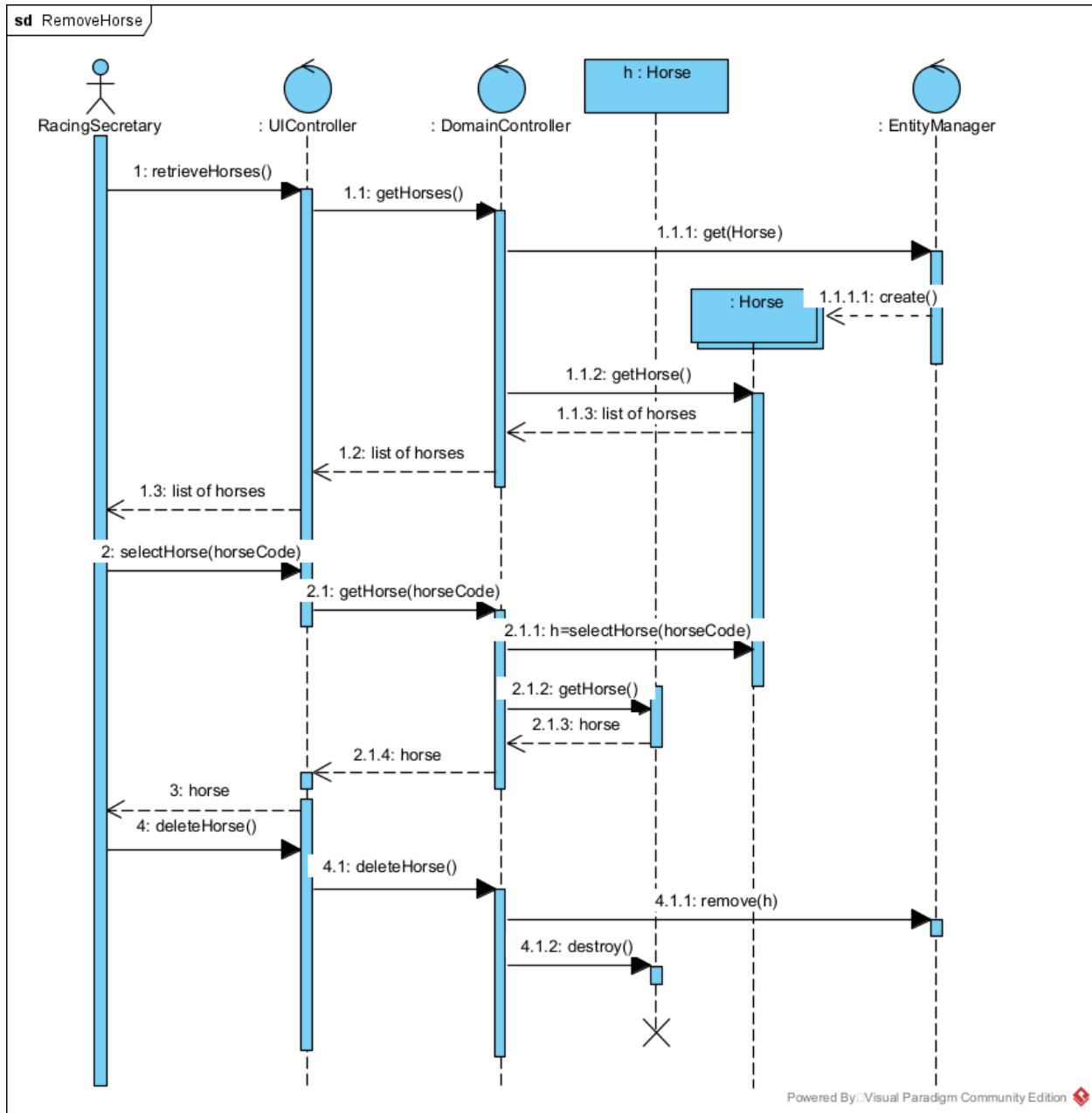
Before you can change an object, you must retrieve it from the data source.



Step 2.1.1 is to select the horse object that matches the requested horseCode. From there, we can update the horse name using a 'set' message. Persist(h) allows for the updated object to be saved to the data source.

