



---

# Introduction to Ventity

**ISDC 2020**

## Agenda

- **Logistics**
- **Introduction**
- **The environment**
- **Concepts & components**
- **Model exploration**
- **Modify a competitive dynamics model**

## Setup

- **Download the course material folder and unzip to a convenient location**
  - Source: <http://ventanasystems.com/conference> , or use the conference schedule page
  - Choose a location you control, i.e. not locked by your operating system
- **Install Ventity 4 from <http://Ventity.biz/download>**
  - Licenses are unlocked through the end of September

## Goals

### **Mechanics**

- **Environment**
- **Diagramming**
- **Equations**
- **Units**
- **Run control**
- **Run naming**
- **Entity initialization data**
- **Charts/tables**
- **Slider controls**
- **Entity picker**
- **Time series data**

### **Modeling Concepts**

- **Model-data separation**
- **Entities**
  - Types vs. instances
  - Design - what goes where?
  - State focus
  - Attributes
- **Attributes**
- **References**
- **Collections & aggregates**
- **Actions & triggers**
- **Relationship entities**

## Orientation

- **Diagramming**
- **Equations**
- **Units**
- **Run control**
- **Naming**
- **Entity Initialization Data**
- **Output**
- **Charting**
- **Tracing**
- **Picking**
- **Macros**
- **GIS**

**File & Run Controls**

**Attribute/Reference Grid**

**Diagram Toolbar**

**Model Overview**

**Diagram Inspector**

**Entity Picker**

**Console & Error List**

The screenshot displays the Ventana Supply Chain 7 software interface. The central area features a complex supply chain model diagram with entities like Level, Supplier, Customer, Orders, In Transit, and Inventory, connected by various processes and data flows. Surrounding the diagram are several functional panels:

- Top Left:** File & Run Controls menu and Model Overview tree.
- Top Center:** Attribute/Reference Grid showing relationships between entities and attributes.
- Top Right:** Diagram Toolbar with icons for stock, auxiliary, lookup, attribute, link, trigger, label, notes, chart, button, image, multi-slider, and multi-chart.
- Right Side:** Diagram Inspector panel for the 'supplychainlevel' entity, showing a list of attributes and their values.
- Bottom Left:** Entity Picker panel for selecting entity types.
- Bottom Center:** Console & Error List panel showing system messages and warnings.
- Bottom Right:** Data tables for 'Orders' and 'Inventory' showing columns for Entity, Min, Max, and Input.

**Orders Table:**

Entity	Min	supplychainlevel.supply line att...	Max	Input
retailer	0		2	1
wholesaler	0		2	1
distributor	0		2	1
factory	0		2	1

**Inventory Table:**

Entity	Min	supplychainlevel.stock adjustm...	Max	Input
retailer	2		8	4
wholesaler	2		8	4
distributor	2		8	4
factory	2		8	4

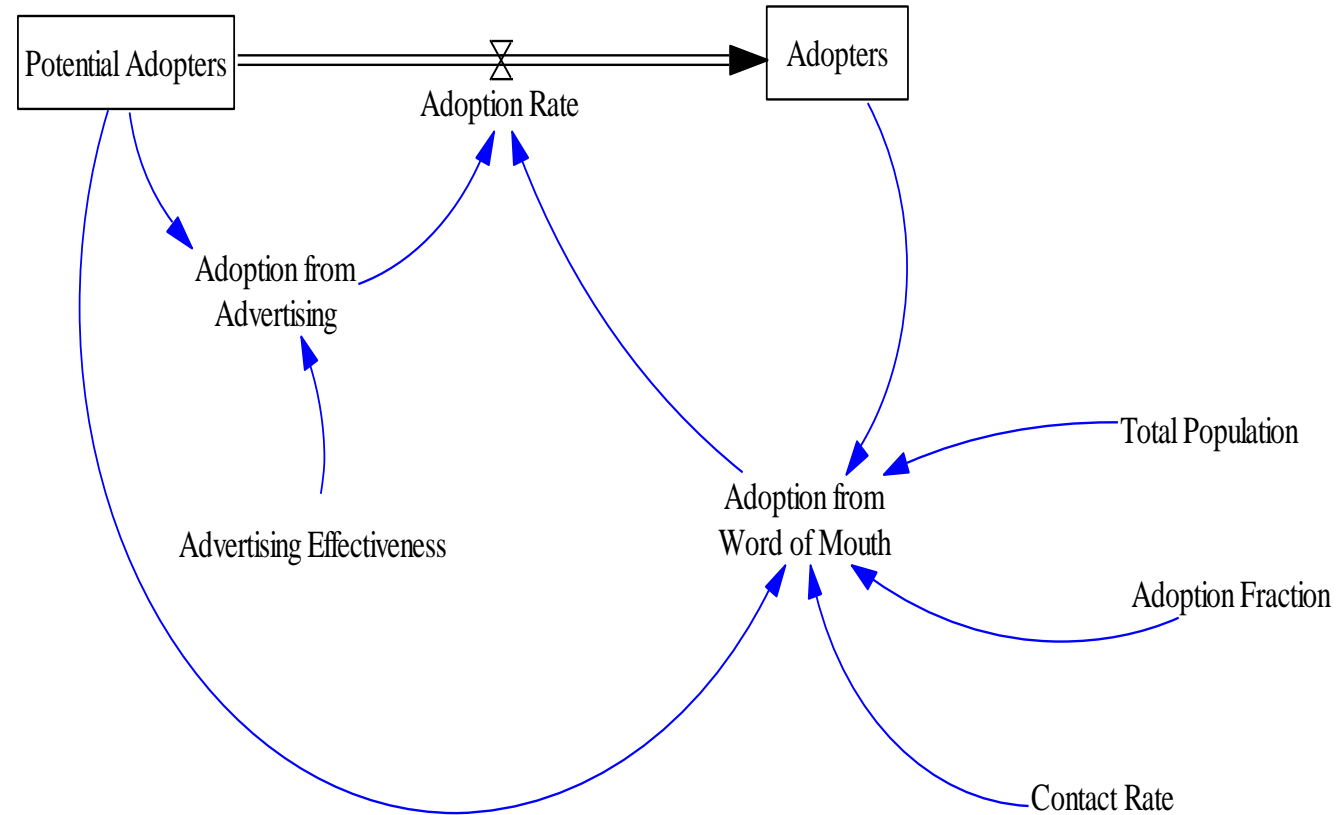
# **ENTITIES & WHAT GOES WHERE?**

## Principles

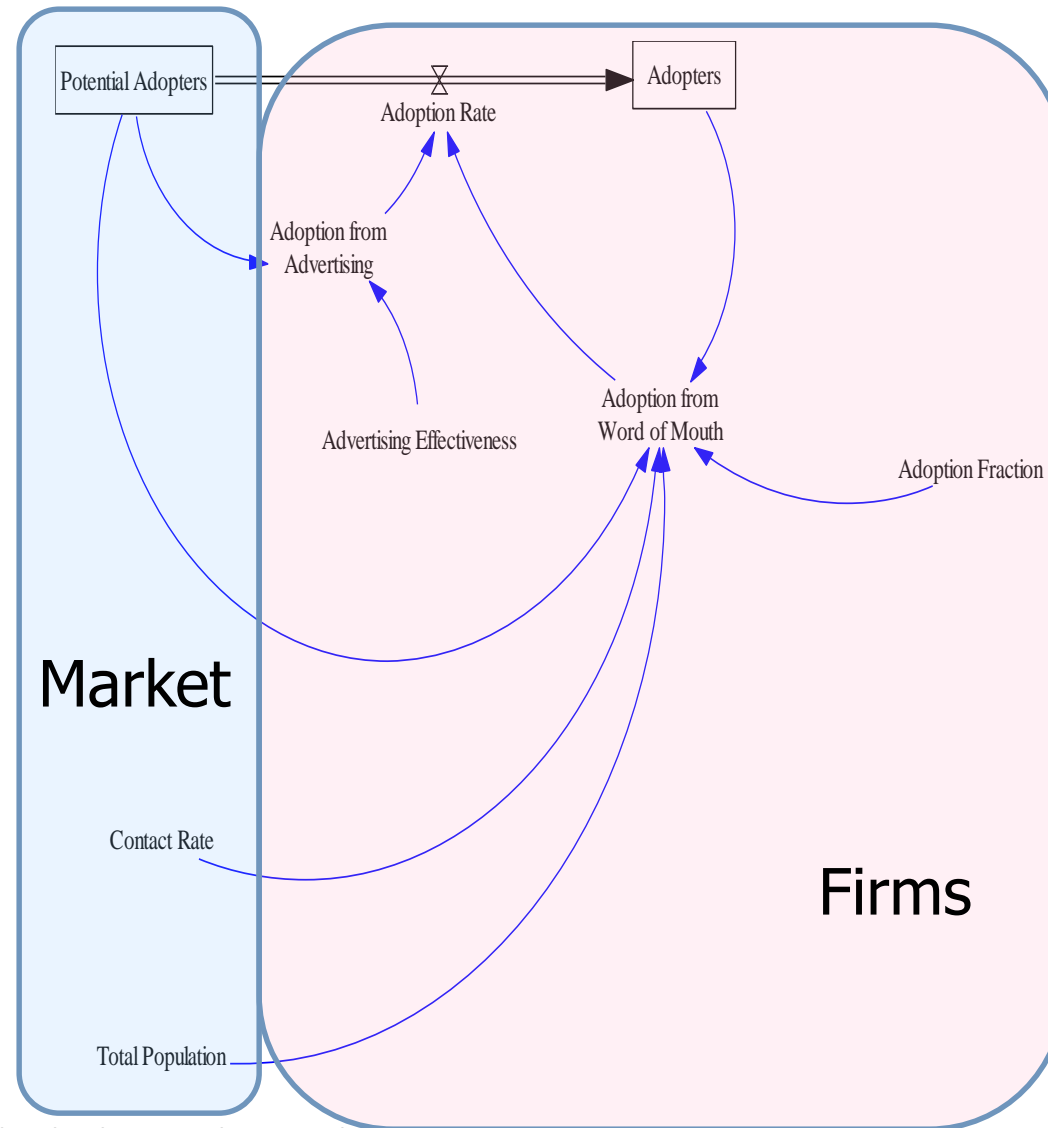
- **Everything in an Entitytype shares a common level of detail**
- **Variables belong in the Entitytype with the required level of detail**
- **Global parameters can go in the Model (or another singleton)**



