## Contents

Introduction to the tutorial ................................................. 1
Using the mouse ............................................................... 2
   Centering and zooming .................................................. 2

Section 1: Getting started
Model description ........................................................... 3
Opening AutoMod ............................................................. 4
   Creating a new model .................................................... 4
   Creating new systems within the model .............................. 5
Exporting your model ...................................................... 5
Review ................................................................. 6

Section 2: Building a model
Model description ........................................................... 7
Drawing a conveyor system ............................................... 7
   Drawing to scale ......................................................... 8
   Placing stations .......................................................... 9
Opening the process system ............................................ 10
   Defining processes ..................................................... 11
   Defining loads .......................................................... 13
   Defining load graphics ................................................ 15
Arriving procedures ....................................................... 16
   Defining source files ................................................... 17
   Writing arriving procedures ........................................ 17
Running the model .......................................................... 19
   Editing the model ....................................................... 19
Closing the model .......................................................... 20
Review ................................................................. 20

Section 3: Changing views
Opening your model ....................................................... 21
Rotating the picture ......................................................... 21
   Expanding the drawing grid .......................................... 23
Changing the spacing of the drawing grid .......................... 24
Turning text and graphics on and off ............................... 25
Using keyboard commands ............................................. 26
   Saving the configuration of windows and views ................. 26
Review ................................................................. 27

Section 4: Adding queues and resources
Model description ........................................................... 28
Changing the load creation rate ....................................... 28
Defining resources .......................................................... 29
   Placing resource graphics ........................................... 30
Defining queues ............................................................ 31
   Placing queue graphics ............................................... 32
Editing process arriving procedures ................................ 33
Changing the animation step .......................................... 35
Review ................................................................. 36

Section 5: Completing the conveyor system
Model description ........................................................... 37
Adding conveyor sections ............................................... 37

---

**Applied AutoMod 12.4**

**Beginning AutoMod Tutorial**
Section 6: Creating an AGV system
Introduction ................................................................. 52
  Terms ................................................................. 52
  Vehicle motion ....................................................... 52
System description ...................................................... 53
Adding an AGV system to the model ............................... 55
Adjusting the grid ..................................................... 55
Drawing paths .......................................................... 56
  Editing and deleting paths ....................................... 57
Filling paths .......................................................... 57
Defining blocks ....................................................... 60
  Placing block graphics .......................................... 60
  Hiding block graphics ........................................... 62
Review ................................................................. 63
Section 7: Defining and placing control points
Control points .......................................................... 64
  How vehicles claim control points ............................... 64
Defining control points ............................................. 64
Adding queues to the model ....................................... 65
Review ................................................................. 67
Section 8: Defining vehicles
Defining a vehicle .................................................... 68
  Defining starting locations for vehicles ....................... 69
  Defining a vehicle graphic ..................................... 70
Review ................................................................. 71
Section 9: Scheduling vehicles
Scheduling vehicles .................................................. 72
  Scheduling lists ................................................... 72
Building the system in phases ...................................... 73
Creating work lists .................................................. 73
Creating a park list ................................................... 74
Modifying the model .................................................. 75
Adding the storage queue .......................................... 75
Adding a new storage process ...................................... 76
  Defining the storage process’ arriving procedure .......... 76
Verifying the model ................................................... 77
Review ................................................................. 77
Section 10: Adding remaining processes
Model description ..................................................... 78
Adding queues .......................................................... 79
Adding resources ..................................................... 79
Creating the new processes ......................................... 80
  Defining the process arriving procedures .................... 80
Modifying the work and park lists ............................... 82
  Editing work lists .................................................. 83
  Defining work lists ................................................ 83
  Defining park lists ................................................ 84
Tutorial summary .......................................................... 84
Beginning AutoMod Tutorial

This tutorial provides basic training for the AutoMod software. The chapter is divided into sections so you can complete the tutorial in small increments. The material in the tutorial is designed to take you through the steps of creating, editing, and running a model.

Introduction to the tutorial

This tutorial provides step-by-step procedures for building a model. You will build the model in verifiable stages, adding more processes and detail in each lesson.

The model is a small system with eight processing steps. To move products (represented by loads) from process to process, you will create a conveyor system and an automated guided vehicle (AGV) system.
Using the mouse

- **Note**  This tutorial was designed for use with the standard graphics version of the AutoMod software. If you are using the VR graphics version of the software, the mouse and view controls differ from those documented in this tutorial. See chapter 3, “Using VR Graphics,” in the AutoMod User’s Guide for more information.

This guide often refers to using the main mouse button. Unless you have changed your system defaults, the main mouse button is the left mouse button.

The main mouse button has four functions: selecting items for editing, clicking buttons, using the menus, and placing graphics. Unless otherwise specified, use the main mouse button to perform the steps in the tutorial.

Centering and zooming

If you are using a three-button mouse, the middle mouse button is used for centering and zooming. If you are using a two-button mouse, the right mouse button is used for centering and zooming.

To center the display at the cursor’s position on the grid, place the cursor at the desired location and click the appropriate mouse button.

To zoom in on an area, place the mouse cursor in the AutoMod window, hold down the button indicated above for centering, drag the mouse to “fence” the area, and release the mouse button.

- **Note**  You can undo the last zoom (or other view change) with the keyboard command CONTROL+SHIFT+U. Also, you can return to the top view by pressing v.
Section 1: Getting started

In this section, you will learn how to do the following:

- Create a new model.
- Name and create systems within a model.
- Save your work.

Model description

In the next few sections, you will build a simple model that simulates product assembly in a conveyor system. Later, you will expand the model by adding vehicles to transport the assembled products to inspection, labeling, and storage areas. The model's conveyor system will look like the following picture:

In the conveyor system, products are assembled in two steps. The first step of the assembly occurs in queue Q_Assemble. After completing this step, the assembly operator places the products onto the conveyor at station sta1. Products then travel to station sta2, where a second operator completes the product's assembly on the conveyor.

The finished products then continue on the conveyor to either staout1 or staout2, where they transfer to automated guided vehicles (AGVs).
Opening AutoMod

There are two different environments within the AutoMod software:

- Edit environment
- Simulation environment

The edit environment is used to create and define a model. The simulation environment is used to view and analyze the actions of a compiled model.

➤ To enter the edit environment and begin creating a model

From the Start menu, select AutoMod from the AutoMod program group. The AutoMod window opens.

Creating a new model

➤ To create a new model

1. Click File > New (open the File menu and click New).
2. Create a new directory for the model.
3. Name the model Demo1, then press ENTER.

The name appears in the title bar of the AutoMod window. The Process System palette appears, and the Selection dialog box opens.
Creating new systems within the model

There can be many different systems within each model. Systems fall into three categories: process, movement, or static.

The process system contains the model logic (think of it as the brain of your model). Processes describe the manufacturing steps necessary to make products. When you create a new model, a process system with the same name is automatically created.

In addition to a process system, you add an unlimited number of movement systems. The following types of movement systems are used in this tutorial:

- Conveyor
- AGV (path mover)

These movement systems are used to move loads from one area of your model to another. A model can contain either one movement system or a combination of several movement systems.

Finally there are static systems, which can be used to show walls or other non-moving elements of a structure. A static system enhances model graphics. This tutorial does not cover static systems.

You are now going to add a conveyor system to your model.

Creating a conveyor system

► To create a conveyor system
1. Click System > New.
   The Create A New System dialog box opens.
2. From the System Type list, select Conveyor.
3. Name the system Conv, then press ENTER.
4. Click Create.
   The Conveyor System palette opens. This palette contains the drawing tools needed to create conveyor sections and stations.

Exporting your model

At this point, you will export the model to save your changes. Exporting does the following:

- Saves the model
- Creates a backup (archive) model that is upwardly compatible with newer versions of AutoMod

► Tip Exporting is equivalent to saving in other applications. Therefore, export your model frequently.