How to Remove an Object and Delete from the Entity Manager

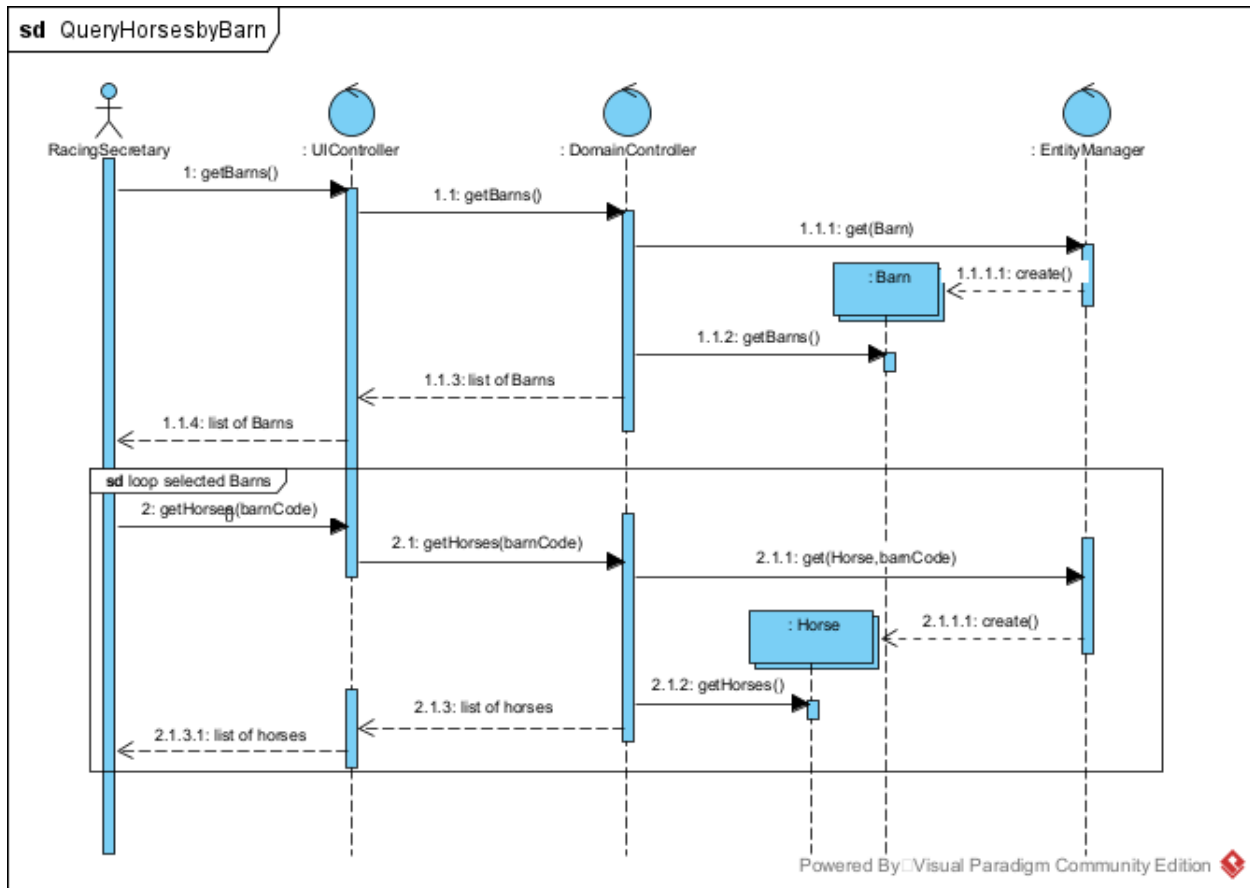
You must retrieve what you want to remove from the data source before you can remove it!



We send a remove message to the Entity Manager to delete the data from the data source and then use the Destroy Message to destroy the object in memory.