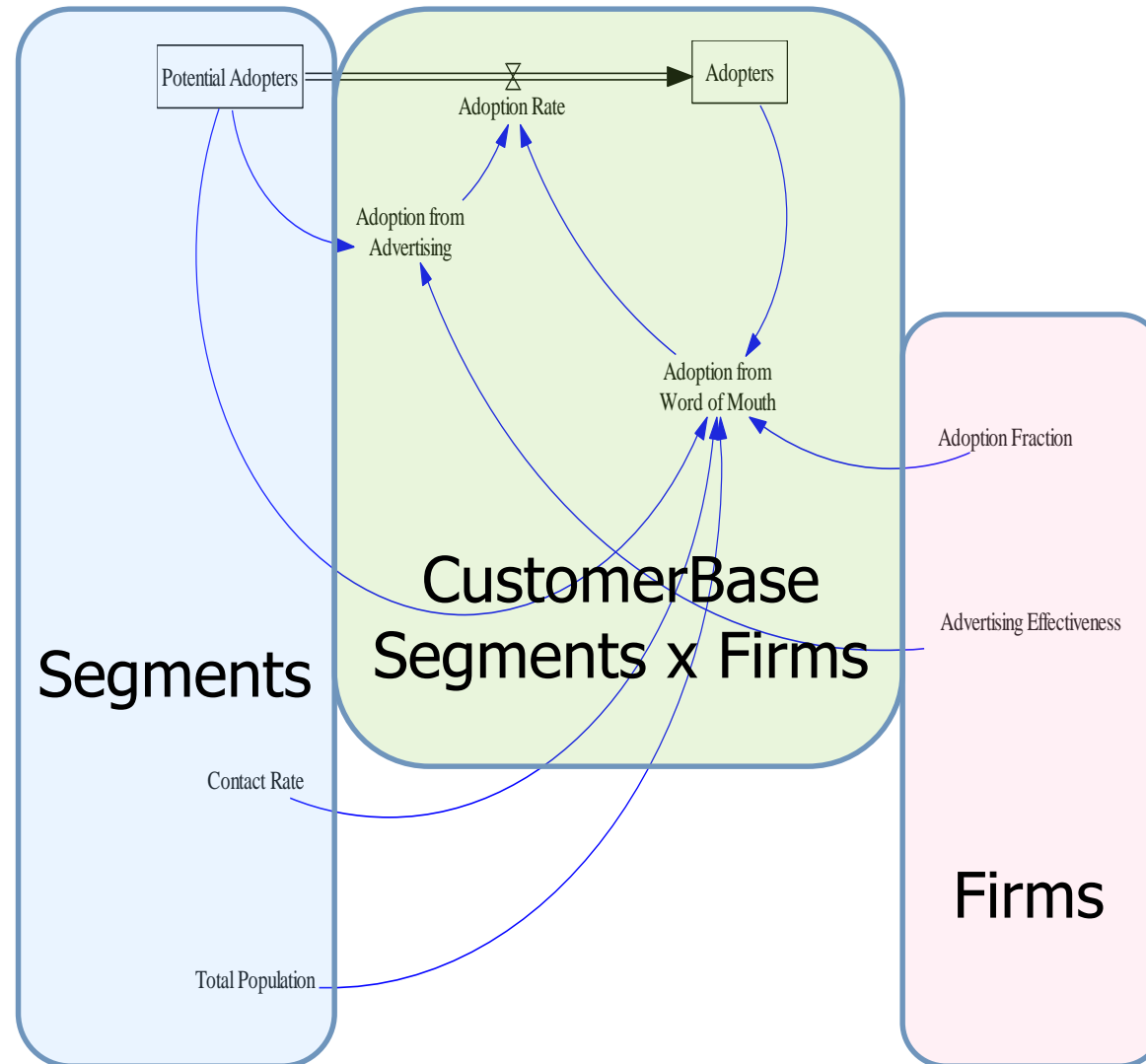
# 1. Classic Bass Model



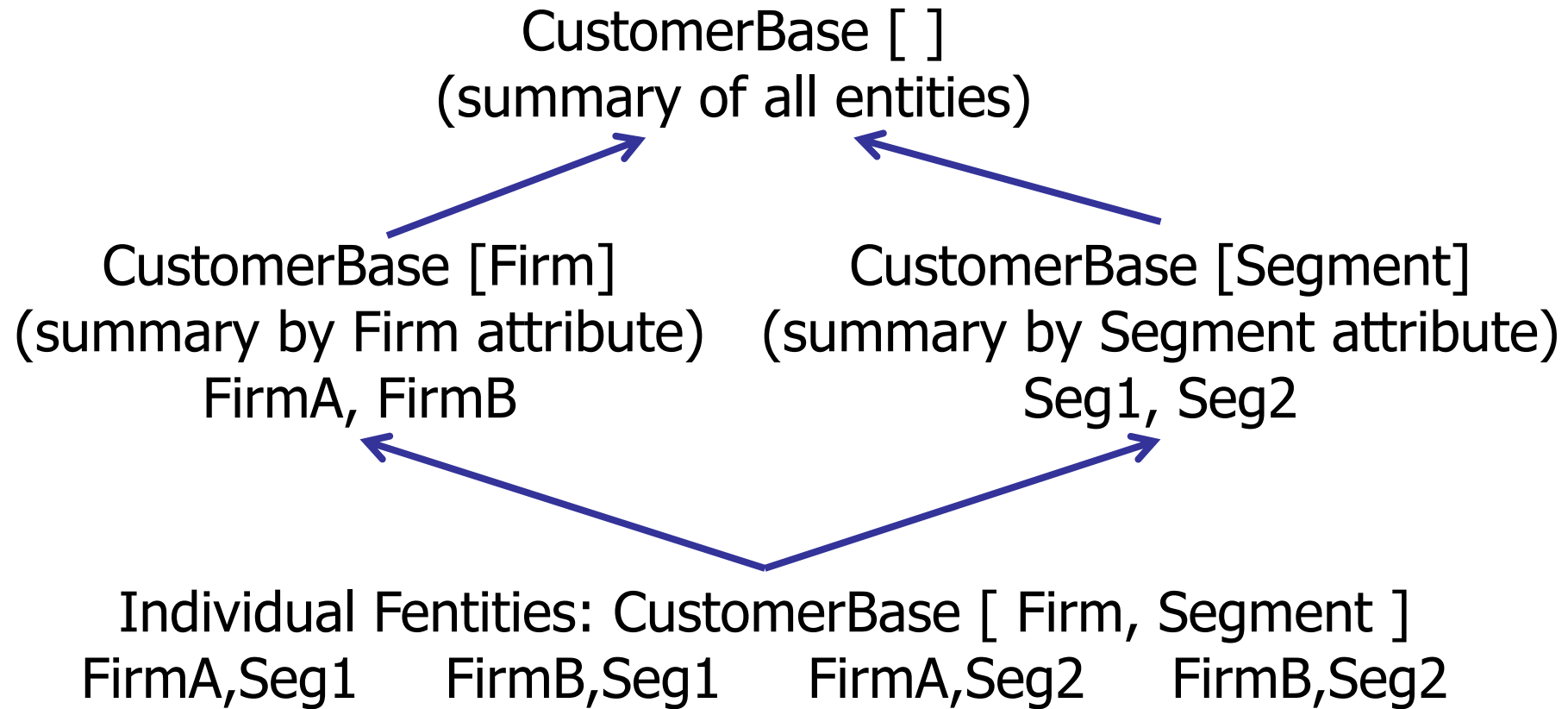
## 2. ... with multiple firms



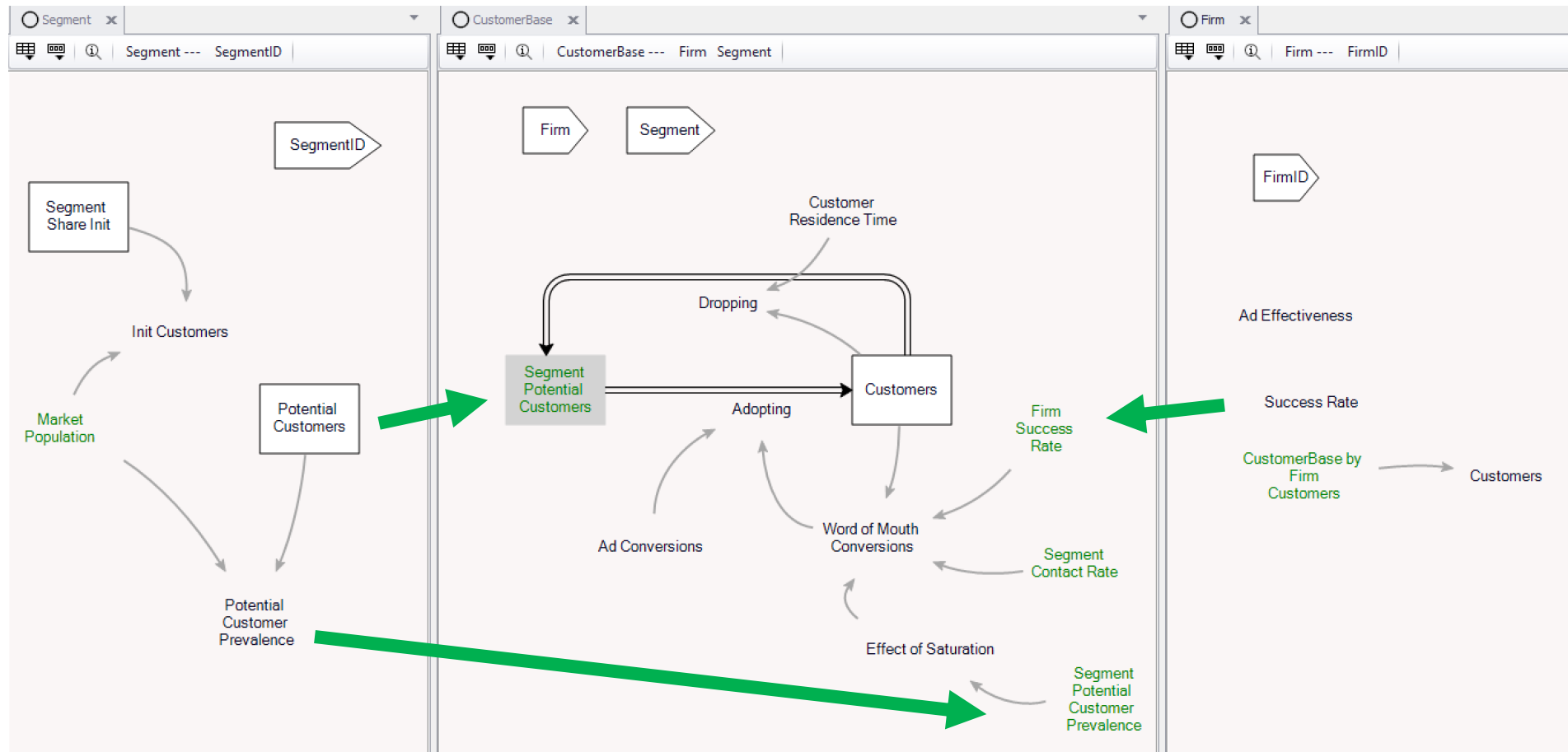
### 3. ... multiple segments & firms



## Collection Hierarchy



# Implementation



## 1-2 Data

- **There are 2 kinds of “data”:**
  1. Entity Initialization Data, used to set up the model entities (similar to Vensim .cin files)
  2. Time Series Data, used for comparison or driving model variables

# 1. Entity Initialization Data

CustomerBase						Firm
	<input type="checkbox"/>	Time	CalendarTime	FirmID	Success Rate	
▶ 1	<input checked="" type="checkbox"/>	0		FirmA	0.009	
2	<input checked="" type="checkbox"/>	0		FirmB	0.011	
3	<input type="checkbox"/>		07/21/2014	NewFirm	0.015	
*						

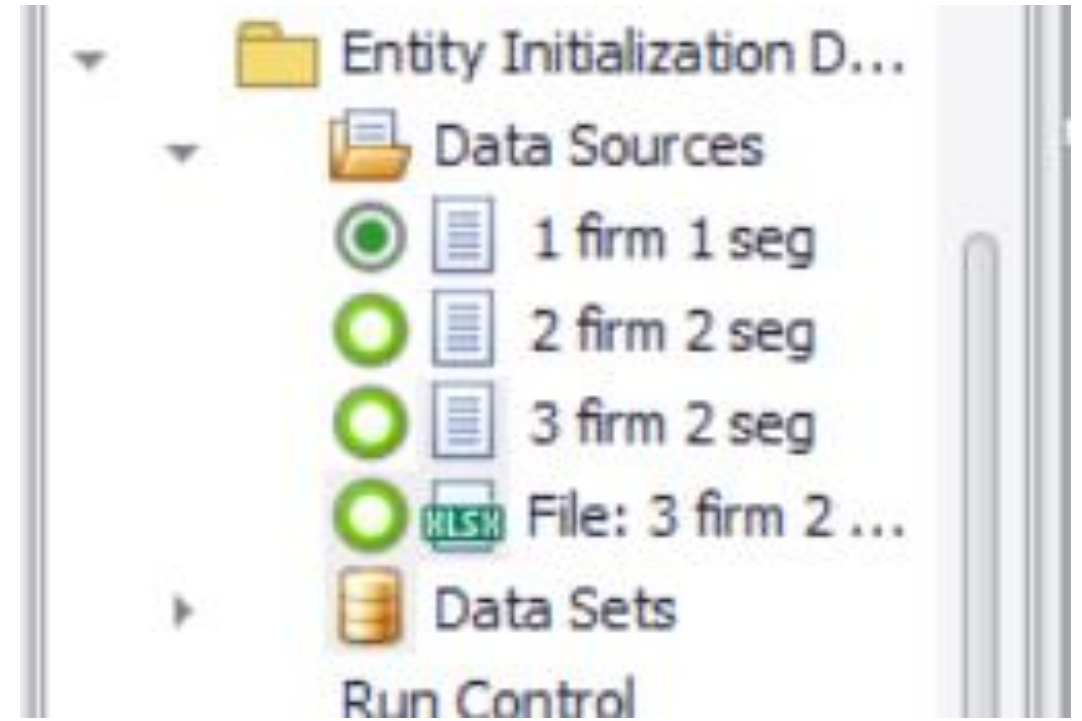
CustomerBase						Firm
	<input type="checkbox"/>	Time	CalendarTime	Firm	Segment	Customers
▶ 1	<input checked="" type="checkbox"/>	0		FirmA	Segment1	
2	<input checked="" type="checkbox"/>	0		FirmB	Segment1	
3	<input checked="" type="checkbox"/>	0		FirmA	Segment2	
4	<input checked="" type="checkbox"/>	0				
5	<input type="checkbox"/>					
*	<input type="checkbox"/>					