► To export a model
1. Click File > Export.
2. Click Yes to confirm that you want to export the model.
Review

You have defined a model named Demo1. This model consists of two systems: a process system called Demo1 and a conveyor system called Conv. You have exported the model.
Section 2: Building a model

In this section, you will learn how to do the following:

• Draw straight conveyor sections.
• Define products (loads).
• Define processes.
• Move products down a conveyor.
• Build and run the model.

Model description

In the first phase of the model, you will draw a conveyor section and send products from one end to the other. In the AutoMod software, loads represent a quantity of product or material.

You want loads to enter the model at the first conveyor station, sta1, and travel to the second conveyor station, sta2, at the other end of the conveyor (see the illustration below). After arriving at sta2, loads leave the system (die). You are going to create a new load at station sta1 every five seconds.

Drawing a conveyor system

Before you draw the conveyor section, zoom in on the upper left half of the grid.

► To zoom in on the upper left half of the grid
1. Using the middle mouse button (right mouse button on a two-button mouse), click and drag a box around the upper half of the grid.

Tip To undo a zoom, you can either press CONTROL+SHIFT+U to undo the last change or press v to return to the top view.

2. On the Conveyor palette, click Single Line.
The Single Line dialog box opens.

Note The default names of sections start with the base name sec, followed by a number that increments every time you place another section. If you wish to change the name of the section you are about to place, type in the name and press ENTER before placing it.
3. Select the **Orthogonal** check box so the section you are adding will be horizontal or vertical in the AutoMod window.

**Note** You will use the default attributes for conveyor speed, width, and so on.

**Drawing to scale**

It is important that you draw the conveyor section using the correct scale; the section must be 80 feet long. Use both the grid and the Measurement dialog box to help you. By default, the grid lines are five feet apart, with thicker lines every 50 feet.

**To draw the conveyor section**

1. In the AutoMod window, click the measurement icon to open the Measurement dialog box (you might need to move the Single Line dialog box to see the icon). Move the Measurement dialog box below the AutoMod window.

2. Select **Track Mouse**.

   **Tip** Selecting the **Snap** option in the Measurement dialog box causes the graphics you are placing to move in increments determined by the grid spacing. Throughout this tutorial, you should clear **Snap** before drawing or placing objects that require fine positioning. Do not clear the **Snap** option now.

3. In the AutoMod window, click once in the upper left corner of the grid where the two thicker lines intersect (refer to the illustration below).

   **Tip** If you want to delete the section you are drawing and have placed only one point, press **ESC**.

4. Move the mouse to the right. Watch the Measurement dialog box’s **Length** field to see how long the conveyor section is. When the **Length** field is around 80, use the grid to locate the 80-foot mark and click the mouse again.

   **Note** The Measurement dialog box tracks the mouse, not the section length, so the value in the dialog box may not be exactly 80. That is why you should also use the grid. If you cannot tell from the grid whether the section is the right length, refer to “Editing sections” below to change the section’s length.
Notice that the direction of travel for the conveyor is from the left to right, or first mouse click to second mouse click.

![Diagram of conveyor section]

**Tip** If you need to change the direction of a conveyor section, use the `Select` tool on the Conveyor palette to select the desired section, then click `Edit > Change Direction`.

5. Close the Measurement dialog box.

**Editing sections**

If you want to delete a conveyor section and have placed only one of the two ends, press `ESC` to delete it. If you have already placed both of the ends of the path, you can move, delete, or edit it.

**To move, delete, or edit a section**

1. On the Conveyor palette, click `Select`.
2. Click the conveyor section you want to move, delete, or edit.
3. From the `Edit` menu, select `Move`, `Delete`, or `Edit`.
4. Drag or redraw the section, or edit the section’s ending value to adjust its length.

**Placing stations**

Now that you have drawn the first conveyor section, you are ready to place the first two conveyor stations. Stations are locations at which loads can enter and leave a conveyor section, and stations are also where work is performed. In this model, loads get on the conveyor at station `sta1` and are unloaded at station `sta2` (refer to the illustration below).
To place conveyor stations
1. On the Conveyor palette, click Station. The Station dialog box opens.

![Station Dialog Box](image)

**Note** The default names of stations start with the base name `sta`, followed by a number that increments every time you place another station. If you want to change the name or attributes of the station you are about to place, do so before placing it. For now, use the default names and attributes for the stations.

2. In the AutoMod window, drag the first station into position at the beginning of the conveyor section (the graphic for the station appears when you click the mouse button and is placed when you release the mouse button).

You have now placed `sta1`.

3. Drag the second station into position at the end of the conveyor section.

You have now placed `sta2`.

The AutoMod window contains a single section, as shown below:

```
sec1
  sta1 > sta2
```

**Moving stations**
If you need to move a station you have already placed, use the following procedure.

**To move a station you have already placed**
1. On the Conveyor palette, click Select.
2. Click the station you want to move.
3. Click **Edit > Move**.
4. Click the station and drag it with the mouse to the desired location.
5. Click OK.

**Tip** Remember to export your work frequently.

---

**Opening the process system**
Now that you have created a simple conveyor system in your model, you need to write the logic that moves loads on the conveyor.
To write the logic that moves loads on the conveyor
1. Open the process system.
2. Click System > Open.
   The Open a System dialog box opens.
3. Select Demo1 in the list, then click Open.
   You have moved from the conveyor system (Conv) to the process system (Demo1).
   The Process System palette appears.

Defining processes
The process system allows you to define your manufacturing procedures, called processes. Processes are used to direct and control load movement in a model.
In this section you will define two processes: P_EnterAssembly, which places loads on the beginning of the conveyor, and P_CompleteAssembly, which removes loads from the conveyor.

**To define the processes**

1. On the Process System palette, click **Process**.
   
The Process dialog box opens.

2. Click **New**.
   
The Define A Process dialog box opens.

3. Name the process `P_EnterAssembly`, then press **ENTER**. (Hyphens and spaces are not allowed in names, so use the underscore ( _ ) instead.)

   **Note** You can name a process using any alphanumeric characters and underscores.

4. Click **OK/New**.
   
The dialog box is now ready for you to define the second process.

5. Name the process `P_CompleteAssembly`, then press **ENTER**.

6. Click **OK**.
The two processes you have defined appear in the list.

![Image of process system palette showing processes CompleteAssembly and EnterAssembly]

### Defining loads

Now you are ready to create the loads that are transported in the model. In the AutoMod software, you can define as many load types as necessary to distinguish between different types of products in a system. In this model, we want to process only one type of load.

When you define a load type, you can also define a load creation specification. The specification determines the number of loads of that type to create during simulation and the time between each load creation. The load creation specification also specifies which process the loads execute first. Once created, the loads are sent from process to process, carrying out process procedures.

**To define a load type**

1. On the Process System palette, click **Loads**.
   
The Loads dialog box opens:

2. Because you are defining a new type of product (load), click **New** to the right of the **Load Types** list.
The Define A Load Type dialog box opens.

3. Name the load type L_Carton, then press \textbf{ENTER}.
   The L represents load and Carton represents the load type.

\begin{itemize}
  \item \textbf{Note} Remember that names of entities (which include processes and loads) cannot have spaces or hyphens in them. For this reason, use the underscore ( _ ).
\end{itemize}

Now you need to define the creation specification for this load type.

4. Click the \textbf{New Creation} button.
   The Define a Creation Spec dialog box opens.

5. Click \textbf{First Process}.
   The Pick A First Process dialog box opens.

6. Since you want your loads to start at process \texttt{P_EnterAssembly}, select \texttt{P_EnterAssembly} in the Pick A First Process dialog box, then click \textbf{OK}.
   The process \texttt{P_EnterAssembly} appears to the right of the First Process button in the Define a Creation Spec dialog box.

   A \textit{generation limit} is the total number of load creation cycles that occur before this creation specification stops producing loads. Leave the generation limit at \textit{Infinite}, so this load type is created continuously throughout the simulation.