Loops in Sequence Diagrams

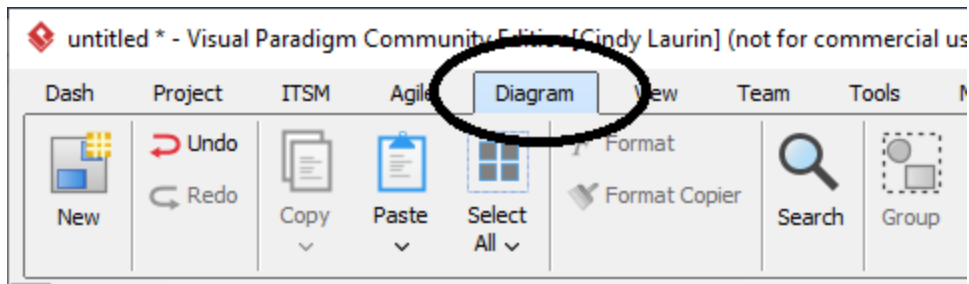
We use frames for define loops in our sequence diagrams.



This example allows the Racing Secretary to select a barn and then retrieve all of the horses residing in that barn, in a loop.

Exporting Diagrams and Copying to Word

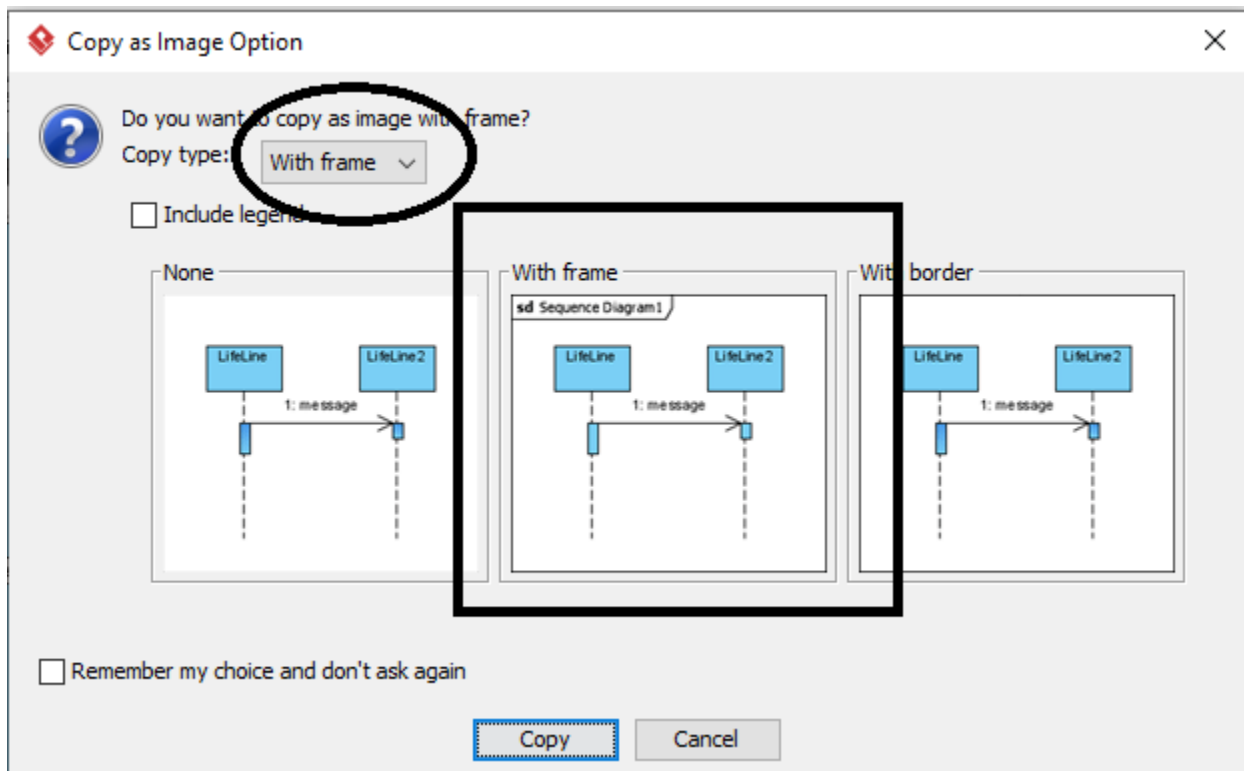
Select the Diagram menu.



Choose Select All.

Choose Copy and choose To Clipboard as Image (JPG).

Change the copy options as follows:



This will allow you to paste your diagram into word. Your diagram name will be included!