CustomerBase		Firm		Market	Model	Segment
	<input checked="" type="checkbox"/>	Time	CalendarTime	SegmentID	Potential Customers	Segment Share Init
▶ 1	<input checked="" type="checkbox"/>	0		Segment1		0.25
2	<input checked="" type="checkbox"/>	0		Segment2		0.75
*	<input type="checkbox"/>					

## 1a. Varying the data selection

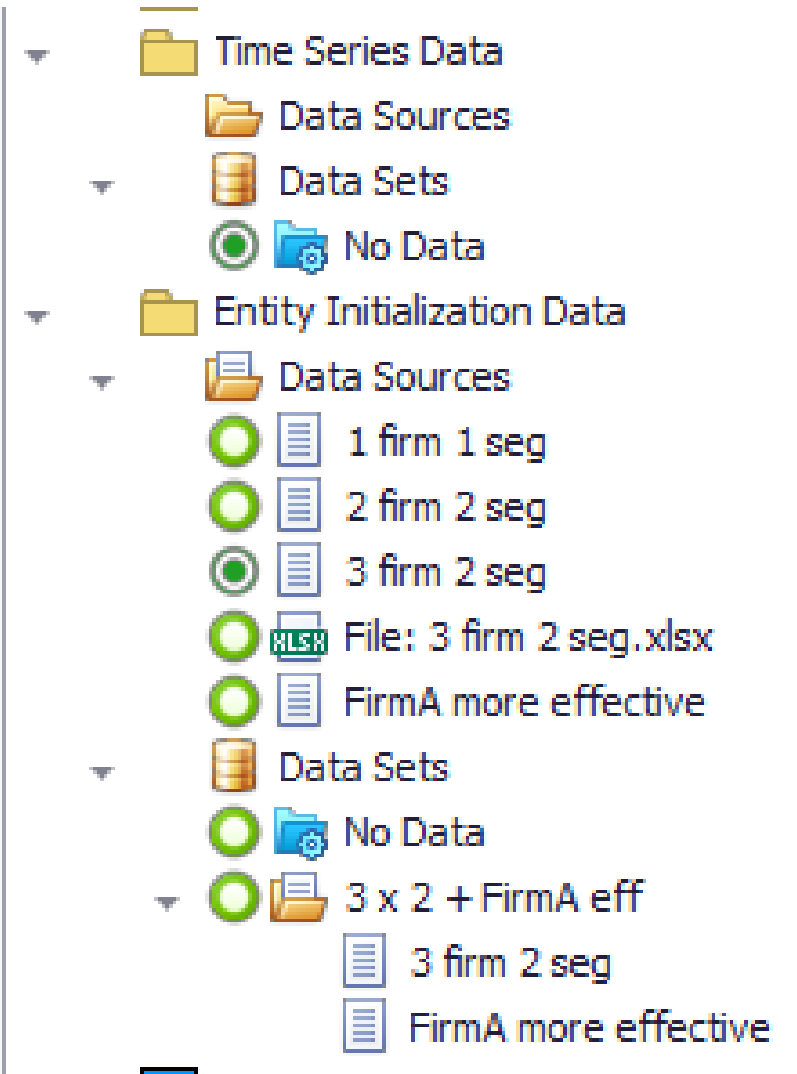
- **Sources and Sets**
- **Run Control and data**
- **On-the-fly data selection**
- **Entity picking**





## 1b. Sources & Sets

- A “Source” is a single file
- A “Set” loads multiple Sources
- As in Vensim, order matters: the last set loaded wins in the event of a conflict



## 1c. The Run Control

- **Run Control settings determine which data sources or sets to use**

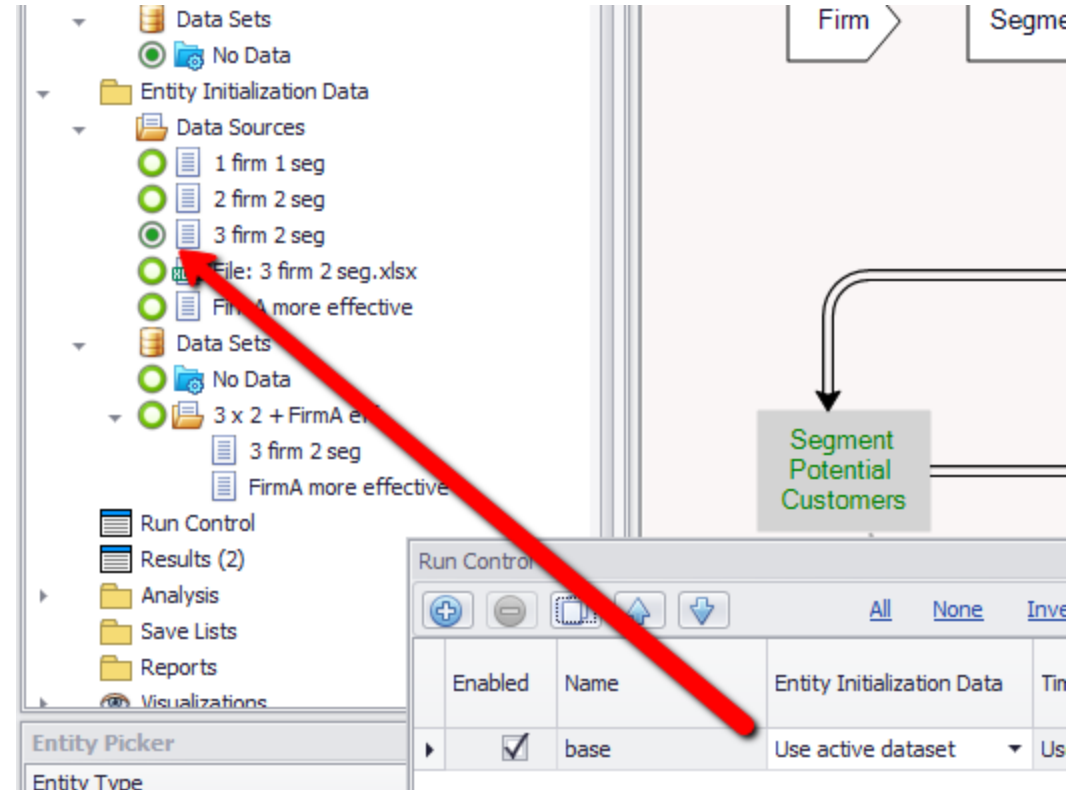
The screenshot displays the Ventana software interface. On the left, a tree view shows the project structure, with 'Run Control' highlighted. A red arrow points from this item to the 'Run Control' dialog box. The dialog box contains a table with the following data:

Enabled	Name	Entity Initialization Data	Time
<input checked="" type="checkbox"/>	base	Use active dataset	Us
<input checked="" type="checkbox"/>	better FirmA	3 x 2 + FirmA eff	Us

A red arrow points from the 'better FirmA' row to the '3 x 2 + FirmA eff' value in the 'Entity Initialization Data' column. In the background, a diagram shows a box labeled 'Segment Potential Customers' with an arrow pointing to it from above.

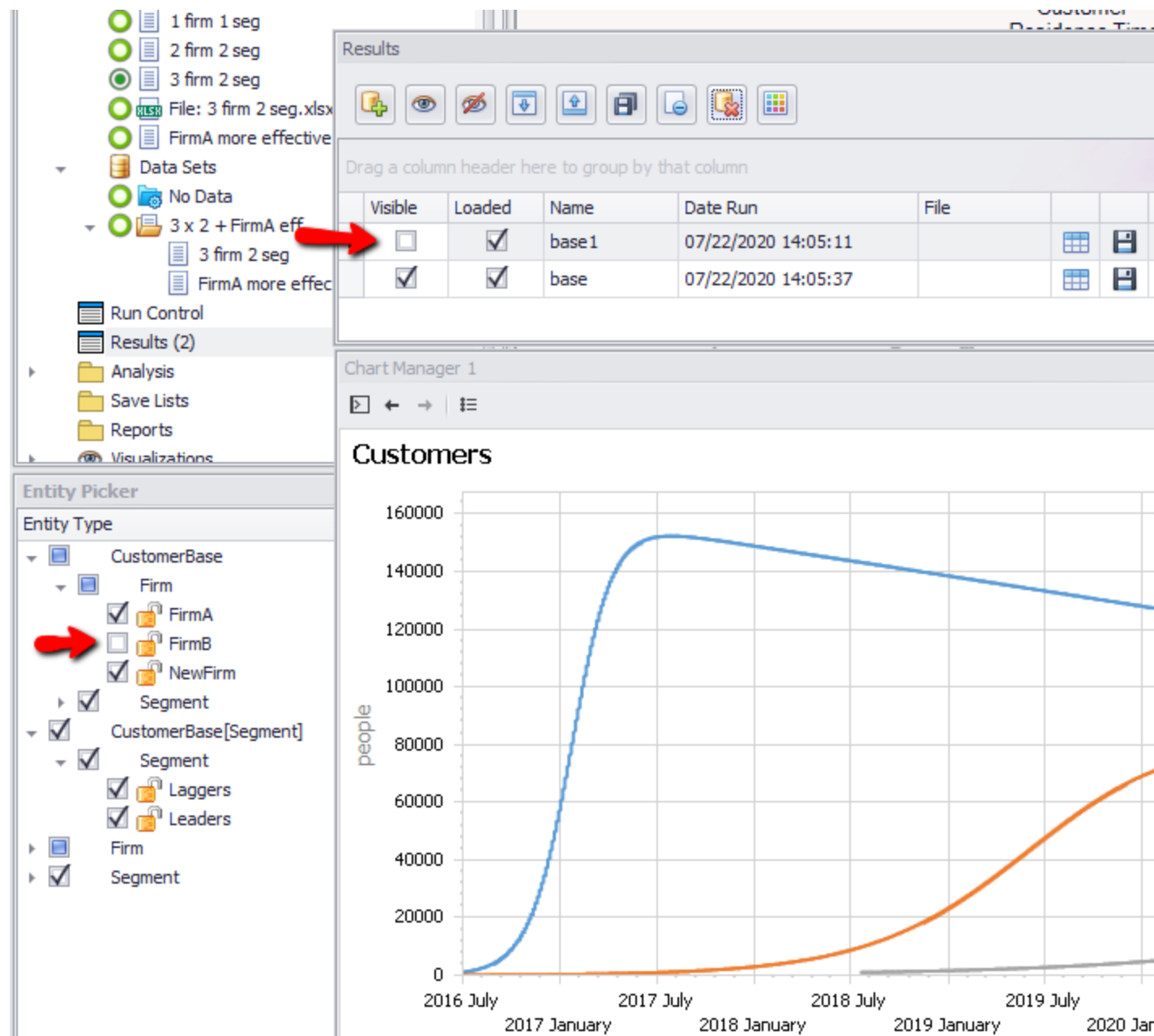
## 1c. On-the-fly selection

- If “use active dataset” is selected in the Run Control, the radio button in the Entity Initialization Data section controls the active data



## 1d. Selecting data visibility

- Initialization data determines which entities exist
- The Entity Picker selects which entities are visible
- The Results pane selects which runs are visible



## 2. Time Series Data

- **Everything about Sources, Sets and selection applies**
- **Formatting the data for import**
- **Mapping**
- **Using data**

## 2a. Formatting Data

- Easiest to use the Help as a cookbook

Ventity™ Help

Hide
 Back
 Forward
 Home
 Print
 Options

[Contents](#)
[Index](#)
[Search](#)
[Favorites](#)

- Welcome
- Examples
- Defining Entity Types
  - Create a New Model
  - Import Entity Types
  - Importing Vensim Models
- Create a New Entity Type
  - Custom Macros
- Data Entities
  - Data Sources**
  - Data Entities and Maps
- Model Encryption
- Initializing Entities
- Simulating
- Examining Results
- The Ventity Workspace
- Further Information

Navigation: Defining Entity Types > Data Entities >
 

### Data Sources

Excel files

A data source is a Microsoft® Excel spreadsheet or workbook containing data associated with particular times. Typically this means time figures over a 6 year period), but it can also mean a single data point (the population in September, 2012). Data files can be stored anywhere in any subfolder of the model folder.

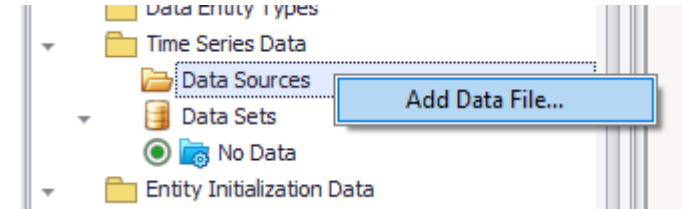
A single worksheet or tab will generally contain information for many entities, but all the entities on a single worksheet or tab must be of the same type. If you have more than one sheet for an entity type (say, Platypus), the sheets should be named Platypus, PI, etc. The text after the underscore does not have to be A, B, C; it can be whatever you like. The requirement is that each sheet name begins with the name of the entity type and the further text to make the name unique.