   The \textit{distribution} specifies the interval of time in the creation cycle. Leave the constant distribution with a mean of five seconds. This distribution creates a load every five seconds and sends it to process \texttt{P_EnterAssembly}.

7. Click \textbf{OK}.
The creation information now appears in the **Distribution** list.

![Distribution List](image)

8. Click **OK**.

   L_Carton now appears in the **Load Types** list.

**Defining load graphics**

Now you are ready to define the size of the loads in the model. The default load size is 1x1x1 feet. You will define a slightly larger load.

1. Select L_Carton, then click the **Edit** button. The Edit A LoadType dialog box opens.
2. Click the Edit Graphic button. The Edit LoadTypes Graphics dialog box opens.
3. Click **Place**.
4. In the AutoMod window, click above station sta1 to place the load’s graphic; a small green box appears, representing a load of type L_Carton.
5. In the Edit LoadTypes Graphics dialog box, select the **Scale All** check box (see the following illustration). This allows you to change all three dimensions of the load at once.
6. Change the Scale value to 2, then press ENTER.

Notice that the X, Y, and Z scale boxes all change to 2, making the load scale 2x2x2 feet and doubling the size of the load graphic in the AutoMod window. Now you will change the dimensions of the load to 2x3x2 feet.

7. Clear the Scale All check box.
   This allows you change one load dimension at a time.

8. To increase the Y dimension of the load, click the Y Scale button (the second button under Scale) to scale the Y dimension.

9. Change the Scale value to 3, then press ENTER.
    The Y dimension of the load increases to three.

You will now hide the load’s picture until the model is running.

> **To hide the load’s picture until the model is running**
1. Click Hide, then click Yes to confirm.
   The load graphic no longer appears in the AutoMod window.
2. Click Done, then OK.

---

**Arriving procedures**

You have defined loads for the model and sent them to their first process, P_EnterAssembly. Now you need to write the instructions for P_EnterAssembly, telling the loads what to do when they get there.

You write instructions for processes in arriving procedures. An arriving procedure is a load’s instructions for a process, including which resources to use and how long each action takes. Loads execute arriving procedures when they enter a process.
Arriving procedures are written in a text file called a source file. A source file can contain one or more arriving procedures. In this model, you will write all procedures in the same source file, which makes editing the procedures easy.

Defining source files

- **To define the source file for your model**
  1. On the Process System palette, click **Source Files**. The Source Files dialog box opens.
  2. Click **New**. Name the file **mycode.m**.
  3. Click **OK**.
  4. Select **mycode.m** in the list, then click **Edit**. A text editor opens a blank file.

**Note**  Source file names must end with a .m extension.

Writing arriving procedures

You have defined two processes in the model: **P_EnterAssembly** and **P_CompleteAssembly**. When a load is sent to a process, it immediately executes the process’ arriving procedure. Because new loads are sent to process **P_EnterAssembly**, this process’ arriving procedure provides the first instructions for load activity during a simulation.

You will write the arriving procedure for **P_EnterAssembly** to cause loads to get on the conveyor at station **sta1**. The arriving procedure will then send loads to the **P_CompleteAssembly** process.

The arriving procedure for **P_CompleteAssembly** will cause loads to travel from station **sta1** to **sta2** on the conveyor. The procedure will then cause loads to leave the simulation (die).
Each arriving procedure must begin with the line `begin <processname>` arriving procedure and must end with the line `end`. Between these lines, you can type the instructions for the loads that enter the process. The procedures for this section use three actions: `move`, `travel`, and `send`.

### Action Description

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>move</code></td>
<td>The <code>move</code> action places the load in a specific place, such as in a queue or at a station on a conveyor system. You can use the <code>move</code> action to get loads into a movement system, and to change loads from one movement system to another (you will use this extensively when you add the AGV system to the model later in the tutorial).</td>
</tr>
<tr>
<td><code>travel</code></td>
<td>Once a load is in a movement system, you can use the <code>travel</code> action to cause the load to use the system to travel from one location to another. The load automatically finds the shortest path between locations in a movement system.</td>
</tr>
<tr>
<td><code>send</code></td>
<td>The <code>send</code> action sends a load to another process. It is usually the last action in an arriving procedure.</td>
</tr>
</tbody>
</table>

**To write the arriving procedure**

1. Type the following procedures in the source file (the bold type is for reference only; you do not need to make the code bold in the editor). AutoMod syntax is case-sensitive and must be typed exactly as shown.

```autoMod
begin P_EnterAssembly arriving procedure
  move into Conv.stal /* Get on the conveyor at station stal */
  send to P_CompleteAssembly /* Leave the current process */
end

begin P_CompleteAssembly arriving procedure
  travel to Conv.sta2 /* Travel from current location to station sta2 */
  send to die /* Leave the simulation */
end
```

The first action in the `P_EnterAssembly` procedure causes each load to move into station `stal` in the conveyor system. The second action sends the load to the next process, `P_CompleteAssembly`. The load then starts performing the arriving procedure of `P_CompleteAssembly`. `P_CompleteAssembly`’s arriving procedure causes the load to travel on the conveyor from the first station to the second (`sta2`). After arriving at its destination, the load leaves the model.

**Tip** Any remarks placed between the symbols `/*` and `*/` in the code are comments and do not affect the simulation. Comments are notes that help you understand what the code is doing.

**Note** Throughout this tutorial, you will expand these arriving procedures to simulate more complex processes.

2. Click **File > Save**, then **File > Exit**.
You have now defined a source file containing the arriving procedures.

- **Note**  If you get any error messages when quitting a source file, they are caused by typing errors. To fix the typing errors, click the **Return to Edit** button in the Error Correction dialog box to return to fix the mistakes on the lines specified. When you are finished, save your changes and exit the file editor.

---

### Running the model

At this point, you have completed everything necessary to export and run the model.

- **To export and run the model**
  1. Export the model.
  2. Click **Run > Run Model**, then confirm by selecting **Yes**.

     It takes a few moments for the model to compile and link, then the simulation environment opens.

     The simulation environment has three dialog boxes:

     | Dialog Box | Function |
     |------------|----------|
     | Status     | Shows whether the model is paused or running, and what the current simulation time is. |
     | Message    | Shows messages and errors as the model runs. |
     | Simulation | Shows the animation of the model. |

  3. Zoom in on the conveyor section by using the middle mouse button (right mouse button on a two-button mouse) to band the desired area.
  4. To start the simulation, click **Control > Continue**.

     Watch the model for a few minutes to see loads traveling on the conveyor from **sta1** to **sta2**.

- **Tip**
  - You can toggle between pausing or continuing the simulation by pressing **p** (lowercase).
  - By default, entities are displayed in wireframe (only the outlines of entities appear in the Simulation dialog box). You can toggle between wireframe and solid views by pressing **w** (lowercase).

---

### Editing the model

- **To return to the edit environment**
  1. Press **p** to pause the model.
  2. Click **File > Run Model Editor**.

     You are now back in the edit environment.
**Closing the model**

As you work through the tutorial, you may want to close your model at the end of a section and resume building your model later. Therefore, exit AutoMod now for practice.

- **To close the model**
  
  Click **File > Exit**.

  You will open the model at the beginning of the next section.

**Review**

You have drawn your conveyor system and placed two stations. You have created all of the logic necessary to move loads from the process `P_EnterAssembly` to the process `P_CompleteAssembly`, that is, from the beginning of the conveyor to the end of the conveyor.

You have defined your loads.

You have built and run the model.
Section 3: Changing views

In this section, you will learn how to do the following:

• Open the model.
• Rotate the picture.
• Expand the drawing grid.
• Change the spacing of the drawing grid.
• Turn text and graphics on and off.
• Use keyboard commands and the Help menu.
• Save the configuration of dialog boxes and views.

The following material shows how to control the view of the model. For additional information on the basic mechanics of screen manipulation, read chapter 2, “Using the Interface,” in the AutoMod User’s Guide.

Opening your model

▶ To open your model
1. Start the AutoMod software.
2. Click File > Open.
   A navigation dialog box opens.
3. Navigate to the model’s .dir directory.

■ Note  When you open a model, the grid is automatically resized to fit the entities currently placed in the model. In this section, you will learn how to expand the drawing grid.

Rotating the picture

The View Control dialog box allows you to change the view of your model. You can view the model from a variety of positions. This dialog box also lets you adjust the size of the grid.
To open the View Control dialog box

1. In the AutoMod window, click the view control icon.

The View Control dialog box opens:

The View Control dialog box contains the following options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotate</td>
<td>Rotating moves the model around the different axes.</td>
</tr>
<tr>
<td>Translate</td>
<td>Translating moves the model along the different axes.</td>
</tr>
<tr>
<td>Scale</td>
<td>Scaling the model makes it bigger or smaller on the screen.</td>
</tr>
<tr>
<td>Child Windows on Top</td>
<td>This prevents the palette and dialog boxes from getting lost behind other windows.</td>
</tr>
<tr>
<td>Perspective</td>
<td>Perspective is a method of drawing in which all lines lead to a vanishing point; this is how we naturally view the world. Orthogonal is a method of drawing in which all lines are at right angles to each other; this is a projection of a 2-D drawing into a 3-D view.</td>
</tr>
<tr>
<td>Solid</td>
<td>Solid displays model entities as solid objects. When Solid is off, only the wireframe outlines of shapes are displayed.</td>
</tr>
<tr>
<td>Friction</td>
<td>Friction controls continuous movement, including translation, rotation, and scaling. When Friction is on, the graphics move only when you indicate. When Friction is off, any movement command causes the model to move continuously until you explicitly stop it, either by toggling friction on again or by pressing the space bar.</td>
</tr>
<tr>
<td>Axis Display</td>
<td>Axis Display causes a triad (X,Y,Z) to be displayed at the model’s origin.</td>
</tr>
<tr>
<td>Screen</td>
<td>Screen controls the model movement. The Screen check box controls the movement of the model in relation to screen coordinates or world coordinates. The world coordinates are in relation to the model’s origin (Axis Display). The screen coordinates are in relation to the current screen view. The axes of the following views originate from the world coordinates.</td>
</tr>
<tr>
<td>Top</td>
<td>View the entity from the positive Z axis.</td>
</tr>
</tbody>
</table>
Take a few minutes to familiarize yourself with the view control.

**Tip** If at any time you lose sight of your model graphics (perhaps by accidentally pressing the left mouse button), press v to restore the top view.

2. Click **Top** to return to the default view.
3. Select **Child Windows on Top**.

## Expanding the drawing grid

When you open a model, the drawing grid is automatically sized to fit the entities in the model. If you are adding entities, you can expand the drawing grid to help you align and position the new entities. For this tutorial, you need to expand the grid so you can draw the rest of the conveyor system and the AGV system.

**Note** If you close and reopen the model at any time during this tutorial, refer to this section to reset the size of the drawing grid.

**To expand the drawing grid**

1. In the View Control dialog box, click **Set Limits**.

   The Set Limits dialog box appears.
2. Change the drawing limits for each dimension to the values shown below:

![Set Limits dialog box]

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>X: Left</td>
<td>-135.0000</td>
<td>10.0000</td>
</tr>
<tr>
<td>Y: Back</td>
<td>-90.0000</td>
<td>80.0000</td>
</tr>
<tr>
<td>Z: Bottom</td>
<td>0.0000</td>
<td>10.0000</td>
</tr>
</tbody>
</table>

3. Click **OK** to close the Set Limits dialog box.
4. Close the View Control dialog box.

You are now ready to further customize the drawing grid using the grid control.

---

### Changing the spacing of the drawing grid

The Grid Control dialog box allows you to change the position of the drawing grid and the distance between grid lines. All entities are drawn or placed on the plane of the current grid.

**To open the Grid Control dialog box**

1. In the AutoMod window, click the grid control icon. The Grid Control dialog box opens.

![Grid Control dialog box]

#### Option Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Origin/Orientation</td>
<td><strong>Origin</strong> and <strong>Orientation</strong> allow you to adjust the position of the grid. <strong>Origin</strong> should be thought of as translational movement of the grid, and <strong>Orientation</strong> as rotational grid movement.</td>
</tr>
<tr>
<td>Align Grid to ...</td>
<td>This allows you to move the grid to predefined positions: Top, Bottom, Front, Back, Left Side or Right Side. Changing the grid alignment allows you to place graphics on different axes.</td>
</tr>
</tbody>
</table>
Take a few minutes to familiarize yourself with the grid control.

2. When you are finished, type zeros in all **Origin** and **Orientation** values of the Grid Control dialog box.
3. Set the **Grid Spacing** and line values to the defaults (shown in the illustration above).
4. Close the Grid Control dialog box.

### Turning text and graphics on and off

The Display dialog box allows you to hide or display each system or system entity. It also allows you to change colors of system entities, such as conveyor sections, queues, resources, and so on.

1. **To configure display options**

   Click **Model > Display**.

   The Display dialog box opens.
2. To change colors of system entities, select the color from the list.
3. To see the entity change color in the AutoMod window, click Apply.
4. You can also make entities visible or invisible by changing the Visible check box. For example, if you do not want the text in your process system displayed, clear the Visible check box next to Text.
   Take a few minutes to familiarize yourself with the Display features.
5. When you are finished, reset all system entities to visible.
6. Click OK.

Using keyboard commands

Many of the viewing changes you have just made can also be accomplished using the keyboard. There is an online Help topic that you can use as a quick reference for keyboard shortcuts.

► To view the keyboard commands
1. In the AutoMod window, type h.
   The AutoMod Help displays a list of keyboard commands.
   Take a few minutes to get familiar with some of the basic commands, such as the following:
   h = Help menu display
   X = rotate X axis clockwise
   x = rotate X axis counterclockwise
   Y = rotate Y axis clockwise
   y = rotate Y axis counterclockwise
   Z = rotate Z axis clockwise
   z = rotate Z axis counterclockwise
   P = perspective
   G = grid toggle
   f = friction toggle

Saving the configuration of windows and views

To save the current size and position of your windows, save the startup configuration. Then every time you open this model, the view will appear as you saved it.

For example, you can organize the placement of the windows on your screen in any manner you desire: resize the AutoMod window, reposition the palette, change system display preferences, and set options from the View Control.

► To save the Startup Configuration
   Once your preferences are set, click File > Save Startup Config.
   A new file called .am2rc is created in your directory; it contains the instructions for setting up the model. When you open any model within the directory containing the .am2rc file, the screen appears as you previously set it.
Review

You have now learned how to change the orientation of your picture.
You have expanded the drawing grid and learned how to change its spacing.
You have learned how to adjust the appearance and size of windows, text, and graphics. You have also learned how to save the configuration of these items in the startup configuration.
You have learned keyboard commands.
Section 4: Adding queues and resources

In this section, you will learn how to do the following:
• Change the load creation rate.
• Define resources.
• Define queues.
• Edit process arriving procedures.
• Change the animation step to make the simulation run faster or slower.

Model description

In this section, you will add operators and time delays to your model to simulate the assembly of loads in the conveyor system.

You will make the following changes to the model:
1. Change the load creation rate from one load every five seconds to one load every two minutes.
2. Add two resources to assemble the loads in the system.
3. Add a queue $Q_{Assemble}$ to hold loads at the beginning of the conveyor.

Changing the load creation rate

Because you will be adding assembly delays to the model, you need to slow the load creation rate from one every five seconds to one every two minutes.

► To slow the load creation rate
   The Loads dialog box opens.