The worksheet may be organized in any one of four common layouts. Ventity automatically detects which one is being used, based on keywords as follows. The examples shown are for variables Weight and Age of entity type Platypus having key attribute PlatypusID.

Layout	Description	Requirements	Example																														
Tidy	Each row specifies an entity and time; each column, a variable	First row contains keyword "DateTime" as well as names of variables and <a href="#">key attributes</a>	<table> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>D</th> <th>E</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>PlatypusID</td> <td>DateTime</td> <td>Weight</td> <td>Age</td> <td></td> </tr> <tr> <td>2</td> <td>Albert</td> <td>0</td> <td>30</td> <td>14</td> <td></td> </tr> <tr> <td>3</td> <td>Albert</td> <td>1</td> <td>35</td> <td>15</td> <td></td> </tr> <tr> <td>4</td> <td>Beatrice</td> <td>0</td> <td>28</td> <td>12</td> <td></td> </tr> </tbody> </table>		A	B	C	D	E	1	PlatypusID	DateTime	Weight	Age		2	Albert	0	30	14		3	Albert	1	35	15		4	Beatrice	0	28	12	
	A	B	C	D	E																												
1	PlatypusID	DateTime	Weight	Age																													
2	Albert	0	30	14																													
3	Albert	1	35	15																													
4	Beatrice	0	28	12																													

## 2b. Creating a Mapping

- **Add the file SyntheticData.xlsx**
- **Detect Source Entity Types**
- **Create Data Entity Types**



Time Series Data Mapping - File: SyntheticData.xlsx

Mapping Name: SyntheticData.xlsx  
 File: SyntheticData.xlsx

Time Series Data Description:

**Detect Source Entity Types** **Create Data Entity Types**

Source Entity Type: **customerbase**

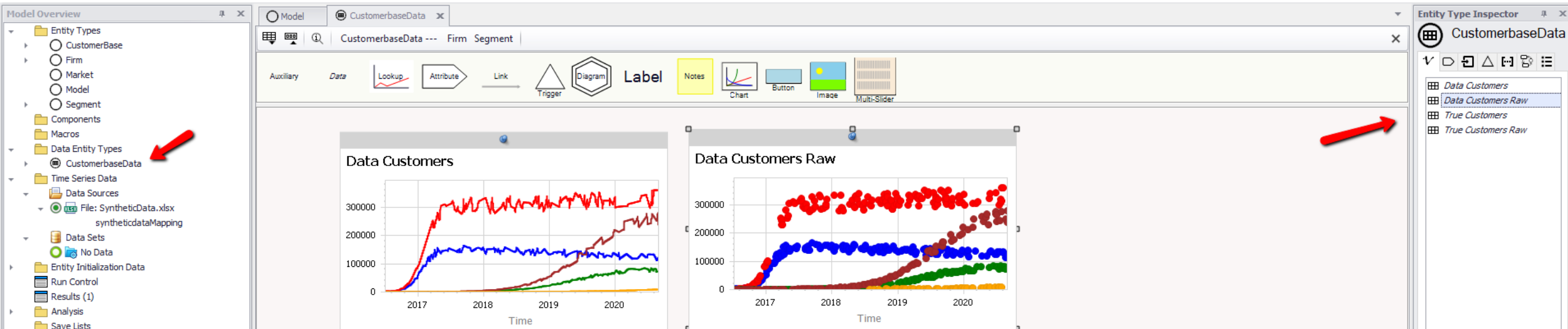
Data Entity Type: Select Data Entity Type **CustomerbaseData**

**customerbase**

Source Name	Source Type	Interpolation Type	Aggregation Type	Source Units	Source Description	Source Measurement Type	Data Name	Data Type	Data Key
Firm	Key						Firm		
Segment	Key						Segment		
True Custo...	Variable	Straight					True Custo...		
Data Custo...	Variable	Straight					Data Custo...		

## 2c. Viewing Data

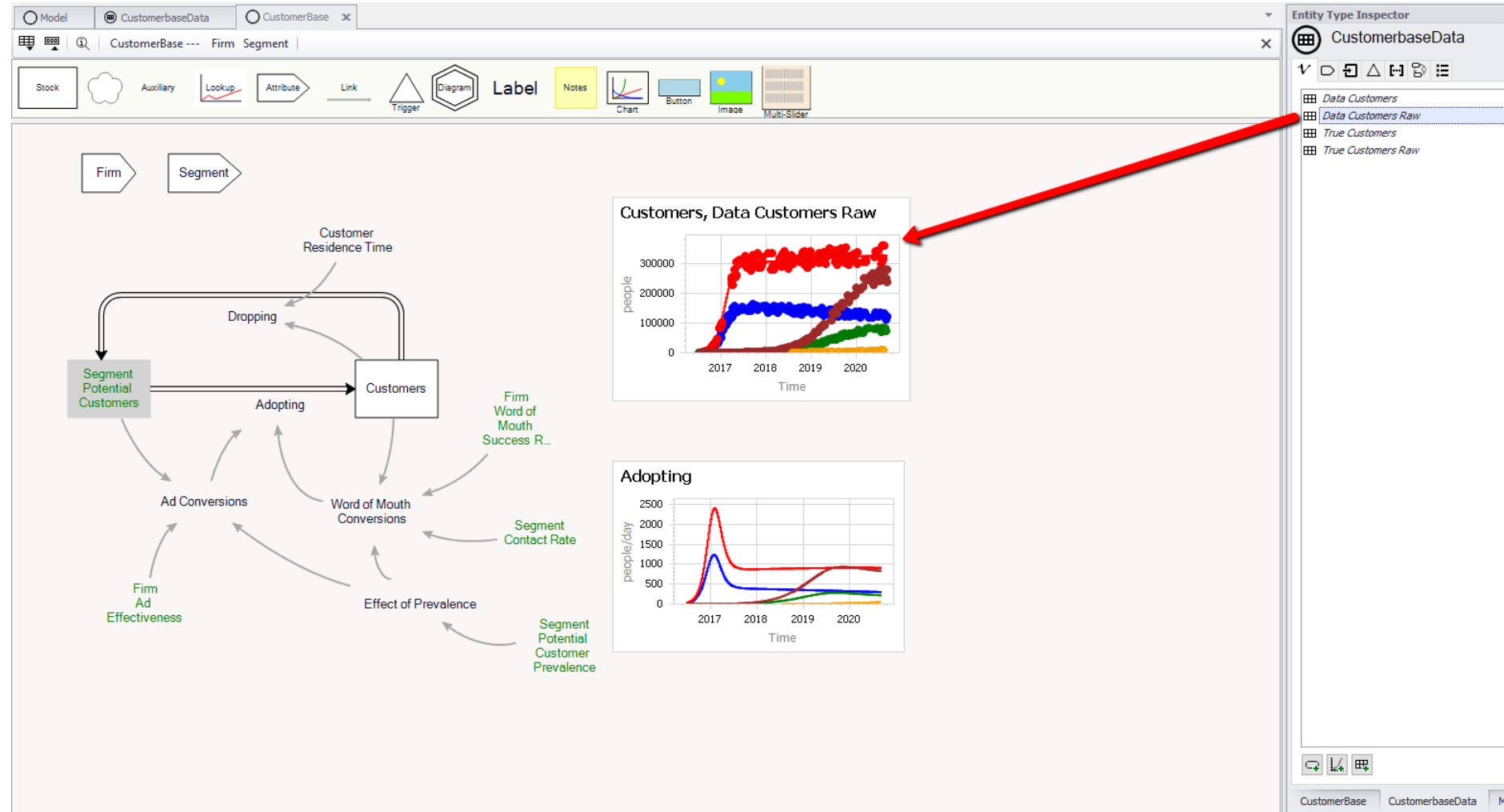
- Now you have a data entity containing the data variables.
- There's no diagram (though you could drag variables from the inspector to create one)
- You can drop charts onto the diagram for viewing





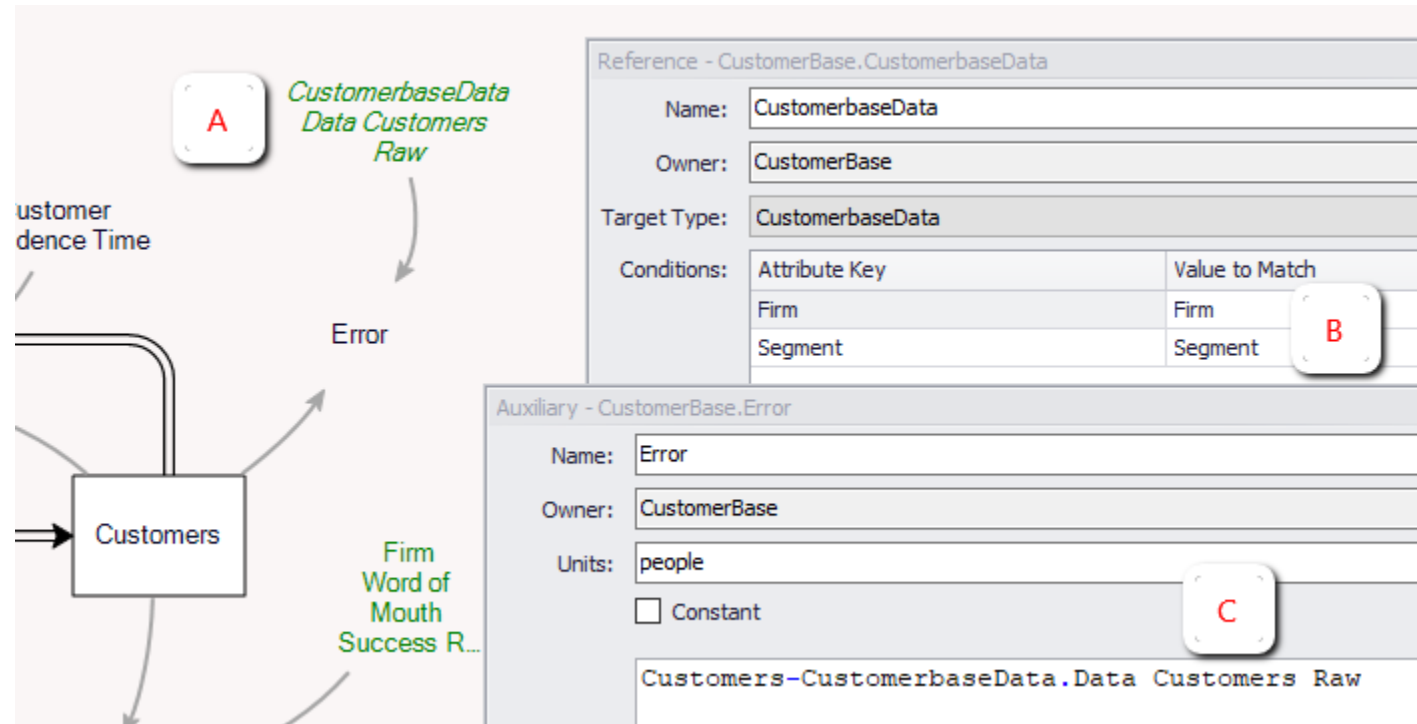
## 2d. Creating a Comparison

- Drag the data from the inspector onto a chart with the corresponding model variable



## 2e. Using the Data

- A. Drag the data variable onto the CustomerBase diagram**
- B. Right-click and edit the Reference to match Firm and Segment**
- C. Add an auxiliary to compute the Error as the difference between simulated Customers and the data.**



### 3. Add Financials to the Firm

- **Price and revenue**
- **Firm profit**
- **Sum revenue by Firm**
- **Reference the collection from the Firm**

## 3a. Adding Firm profit & loss

- **Options**
  - Build from scratch
  - Import a component (recommended)