2. From the Load Types list, select $L_{Carton}$.
3. Click Edit to the right of the Load Types list.
   The Edit A LoadType dialog box opens.
4. Select the creation specification that reads:
   Constant 5 sec Infinite P_EnterAssembly 0 stream0
5. Click **Edit** in the bottom of the dialog box.
   The Define a Creation Spec dialog box opens.
6. Change the **Mean** to 2 minutes, as shown below:

   ![Define A Creation Spec dialog box](image)

   Change the mean to 2 minutes.

7. Click **OK**.
   The Define A Creation Spec dialog box closes.
8. Click **OK**.
   The Edit A LoadType dialog box closes.

---

**Defining resources**

Resources are used to represent machines, operators, tools, fixtures, and other entities that process loads. In your model, you will use two resources to represent operators. The first operator assembles loads and places them on the conveyor at station **sta1**. The second operator completes the assembly on the conveyor at station **sta2**.

- **To define resources for your model to represent operators**
  1. On the Process System palette, click **Resources**.
     The Resources dialog box opens.
  2. Click **New** to the right of the **Resources** list.
     The Define A Resource dialog box opens.
  3. Name the resource **R_Operator1**, then press **ENTER**.
     The default capacity represents how many loads a resource can work on at one time. In this model, the resource **R_Operator1** can assemble only one load at a time, so leave the **Default Capacity** set to **1**.
  4. Click **OK, New**.
     You are now ready to define the second resource.
  5. Name the resource **R_Operator2**, then press **ENTER**.
  6. Click **OK**.

You have just defined two resources: **R_Operator1** and **R_Operator2**. You will instruct loads to claim and use these resources when you edit the process arriving procedures later in this section.
Placing resource graphics

You are going to import a human graphic to make the representation of the operators more realistic. You will place the operators at either end of the conveyor (see the illustration below).

To import and use a human graphic

1. From the Resources list, select R_Operator1.
2. Click Edit.

   The Edit A Resource dialog box opens.

   4. Zoom in on the conveyor.
   5. From the Shape Definition list in the Edit Resource Graphics dialog box, select Import.
   6. Navigate to the demos/graphics/cell directory in the software installation directory (AutoMod by default).
   7. Double-click the file man.cel.

   Tip To position the operator without snapping to nearby grid lines, open the Measurement dialog box, then clear the Snap option.

8. Click Place, then drag the operator into position at the beginning of the conveyor (see the illustration on the previous page).
9. Click Done, then OK.
Placing R_Operator2

Tip If the right end of the conveyor is out of view, click the middle mouse button (use Control + the left button on a two-button mouse) in the right side of the dialog box. This centers the screen where you clicked the mouse, moving the right side of the picture into view. This is an alternative to pressing v and rezooming the view.

To import and rotate a human graphic
1. Repeat steps 1 through 8 above to place the second operator (complete only steps 1 through 8; do not close the Edit Resource Graphics dialog box).
   Once the operator graphic is placed, you need to rotate it so it is facing the conveyor.
2. Click the Rotate Z button (the button’s current value is 0).
3. Type 180, then press ENTER.
   The operator rotates to face the conveyor.
4. Click Done, then OK.

Defining queues

Queues represent a temporary holding area, such as a loading dock or a workbench. In order for a load to get off a conveyor section or vehicle, it must be moved into a queue or a movement system (or sent to die).

You will now define a queue Q_Assemble at the beginning of the conveyor where the first step of each load’s assembly takes place.

To define a queue at the beginning of the conveyor
1. On the Process System palette, click Queues.
   The Queues dialog box opens.
2. Click New.
The Define a Queue dialog box opens.

3. Name the queue Q_Assemble, then press ENTER.
4. Change the Default Capacity to i, then press ENTER.
   The word Infinite appears, meaning that there is no limit to the number of loads that can be in the queue at one time.
5. Click OK.
   The Queues dialog box opens, with the newly defined queue listed.

Placing queue graphics

You need to place the assembly queue graphically in order for it to be visible during a simulation.

- To place the assembly queue graphically
  1. Select Q_Assemble in the Queues list.
  2. Click Edit.
     The Edit A Queue dialog box opens.
  3. Click Edit Graphic.
     The Edit Queue Graphics dialog box opens.
4. Click **Place**.

5. Drag the queue into position above the resource (see the following illustration).

6. Click **Done**.

After you have finished placing Q_Assemble, your model appears as shown below:

7. Export the model.

---

**Editing process arriving procedures**

To instruct loads to use the resources and queue that you have defined, you must edit the arriving procedures for processes P_EnterAssembly and P_CompleteAssembly.

Loads executing the **P_EnterAssembly** process must first move into the queue Q_Assemble and wait while being assembled. The first assembly step requires an amount of time that is uniformly distributed between 85 and 115 seconds. After the step is complete, R_operator1 takes 10 seconds to remove the load from the queue and place it on the conveyor.

Because the capacity of R_Operator1 is one, the operator can assemble only one load at a time. Any additional loads that are awaiting assembly accumulate in Q_Assemble until the operator is available (loads are served on a first come, first served basis). Once placed on the conveyor, the loads are sent to the process P_CompleteAssembly.

Loads executing the **P_CompleteAssembly** process travel down the conveyor to station sta2. Each load remains on the conveyor at sta2 while using R_Operator2 for a time that is uniformly distributed between 50 and 70 seconds, which is the amount of time required to complete the second step of the assembly. After using the operator, the assembled loads are sent to die.
Writing the procedures to simulate these activities requires using the following AutoMod actions:

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>The <code>get</code> action claims one unit of a resource’s capacity. If a resource has no available capacity when a load tries to claim it, the load is delayed until a unit of the resource’s capacity becomes available.</td>
</tr>
<tr>
<td>free</td>
<td>The <code>free</code> action frees one unit of a resource’s capacity. The same load that gets a resource must also free it.</td>
</tr>
</tbody>
</table>
| wait     | The `wait` action delays a load for a specified amount of time. You can create random delays by using a distribution with the `wait` action. For example, in this model, loads delay in the first assembly step for a time that is uniformly distributed between 85 and 115 seconds. The AutoMod syntax for defining a uniform distribution requires you to calculate the mean and offset for the range of possible values. To find the offset, subtract the minimum value from the maximum value and divide by two: \[
\frac{115 - 85}{2} = 15
\]
To find the mean, subtract the offset from the maximum value: \[
115 - 15 = 100
\]
The syntax for delaying the load is:  
```autoMod
wait for uniform 100,15 sec
```

- **Note**  
The AutoMod software supports several distributions for generating random values in a simulation, including constant, uniform, normal, and so on. For more information about distributions and the syntax required to use them in a model, see the AutoMod syntax Help.

| use      | As an alternative to using the `get`, `wait`, and `free` actions, you can perform the same claim, delay, and release of a resource using a single `use` action. |

**To edit the `P_EnterAssembly` and `P_CompleteAssembly` arriving procedures**

1. On the Process System palette, click **Source Files**. The Source Files dialog box opens.
2. Double-click **mycode.m**.
3. Insert the bold lines into the following procedures:

```autoMod
begin P_EnterAssembly arriving procedure
  move into Q_Assemble /* Moves the load into the queue */
  get R_Operator1 /* Claim the operator */
  wait for uniform 100,15 sec
  /* Delay for first step of assembly */
  wait for 10 sec
  /* Delay to place the load on the conveyor */
  move into Conv.sta1
  /* Get on the conveyor at station sta1 */
  free R_Operator1 /* Release the load’s claim on the operator */
  send to P_CompleteAssembly
  /* Leave the current process */
end
```
begin P_CompleteAssembly arriving procedure
  travel to Conv.sta2
  /* Travel from current location to station sta2 */
  use R_Operator2 for uniform 60,10 sec
  /* Delay to complete assembly */
  /* use = get, wait, and free combined */
  send to die /* Leave the simulation */
end

4. Click File > Save, then File > Exit.
5. Export the model.
6. Click Run > Run Model.
7. Click Yes to build the model.

The model takes a few minutes to compile. When the Message dialog box reads Ready to simulate, you are ready to run the model and watch the animation.