**Import**

**Include**

**Go to New Diagram**

**Include Component**

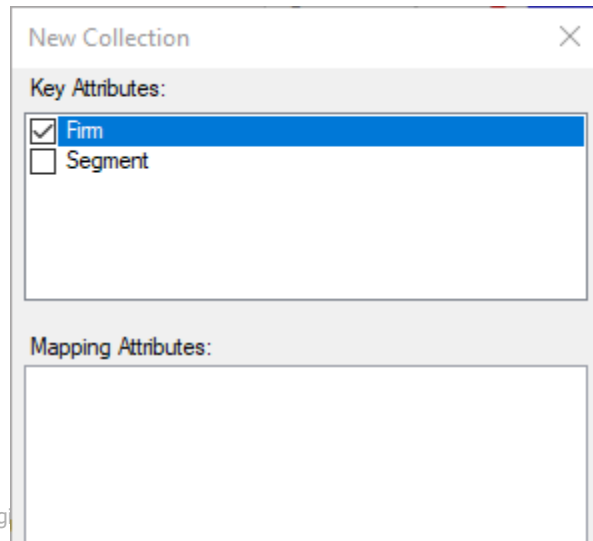
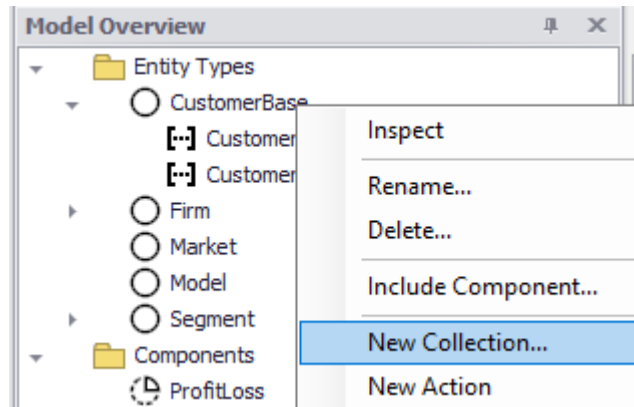
Component: ProfitLoss  
Prefix: ProfitLoss

Include Cancel

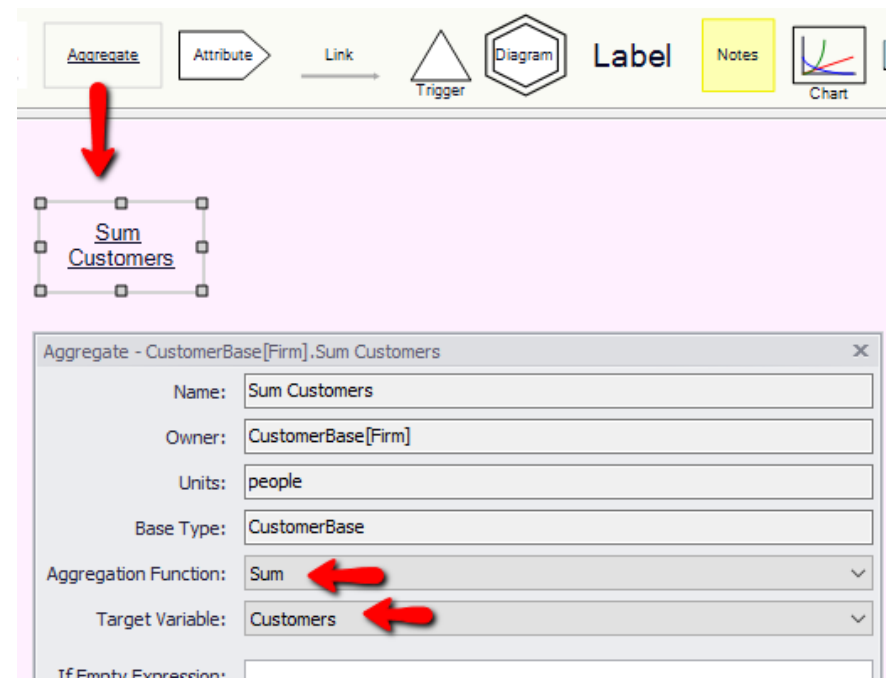
## 3b. Creating a Collection to Sum Customers

- **Need a source for Units Sold**
- **Assume 1 item per customer per day**
- **Same price to all customers**
- **So: need to know Sum(Customers)**

Create a new collection of CustomerBase by Firm



Create an aggregate to sum customers



**A. Drag Sum Customers onto the Firm Profit & Loss diagram**

**B. Edit the reference to match Firm to FirmID**

**C. Write the equation for Units Sold**

**D. Right-click to add Units per Customer per Day**

### 3c. Using the Sum

The screenshot shows the VENTANA software interface with several windows and a diagram.

- Collection Inspector:** Shows a tree view with "CustomerBase [Firm]" and "Sum Customers" (with a "Count" sub-item).
- Diagram:** A central diagram shows "ProfitLoss Price" and "ProfitLoss Revenue" connected to "ProfitLoss Units Sold" (which has a red 'X' icon). "ProfitLoss Units Sold" is connected to "ProfitLoss Gross Profit". A green arrow labeled 'A' points from "Sum Customers" in the Collection Inspector to a box labeled "CustomerBase by Firm Sum Customers" in the diagram.
- Reference - Firm.CustomerBase by Firm:** A window showing the reference configuration.
 

Attribute Key	Value to Match
Firm	FirmID

 A label 'B' is placed next to the "FirmID" value.
- Auxiliary - Firm.ProfitLoss Units Sold:** A window showing the auxiliary configuration.
  - Name: ProfitLoss Units Sold
  - Owner: Firm
  - Units: unit/day
  - Constant: ☐
  - Equation:  $\text{CustomerBase by Firm.Sum Customers} * \text{Units per Customer per Day}$
 A label 'C' is placed next to the equation.
- Context Menu:** A right-click context menu is open over the equation, showing options:
  - Create New Stock "Units per Customer per Day"
  - Create New Auxiliary "Units per Customer per Day"
  - Create New Attribute "Units per Customer per Day"
 A label 'D' is placed next to the menu.

## 4. Add Slider Controls & Charts

- **Where should we put them?**
  - The Model entity is often convenient for top-level controls
  - Detailed drill-down can live in other entities
- **Slider variables must be Constants**
- **A slider affects all runs, unless “Ignore First Run” = True**
  - This facilitates a nice treatment of base case vs. policy runs

## 4a. Adding a Slider

- Drag onto diagram
- Right-click to edit
- Choose the Bound Variable
- Edit settings as needed

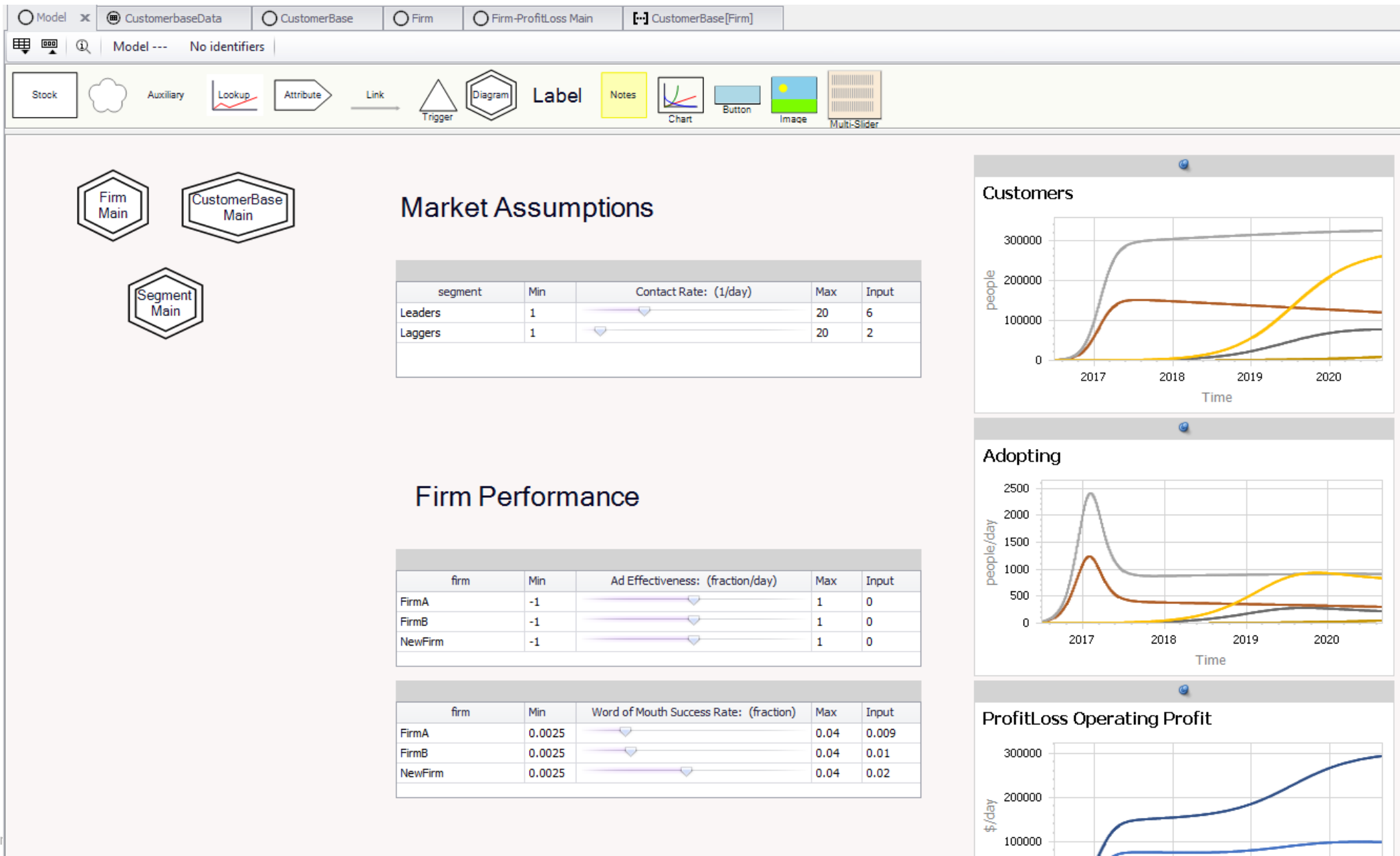
The screenshot shows the Ventana software interface. At the top, there is a toolbar with icons for Trigger, Diagram, Label, Notes, Chart, Button, Image, and Multi-Slider. Below the toolbar, the main area is titled 'Market Assumptions'. It contains a table with columns 'segment', 'Min', and 'Contact Rate: (1/day)'. The table has two rows: 'Leaders' and 'Laggers', both with a 'Min' value of 1. A red arrow points from the 'Multi-Slider' icon in the toolbar to the slider in the 'Contact Rate' column of the table. To the right of the table, there is a 'Misc' settings panel. It has a 'Name' field set to 'Segment.Contact Rate\_C1' and a 'Bound Variable' dropdown menu. The dropdown menu is open, showing a list of variables. A red arrow points from the 'Bound Variable' dropdown to the 'Segment.Contact Rate' option in the list. Other variables in the list include 'CustomerBase.Customer Residence Time', 'CustomerBase.Customers', 'Firm.Ad Effectiveness', 'Firm.Price', 'Firm.Word of Mouth Success Rate', 'Market.Population', and 'Segment.Population'.

segment	Min	Contact Rate: (1/day)
Leaders	1	
Laggers	1	

Misc	
Name	Segment.Contact Rate_C1
Bound Variable	Segment.Contact Rate
Entity Type	CustomerBase.Customer Residence Time
SliderCaption	CustomerBase.Customers
Precision	Firm.Ad Effectiveness
Auto Set Min/Max Values	Firm.Price
Minimum	Firm.Word of Mouth Success Rate
Maximum	Market.Population
	Segment.Population



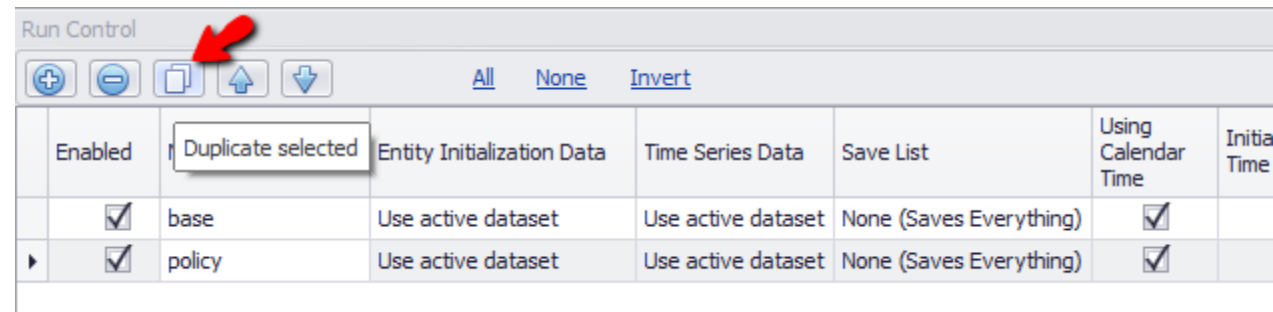
## 4b. A possible layout



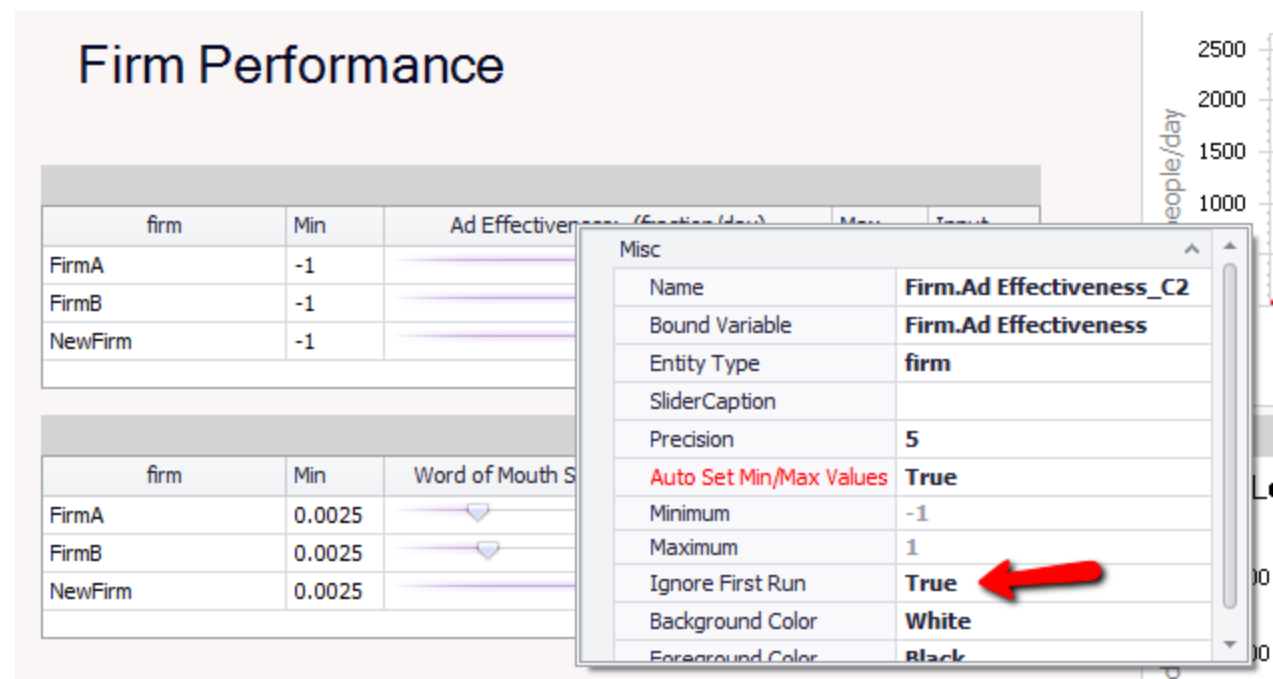
## 4c. Base vs. Policy Runs & "Ignore First Run"

- In the Run Control, use the **Duplicate** button to copy the base run, and rename the new run "policy" (the new time range should match "base")
- For the Firm Performance sliders, set "Ignore First Run" to True

Run Control

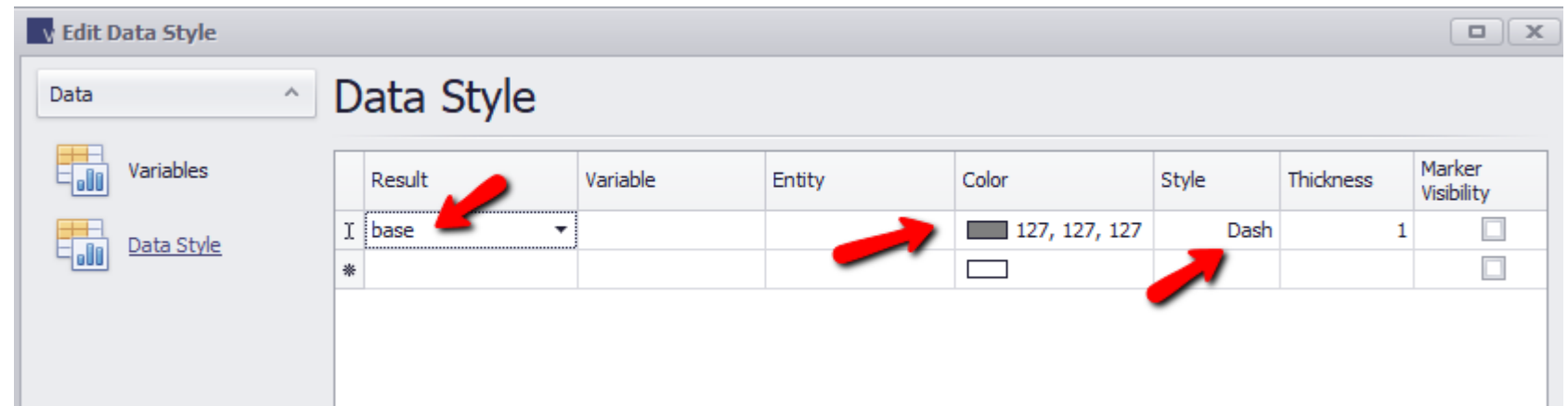
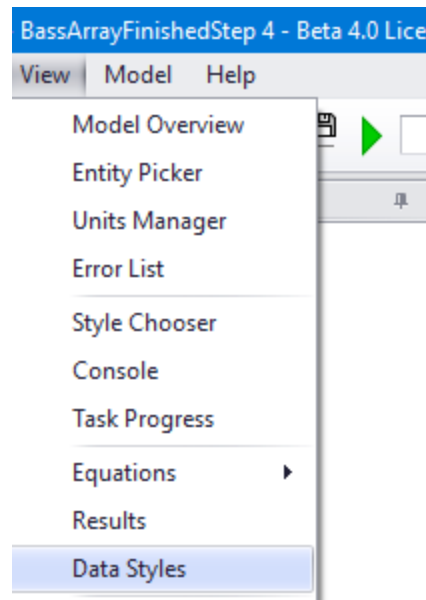


	Enabled		Entity Initialization Data	Time Series Data	Save List	Using Calendar Time	Initial Time
	<input checked="" type="checkbox"/>	base	Use active dataset	Use active dataset	None (Saves Everything)	<input checked="" type="checkbox"/>	
	<input checked="" type="checkbox"/>	policy	Use active dataset	Use active dataset	None (Saves Everything)	<input checked="" type="checkbox"/>	



## 4d. Setting Base to a low-key style

- Run the model
- Select View>Data Styles
- Switch to the Data Styles pane
- Set the line style for “base” to something unobtrusive

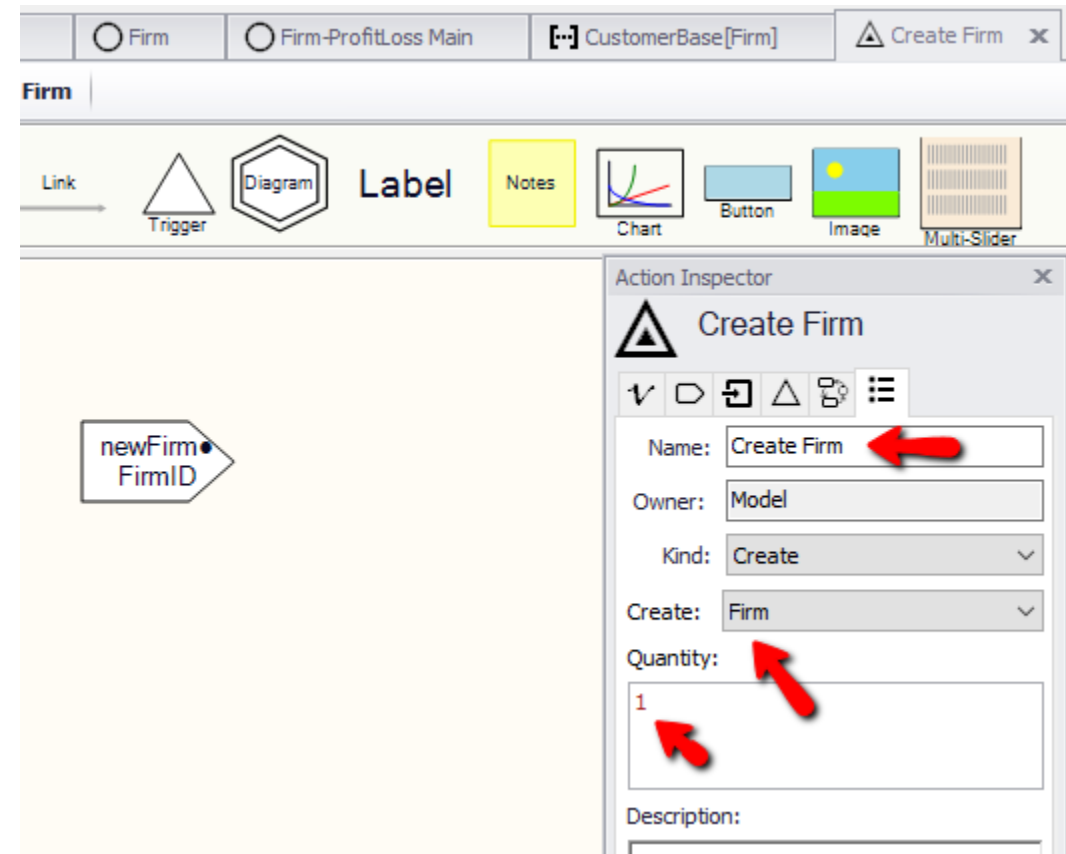
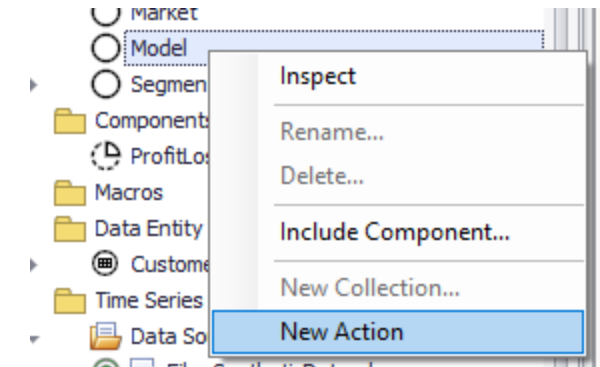


## 5. Add Firm Launch

- **Where?**
  - The Model is convenient
  - The Firm[] might be an alternative
  - Putting it in the Firm is usually undesirable, unless Firms are spawning more Firms (e.g., spinoffs)
- **Parts needed**
  - An Action that creates a Firm
  - A Trigger that fires the action

## 5a. The Action

- **Right-click the Model to make a New Action**
- **Edit the result to set the name, type and quantity**



## 5b. The Trigger

- Drag a Trigger onto the Model diagram
- Create variables to flag the event of a launch (a frequency of .001/day works since there are ~1500 days); you may want to give the base and policy runs the same random seed in the Run Control
- Set the Trigger condition to the flag value
- Set the Action to point to the action created in previous step

The screenshot displays the VENTANA software interface. At the top, the 'Model' tab is active, showing a diagram with three hexagonal nodes: 'Firm Main', 'CustomerBase Main', and 'Segment Main'. A 'Trigger' icon is being dragged onto the diagram, labeled 'B'. Below the diagram, a 'Create Firm' action is shown, labeled 'A'. The 'Market Assumptions' table is visible on the right, showing segments 'Leaders' and 'Laggers' with their respective 'Min' and 'Max' values. The 'Auxiliary - Model.Firm Enters' dialog is open, showing the 'Firm Enters' variable and the formula `RandomBernoulli(Firm Entry Frequency*Time Step)`, labeled 'B'. The 'Trigger - Model.Create Firm' dialog is also open, showing the 'Create Firm' action, labeled 'C'. The 'Trigger If' condition is set to `Firm Enters>0`, labeled 'C'. The 'Action to Invoke' is set to 'Create Firm', labeled 'D'.

segment	Min	Contact Rate: (1/day)	Max	Ir
Leaders	1		20	6
Laggers	1		20	2

Auxiliary - Model.Firm Enters

Name: Firm Enters

Owner: Model

Units: dmnl

☐ Constant

`RandomBernoulli(Firm Entry Frequency*Time Step)`

Trigger - Model.Create Firm

Name: Create Firm

Owner: Model

Seq. No.: 1

When: Period Start

Trigger If:

`Firm Enters>0`

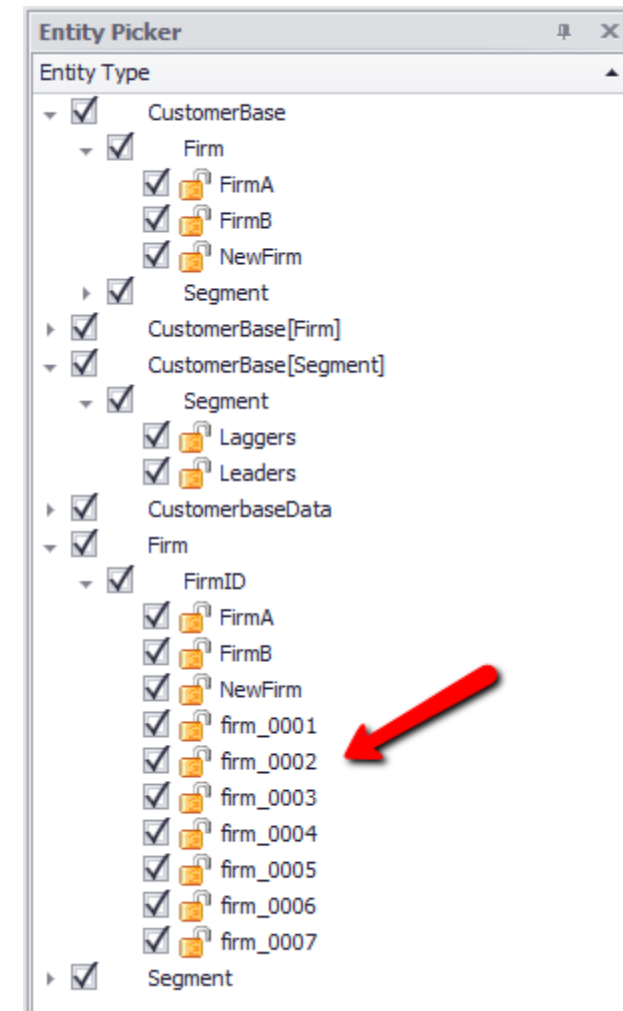
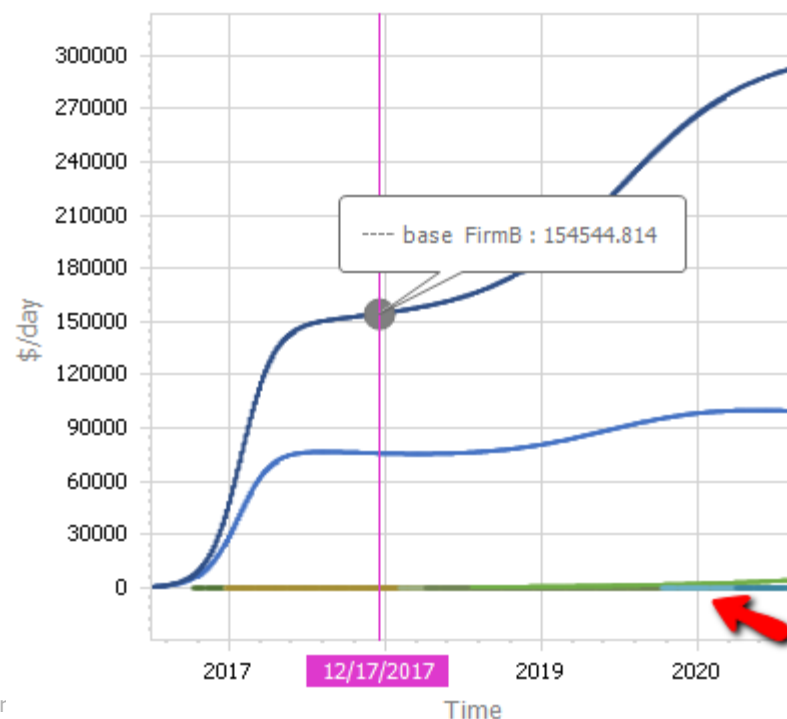
Action to Invoke:

Create Firm

## 5c. Viewing Firms

- The new firms are hard to spot on charts, because they have no Customers
- They're easier to find in the Entity Picker or data tables

ProfitLoss Operating Profit



## 6. Why no customers? Firm Segment Entry

- **Newly-created firms aren't active in any market Segments, i.e. they have no CustomerBase entities associated with them.**
- **To fix this:**
  - A Firm can check the CustomerBase[Firm] subcollection to see whether it's active in any segments
  - If none, the Firm can run a Process List action to pick a Segment to enter.
  - The Firm messages the Segment to add a new CustomerBase entity with matching Firm and Segment attributes.



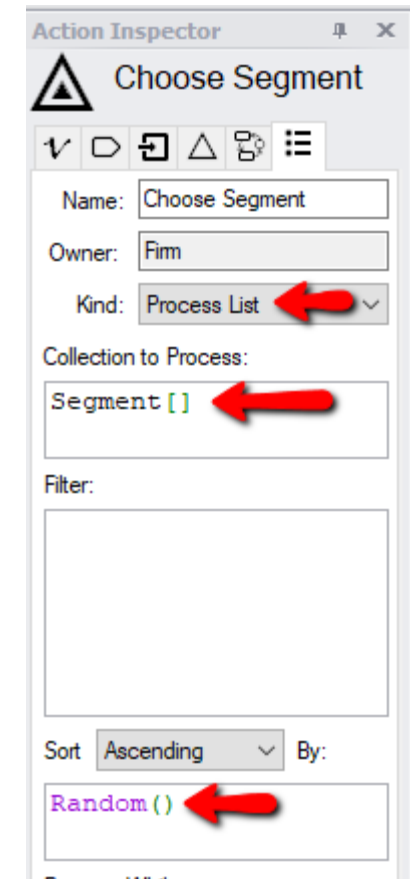
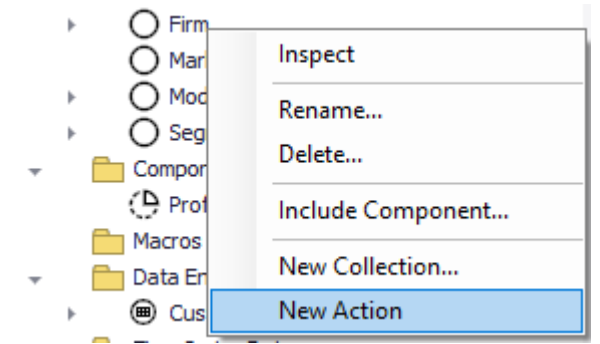
## 6a. The Firm entry trigger

The screenshot shows the Ventana software interface. At the top, there are tabs for 'Firm', 'Firm-ProfitLoss Main', and 'CustomerBase[Firm]'. Below the tabs is a toolbar with icons for 'Trigger', 'Diagram', 'Label', 'Notes', 'Chart', 'Button', 'Image', and 'Multi-Slider'. In the center, there is a diagram with a triangle and a red 'X' icon, labeled 'Choose Segment'. A green arrow points from the 'Collection Inspector' to the 'Count' property of the 'Sum Customers' collection. The 'Collection Inspector' shows the 'CustomerBase[Firm]' collection with the 'Count' property selected. The 'Trigger - Firm.Choose Segment' dialog is open, showing the trigger name 'Choose Segment', owner 'Firm', sequence number '1', and trigger condition 'CustomerBase by Firm.Count = 0'. The 'Action to Invoke' field is empty.

- Add a Trigger to the Firm
- Drag **CustomerBase[Firm].count** onto the diagram
- Set the trigger to fire when count=0
- Leave the action blank (come back and fill it in after we create the action)

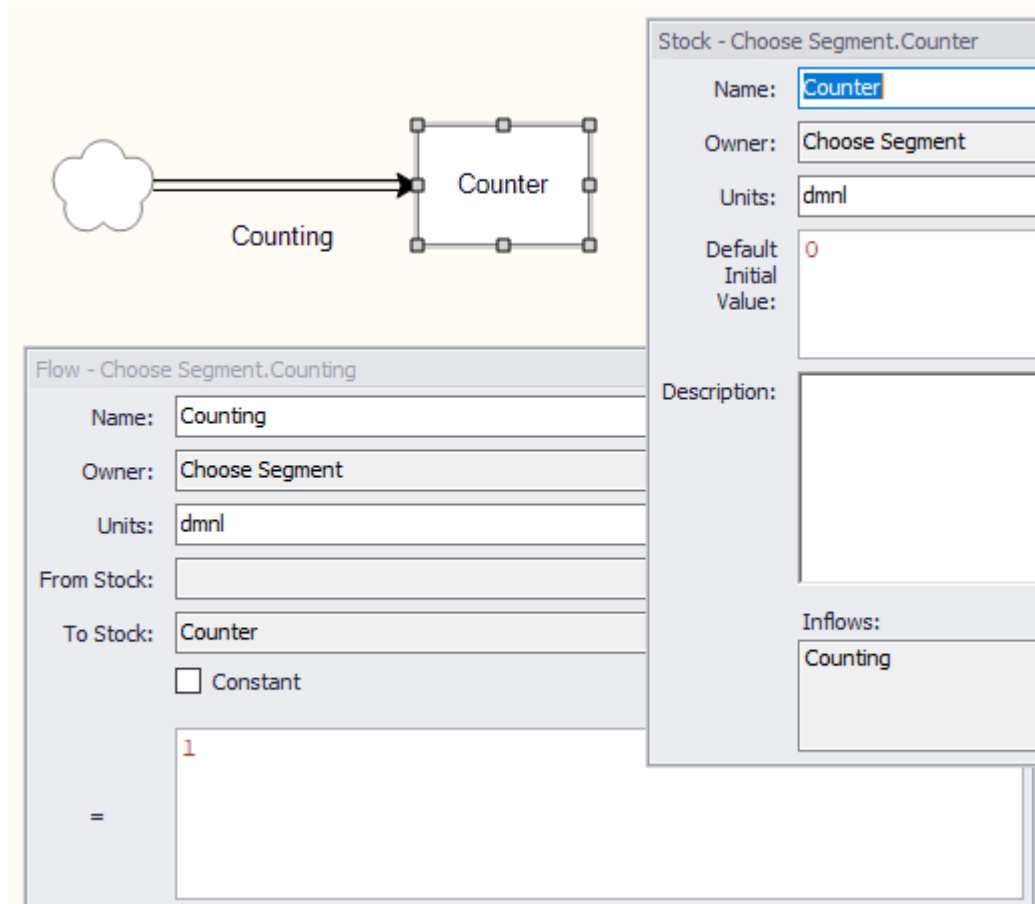
## 6b. The Process List action

- Create a new action in the Firm
- Set the Kind, Collection to Process and Sort



## 6c. Limiting entry to 1 segment

- Create a Counter stock and flow in the action
- Notice that, because an action is a discrete event, the stock and flow have the same units
- This stock exists only in the action, so it will count the number of iterations of the Process List loop



- Set the Process While field to Counter=0
- Create a trigger for Create CustomerBase (we'll fill it in later)

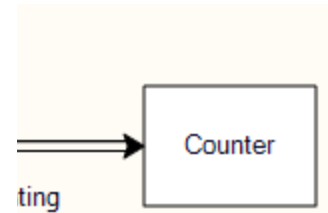
Filter:

Sort: Ascending By:

Random()

Process While:

Counter=0

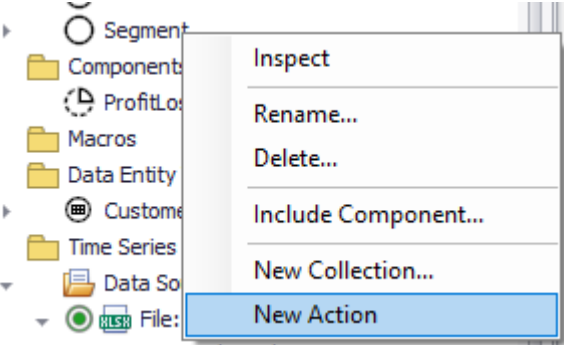


## 6d. Creating the CustomerBase

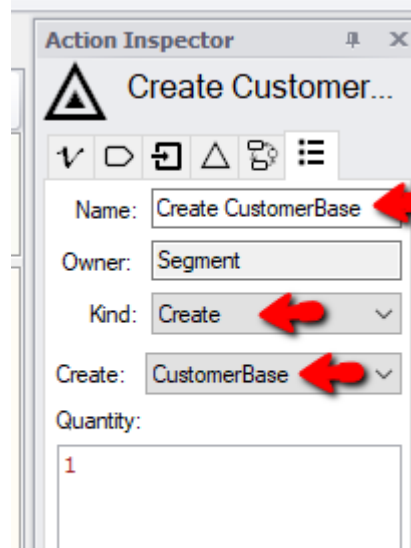
- A. Create a new action in the Segment**
- B. Name it, and set the Kind and Create type**
- C. Switch to the References tab and edit the Invoker**
- D. Change the type to Firm**