---

### Changing the animation step

The animation step is the period of simulated time between animation updates. Setting a shorter animation step results in smoother animation, but it also slows the simulation because graphics need to be redrawn more frequently. Conversely, setting a faster animation step results in more jumpy animation, but a faster simulation. The animation step at the beginning of a simulation is set to one second.

**To change the animation step**

1. Click Control > Animation Step.
   The Animation Step dialog box opens.
2. Change the animation step to 0.5, as shown below, then press ENTER.

   ![Animation Step Dialog Box](image)

   - **Note** The Animation Step dialog box also allows you to synchronize the simulation to a multiple of real time. This option is not discussed in this tutorial (see the AutoMod User’s Guide for more information).

3. Click OK to close the Animation Step dialog box.
   The animation step has now been changed to redraw graphics every half-second.
4. Zoom in on the conveyor, then press p to continue the simulation.

**Tip** You can double the animation step during a simulation by pressing D (uppercase). You can halve the animation step during a simulation by pressing d (lowercase).
During the simulation, resources change color to represent their state:
  - Green is busy.
  - Blue is idle.

Notice the operators change states from idle to busy, or blue to green, when they are claimed by loads in the simulation.

When you have watched the simulation to your satisfaction, return to the editing environment:

5. Press p to pause the model.
6. Click File > Run Model Editor.

**Review**

You have defined two operators. One operator assembles and places loads on the beginning of the conveyor, the other operator completes the loads’ assembly at the end of the conveyor.

You have defined a queue where loads wait for assembly at the beginning of the conveyor.

You have written arriving procedures for P_EnterAssembly and P_CompleteAssembly, which tell loads which resources to use, where to travel, and so on.

You learned how to change the animation step to make a simulation run faster or slower.
Section 5: Completing the conveyor system

In this section, you will learn how to do the following:

- Add conveyor sections and stations.
- Define and use load attributes.
- Define and use variables.
- Use Run Control snaps.
- Examine statistics from a standard report.

Model description

In this section, you will make the following changes:

1. Add conveyor sections to complete the conveyor (refer to the conveyor illustration on the next page).
2. Edit R_Operator2’s arriving procedure to send assembled loads to either station staout1 or station staout2, with equal probability of going to either station.
3. Define a load attribute to track how long each load is in the system. Print the value to the Message dialog box.
4. Define a variable to track the total number of loads processed.
5. Define a run control to automatically stop the simulation after 8 hours.

Adding conveyor sections

To add the remaining sections in the conveyor system

1. Open the conveyor system Conv.
2. On the Conveyor palette, click Single Line.
   The Single Line dialog box opens.
   
   ![Single Line dialog box]

   3. If the Orthogonal check box is not selected, select it now.
   4. Click the Snap to Section button.
   5. Open the Measurement dialog box.
6. Select **Track Mouse**.
   
   Use the following illustration for steps 7 through 9.

   ![Illustration of conveyor sections](image)

7. Click below the right end of sec1. This is the starting point for sec2.

8. Move the cursor down 30 feet to the end of sec2 (refer to the illustration), then click the mouse button again.

   The conveyor sections are perpendicular. A transfer (a rectangle) is automatically drawn between the two sections.

   **Tip** If you want to delete a section you are drawing and have only placed one end, press ESC. If you have placed both ends, see “Editing sections” on page 9 for information on deleting the section.


10. To draw section 5, click **Select** on the Conveyor palette. Select section sec4, then click **Edit > Copy**.
11. Change the **X To** value to **16** (this moves the section 16 feet on the X axis), then click **OK**, as shown below:

<table>
<thead>
<tr>
<th>Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amount of Move:</td>
</tr>
<tr>
<td>X</td>
</tr>
<tr>
<td>From:</td>
</tr>
<tr>
<td>To:</td>
</tr>
<tr>
<td>Rotate:</td>
</tr>
</tbody>
</table>

The copied section is automatically named **sec4_1**.

12. To rename the section, select **sec4_1**, click the **Edit** menu, then **Edit**.
13. Change the section name to **sec5**.
14. Click **OK, Quit Edit Each**.
15. Close the Measurement dialog box.

---

### Adding stations

Now you need to add exit stations where assembled loads can transfer out of the conveyor system.

- **To add exit stations to the conveyor system**
  1. On the Conveyor palette, click **Station**.
     
     The Station dialog box opens.

     ![Station dialog box]

     2. Change the station name to **staout1**.
3. Drag the station into the correct position on section sec4 (refer to the illustration below).

4. Drag station staout2 into the correct position on section sec5.

   Tip  If you are not happy with the placement of any station, you can move it:
   a. On the Conveyor palette, click Select.
   b. Select the station you want to move.
   c. Click Edit > Move.
   d. Use the mouse to click and drag the station to the desired location.
   e. Click OK.

   Your conveyor path and stations are now drawn.

5. Export the model.

---

**Editing the process system**

You are now ready to edit the process system and cause loads to travel down the rest of the conveyor before exiting the simulation. You will also learn how to track the throughput and cycle time of loads in the simulation.

**Adding an arrayed process (P_Out)**

Instead of allowing assembled loads to immediately leave the system, you now need to send the loads to one of the exit stations at the end of the conveyor. To direct the loads, you could create two separate procedures (one to send loads to station staout1 and the other to send loads to station staout2). However, because the only difference between the two procedures would be the name of the destination station, you can save time by creating two arrayed processes that share the same arriving procedure.
Arrayed processes are copies of the same process. Other entities (such as queues and resources) can also be arrayed. Arrays are useful when you need to create a group of entities that share the same characteristics. Defining the processes as an array makes modeling easier, because you only need to write and edit one arriving procedure for both processes.

Another benefit of using arrays is that they allow you to align multiple entities of different types to form assembly lines or work cells. In this model, you will associate each member of the process array with a corresponding exit station on the conveyor. That is, loads that are sent to the first process will also travel to the first exit station (staout1). Similarly, loads that travel to the second process will travel to the second exit station (staout2).

**To define a process array**

1. Open the process system.
2. On the Process System palette, click **Process**.
   The Process dialog box opens.
3. Click **New** to define a new process.
   The Define A Process dialog box opens.
4. Name the process **P_Out**, then press **ENTER**.
   Now you are ready to create the process array.
5. Change the **Number of Processes** to 2, then press **ENTER**.
   The Define a Process dialog box appears as shown below:

   ![Define A Process Dialog Box](image)

   6. Click **OK** to close the Define A Process dialog box.

   You have now defined two arrayed processes. In the AutoMod syntax, you can refer to an individual process by appending the process number (in parentheses) to the name. For example, **P_Out(1)** and **P_Out(2)**.

**Sending loads to a member of a process array**

When loads have completed assembly, they must be sent from the **P_CompleteAssembly** process to one of the two arrayed **P_Out** processes. In this model, each load has a 50/50 chance of going to either the first arrayed process, **P_Out(1)**, or the second arrayed process, **P_Out(2)**.
To evenly distribute loads between the processes, you can use a oneof distribution in the *P_CompleteAssembly* arriving procedure. The oneof distribution allows you to randomly select from a series of values or entities based on the frequency of each selection. The syntax for selecting *P_Out1* and *P_Out2* with equal probability is shown in the procedure below.

**Note** For more information about the oneof distribution, see the AutoMod syntax Help online.

To edit *P_CompleteAssembly*’s arriving procedure

1. Edit the source file *mycode.m*.
2. Edit the last line of the procedure to appear as shown below:

```plaintext
begin P_CompleteAssembly arriving procedure
   travel to Conv.sta2 /* Travel from current location to station sta2 */
   use R_Operator2 for uniform 60,10 sec /* Delay to complete assembly */
   /* use = get, wait, and free combined */
   send to oneof(50:P_Out(1), 50:P_Out(2)) /* Each load has a 50 percent chance of going to either P_Out(1) or P_Out(2) */
end
```

Do not close the editor; you will define the *P_out* processes’ arriving procedure next.

**Sending loads to the correct exit station**

The arrayed processes *P_Out(1)* and *P_Out(2)* share the same arriving procedure. In the procedure you need to align each process with a corresponding exit station on the conveyor. If loads are sent to *P_Out(1)*, they need to travel to station *staout1*. If loads are sent to *P_Out(2)*, they need to travel to station *staout2* (see the illustration below).

![Diagram](image)

To align a member of the process array with a conveyor station, you can use the system keyword *procindex*.

**Using procindex to align arrayed entities**

A **system keyword** is a name that represents a numeric value in the software. You can use the system keyword *procindex* in the arriving procedure of an arrayed process to represent the current process’ index number. For example, if a load is sent to process *P_Out(1)*, the value of *procindex* is 1. If a load is sent to process *P_Out(2)*, the value of *procindex* is 2.
You can use the keyword procindex to align arrayed entities of different types. For example, assume loads are sent to an arrayed process, P_Burnish. The arriving procedure for P_Burnish is written as follows:

begin P_Burnish arriving procedure
  move into Q_Wait(procindex)
  use R_Machine(procindex) for 10 sec
  send to die
end

If a load is sent to the first member of the process array, P_Burnish(1), the value of procindex is 1. Consequently, the load moves into the first member of the queue array, Q_Wait(1), and uses the first member of the resource array, R_Machine(1).

If a load is sent to the second member of the process array, P_Burnish(2), the value of procindex is 2. Consequently, the load moves into the second member of the queue array, Q_Wait(2), and uses the second member of the resource array, R_Machine(2).

In this tutorial, you will use the keyword procindex to align loads in the P_Out procedures with the correct conveyor exit stations. Later, you will add arrayed queues, which will also be aligned using procindex.

**Defining P_Out’s arriving procedure**

- To define the arriving procedure that aligns each arrayed process with a corresponding conveyor exit station
  1. At the end of the source file, type P_Out’s arriving procedure, as shown below:

```plaintext
begin P_Out arriving procedure
  travel to Conv.staout(procindex)
  send to die
end
```

- **Note** If you are using procindex, the keyword must be enclosed in parentheses to distinguish it from the location name. This syntax is valid, because the AutoMod software allows you to include parentheses around the numeric portions of location names. For example, when writing a procedure, you could move loads into Conv.staout1 or Conv.staout(1).

  2. Click **File > Save**, then **File > Exit**.

  3. Export the model.

**Tracking cycle time**

Now you will track cycle time for each load using a load attribute. A load attribute is a user-defined entity that stores data. All loads have the same attributes; however, each load’s copy of an attribute may contain data that is unique to that load (such as color, part type, and cycle time).

To track cycle time, you will put a time stamp on each load as it enters the system. As each load leaves the system, you will compare the load’s time stamp to the current simulation time to determine how long the load was in the system.
Defining a load attribute

You can create a time stamp for each load by creating a load attribute, A_Time, and setting it to the current time when each load enters the system in P_EnterAssembly. When the load finishes P_Out, you will subtract the value in A_Time from the current time to calculate cycle time.

▶ To define the load attribute

2. Click New to the right of the Load Attributes list. The Load Attributes dialog box opens.
3. Name the attribute A_Time, then press ENTER. A load attribute can store many types of data (real, integer, time, and so on). In this model, the attribute A_Time represents the time a load spends in the system, and is therefore of type Time.
4. From the Type list, select Time. The Load Attributes dialog box appears as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>Dimension 1:</th>
</tr>
</thead>
<tbody>
<tr>
<td>A_Time</td>
<td>1</td>
</tr>
</tbody>
</table>

5. Click OK.

The attribute A_Time now appears in the Load Attributes list.

Before editing the model logic to track each load’s cycle time, you will create a variable to track the number of loads that complete processing in the simulation (throughput).

Tracking throughput

Tracking throughput involves counting the number of loads that complete processing in the simulation. To count each load, you will create a variable, V_Numdone, and increment the variable each time a load leaves the system. A variable is a user-defined entity that stores data. All loads have access to the same variables and (unlike load attributes) the value of a variable is the same for every load in the model.

▶ Tip When determining how to store data in a model, use variables to track information that applies to the entire model. Use load attributes to track information that is specific to each load.
Defining a variable

To define the variable V_Numdone
1. On the Process System palette, click Variables. The Variables dialog box opens.
2. Click New. The Define a Variable dialog box opens.
3. Name the variable V_Numdone, then press ENTER. Like attributes, variables can also be of different types. This variable is tracking integer data, so you do not need to change the type.
4. Click OK. The newly defined variable appears in the Variables list.


Editing the arriving procedures to track statistics

You are now ready to edit the model logic to use the load attribute and variable that you have created. To track cycle time, you will set the value of the load attribute A_Time to the current simulation time whenever a load enters the simulation. When a load leaves the simulation, you will compare the current simulation time with the attribute value to determine the cycle time.

To track throughput, you will increment the value of the V_Numdone variable each time a load leaves the simulation.

Editing the arriving procedures requires using the following AutoMod syntax:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>set</td>
<td>The set action sets the value of a variable or attribute to a defined value.</td>
</tr>
<tr>
<td>increment</td>
<td>The increment action increases the value of a variable or attribute by a defined value.</td>
</tr>
<tr>
<td>ac</td>
<td>The clock attribute ac refers to the current simulation time of the (absolute) clock. You can use the attribute ac in model logic to get the current time at any point during a simulation.</td>
</tr>
</tbody>
</table>
To track throughput

1. Edit the source file mycode.m.

2. Insert the bold line, shown below, into the P_EnterAssembly arriving procedure:

   ```
   begin P_EnterAssembly arriving procedure
   move into Q_Assemble /* Moves the load into the queue */
   set A_Time to ac /* Stamp the load with its creation time */
   get R_Operator1 /* Claim the operator */
   wait for uniform 100,15 sec /* Delay for first step of assembly */
   wait for 10 sec /* Delay to place the load on the conveyor */
   move into Conv.sta1 /* Get on the conveyor at station sta1 */
   free R_Operator1 /* Release the load’s claim on the operator */
   send to P_CompleteAssembly /* Leave current process */
   end
   ```

   The new line places a time stamp on each load as it enters the system. For example, a load that enters this process at simulation time 60 seconds has its A_Time attribute set to 60.0.

3. Insert the bold lines shown below into the P_Out arriving procedure:

   ```
   begin P_Out arriving procedure
   travel to Conv.staout(procindex) /* Travel to exit station */
   set A_Time to ac - A_Time /* Calculate cycle time */
   print this load, ”Time in system = ” A_Time to message
   inc V_Numdone by 1 /* Count throughput */
   send to die
   end
   ```

   The first two new lines finish tracking the cycle time for A_Time by subtracting the initial value of A_Time from the current clock (ac) and printing it to the Message dialog box. The last bold line increments the variable V_Numdone to track throughput.

4. Click File > Save, then File > Exit.

Defining run control

Currently, the model is defined to simulate indefinitely, until you pause or stop the simulation. You can define a run control to determine how long the model simulates and how often statistics are reported. A snap is a period of time after which statistics are written to the AutoMod report and are possibly reset. The AutoMod report is named Demo1.report and is saved in the model directory.
To define a snap in run control

1. Click Model > Run Control.
   The Run Control dialog box opens.

2. Click New.
   The Define Snap Control dialog box opens.

3. Change the Number of Snaps to 8, then press ENTER.
   By default, each snap is one hour long. Therefore, the model writes statistics to the report file every hour for eight hours.

4. Click OK.
   The Run Control dialog box opens, with the snap control description in its list. After eight snaps, the simulation will stop automatically.