D

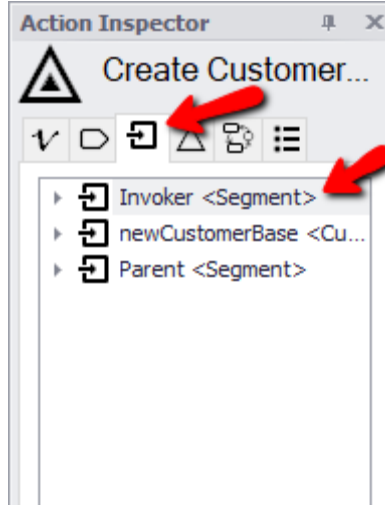
**A**



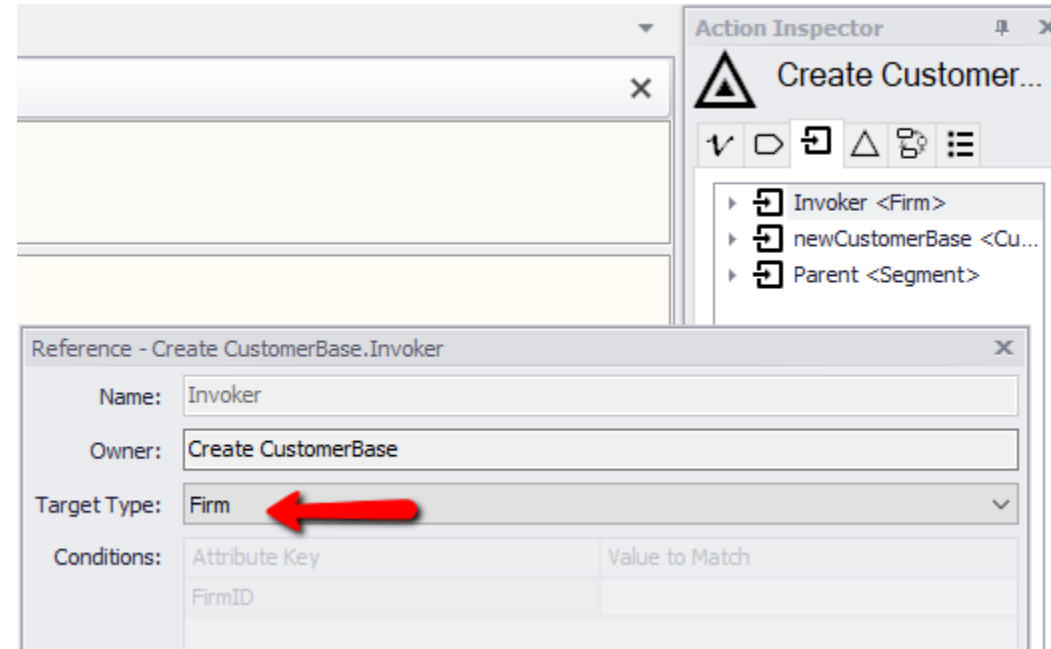
**B**



**C**



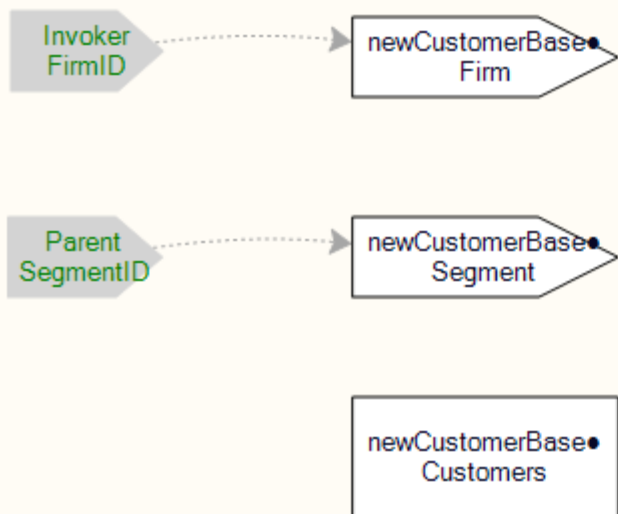
**D**



Copyright © 2018 Ventana Systems, Inc. Ventana is a registered trademark of Ventana Systems, Inc.

## 6e. Define the new CustomerBase's attributes

- You can drag the invoker and parent ids from the inspector, or type them and use the right-click-draw-arrow method to add them to the diagram



Attribute - Create CustomerBase.newCustomerBase.Firm

Name: newCustomerBase.Firm

Owner: Create CustomerBase

Refers to: Firm

☒ Key

New Value: Invoker.FirmID

Attribute - Create CustomerBase.newCustomerBase.Segment

Name: newCustomerBase.Segment

Owner: Create CustomerBase

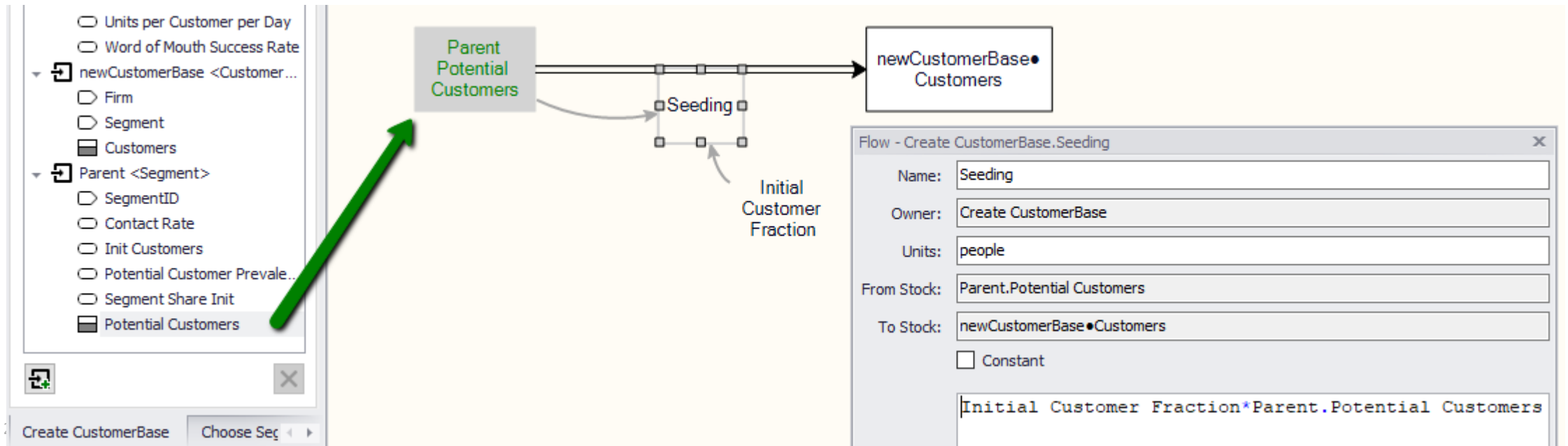
Refers to: Segment

☒ Key

New Value: Parent.SegmentID

## 6f. Define the initial Customers stock

- This is a little tricky – if we add a nonzero numerical value, we're creating new people
- If we use 0, there's no word of mouth for the new company
- Instead, we can seed the market with a small fraction of the Potential Customer stock
- We can do this by creating a cross-entity flow:



## 6g. Finishing up

- We're almost done – we just need to define the triggers we left empty
- It's easy to find them in the error list



CustomerBase  
by Firm  
Count

Trigger - Firm.Choose Segment

Name: Choose Segment

Owner: Firm

Seq. No.: 1

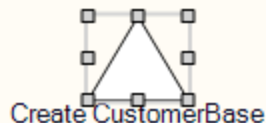
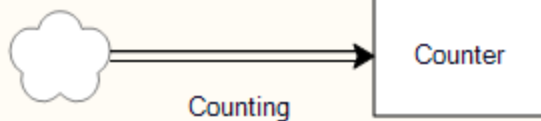
When: Period Start

Trigger If:

CustomerBase by Firm.Count = 0

Action to Invoke:

Choose Segment



Error List	
Source	Description
Trigger "Choose Segment"	Missing Action to invoke
Trigger "Create CustomerBase"	Missing Action to invoke

Trigger - Choose Segment.Create CustomerBase

Name: Create CustomerBase

Owner: Choose Segment

Seq. No.: 1

When: Never

Trigger If:

true

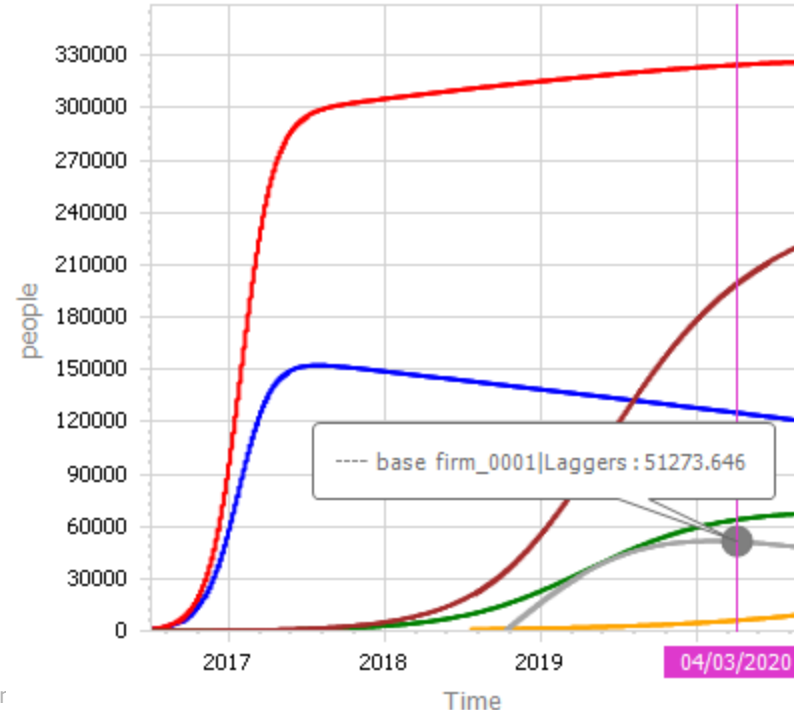
Action to Invoke:

refSegment.Create CustomerBase

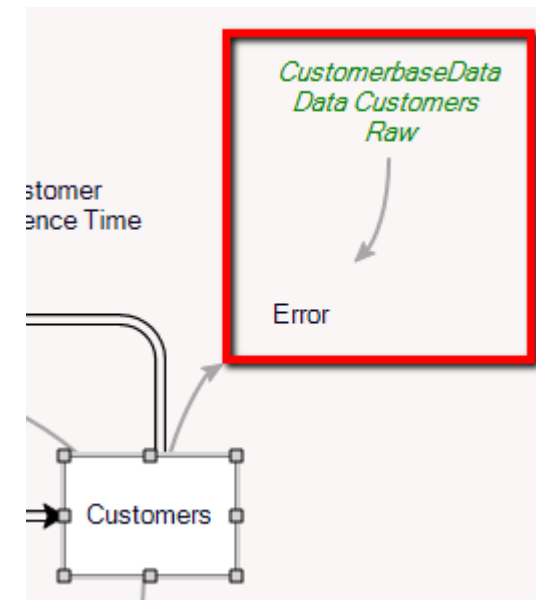
Done!

- **New firms now enter the market with customers and profits**
- **You'll see some warnings due to lack of data for new firms – ignore, or delete the computation**

Customers



Error List	
Source	Description
Runtime	[831] 10/10/2018 - Null reference customerbasedata in customerbase[firm_0001 Laggers] using "firm_0001 Laggers"
Runtime	[831] 10/10/2018 - Null reference customerbasedata in customerbase[firm_0001 Laggers] using "firm_0001 Laggers"







**EXTRA**

## Attributes & References SupplyChain8

- **SupplyChainID identifies each level**
- **Customer and Supplier attributes identify each level's partners in the chain**
- **Special cases:**
  - The factory doesn't have a supplier
  - The Retailer has exogenous customer orders
- **Modify the chain = add more data**

## What if we wanted many-to-many supplier relationships?

- **We’d need another Entity Type:**
  - Supply chain levels to manage stock, place orders, etc.
  - Transactions to relate one level to another, keep track of orders and shipments, etc.

SupplyChainLevel ID	Inventory			
Retail1	#			
Factory1	#			
Retail2	#			
Retail3	TransactionID	FromLevel	ToLevel	Quantity
...	Trans1	Factory1	Retail2	#
	Trans2	Factory1	Retail3	#
	...			

## Why do we need another entitytype for the relationship?

- **There are two “things” at work:**
  - The supply chain level: a self-contained firm, with inventory, staff, financials, etc.
  - The relationship between supply chain levels, with orders and shipments from a level to a level
- **In arrayed SD, you would notice that the firms have [level] detail, while the relationships have [fromLevel, toLevel] detail.**
- **In a database, you would link a firm table to a relationship table**