Building and running the model

You are now ready to run the model and see the effect of your changes in the simulation.

To run the model

1. Export the model.
2. Click Run > Run Model, then build the model.
   The Simulation environment opens.
3. Zoom in on the conveyor.
4. Press p to continue the simulation.
   Watch to make sure loads travel down sections 4 and 5.
Notice that a message is printed to the Message dialog box every time a load leaves the system. Each load’s time in system varies slightly because of the random assembly times.

5. To run the simulation to completion, press g to turn off the animation.

The animation stops and the simulation continues at an accelerated speed.

Once the simulation has run for the length of time defined in the Run Control (eight snaps of one hour each), an End of run message appears in the Message dialog box and the simulation automatically stops.

---

### Checking statistics

Once the model has finished, you can check the simulation statistics by displaying them to the screen, or by viewing the saved AutoMod report.

- **Note** Depending on your placement of the conveyor stations, your statistics may vary slightly from those listed in this section.

### Displaying statistics to the screen

When displaying statistics to the screen, you can view the system-generated statistics for entities in the simulation, as well as the custom throughput statistic you defined using a variable in the model.

In this section, you will display statistics for the following:

- Resources
- Queues
- Processes

You will also display the value of the variables used in the simulation.

- **Note** You can display statistics and variable values both during the run and at the end of simulation. For more information about statistics, refer to chapter 7, "Running a Model," in the *AutoMod User’s Guide*.

### Displaying resource statistics

**To view the resource statistics for the two operators**

1. Click **View > Reports**.

   The Reports dialog box opens.

2. In the entity type list, select **Resources**, then click **Display**.

   The Resources Report dialog box opens. Some of the resource statistics are shown below:

<table>
<thead>
<tr>
<th>Name</th>
<th>Total</th>
<th>Util</th>
<th>Av_Time</th>
<th>Av_Wait</th>
</tr>
</thead>
<tbody>
<tr>
<td>R_Operator1</td>
<td>240</td>
<td>0.91</td>
<td>108.78</td>
<td>0.32</td>
</tr>
<tr>
<td>R_Operator2</td>
<td>238</td>
<td>0.49</td>
<td>59.47</td>
<td>0.00</td>
</tr>
</tbody>
</table>

*R_Operator1* assembled 240 loads (Total). In the model, *R_Operator1*’s assembly time is uniformly distributed between 85 and 115, with a mean of 100 seconds. Because the operator takes an additional 10 seconds to place each load on the conveyor, his average time of 109 seconds is reasonable. The
average wait of loads requiring assembly is very small (less than one second). Loads are required to wait for assembly because the operator is highly utilized; the operator is utilized about 91 percent of the time (Util).

R_Operator2 has assembled fewer loads than R_operator2 (238 compared to 240). This is because he must wait for loads to travel down the conveyor before he can begin work. R_Operator2’s assembly time is uniformly distributed between 50 and 70 seconds, so his average time is approximately the mean (60 seconds). R_Operator2 is only utilized about 50 percent of the time and consequently, loads have no waiting time to complete assembly.

Now take a look at the queue statistics.

**Displaying queue statistics**

- **To view the queue statistics**
  1. In the Simulation dialog box, click View > Report.
     The Reports dialog box opens.
  2. In the entity type list, select Queues, then click Display.
     The Queues Report dialog box opens. Some of the queue statistics are shown below:

<table>
<thead>
<tr>
<th>Name</th>
<th>Total</th>
<th>Cur</th>
<th>Average</th>
<th>Max</th>
<th>Av_Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q_Assemble</td>
<td>240</td>
<td>1</td>
<td>0.91</td>
<td>2</td>
<td>109.10</td>
</tr>
<tr>
<td>Space</td>
<td>240</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.00</td>
</tr>
</tbody>
</table>

The queue Space is a default queue where all loads start the simulation. You can see that a Total of 240 loads were created in the simulation. Newly created loads immediately execute the first action in the arriving procedure of their first process. In this model, their first action is to move into queue Q_Assemble. As a result, loads spend no time in Space.

All created loads (240) entered Q_Assemble. When the simulation stopped, there was one load still in the queue (the Cur statistic indicates the queue’s current quantity). The Max statistic indicates that a maximum of two loads were in the queue simultaneously during simulation (one load being assembled, and another load awaiting assembly). The average time that loads spent in the queue is equal to the average time that loads spent being assembled by R_Operator1 (108.78) plus the average time they spent waiting for the operator (0.32); displayed values are rounded to the nearest hundredth.

Now take a look at the process statistics.

**Displaying process statistics**

- **To view the process statistics**
  1. In the Simulation dialog box, click View > Report.
     The Reports dialog box opens.
2. In the entity type list, select **Processes**, then click **Display**.

   The Processes Report dialog box opens. Some of the process statistics are shown below:

<table>
<thead>
<tr>
<th>Name</th>
<th>Total</th>
<th>Cur</th>
<th>Av_Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>P_CompleteAssembly</td>
<td>239</td>
<td>2</td>
<td>134.91</td>
</tr>
<tr>
<td>P_EnterAssembly</td>
<td>240</td>
<td>1</td>
<td>109.10</td>
</tr>
<tr>
<td>P_Out(1)</td>
<td>108</td>
<td>0</td>
<td>125.50</td>
</tr>
<tr>
<td>P_Out(2)</td>
<td>129</td>
<td>1</td>
<td>109.45</td>
</tr>
</tbody>
</table>

   The statistics indicate that all created loads (240) entered the **P_EnterAssembly** process, and one load is currently processing.

   Notice that only 239 loads entered the **P_CompleteAssembly** process, and two loads are currently processing. To understand the difference between the process total (239 loads) and the resource total for **R_Operator2** (238 loads), remember that the **P_CompleteAssembly** process includes a travel action. The currently processing load is traveling to station **sta2**, but has not yet reached the operator. The travel time also explains why loads spent more time on average in **P_CompleteAssembly** (134.91 seconds) than in **P_EnterAssembly** (109.10 seconds).

   From the total statistics for **P_Out(1)** and **P_Out(2)**, you can see that the total number of loads sent to both processes is not equal (108 vs. 129). This is because the oneof distribution defines a unique probability for each load (the frequencies are not normalized).

   Now take a look at the custom statistic you defined in the model.

   **Displaying variable values**

   ▶ **To view the value of the variable defined in the model**

   1. In the AutoMod window, click **View > Variables**.

      The Variables Report dialog box opens, as shown below:

      ![Variables Report](image)

      The value of the **V_Numdone** variable indicates that 236 loads completed the simulation.

   ■ **Note** The **OPCQuality** and **OPCTimestamp** variables are standard system variables that appear only if your version of the AutoMod software is licensed to use the Model Communications module.

   2. Edit the model.
Viewing the AutoMod report

AutoMod automatically summarizes statistics during the run and writes them to a report file at the end of each snap. The report for your model is named Demo1.report.

▶ To view the report

Note You must close the Simulation environment before opening the AutoMod report.

Using any text editor, open the report file (Demo1.report).

Review

You have now completed the first phase of the model, which defined three processing operations: P_EnterAssembly, P_CompleteAssembly, and P_Out. The next phase adds an AGV system that carries loads through four other processes:

• Inspection
• Labeling
• Repair
• Storage
Section 6: Creating an AGV system

In the next few sections, you will add an automated guided vehicle (AGV) system to your conveyor model and add some more processing steps for your products. You will learn how to do the following:

- Add an AGV system.
- Adjust the grid.
- Draw paths.
- Define blocks.

Introduction

In AutoMod, automated guided vehicles are modeled using a path mover system. Path movers can represent vehicles, manually-operated lift trucks, or people. The path mover drawing tools can be used to create any type of vehicle system in which vehicles follow a specific path or route. When modeling a vehicle system, the concerns are movement, collision avoidance, and vehicle routing and scheduling.

Terms

There are several terms that you will need to understand before reading this part of the tutorial.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>control point</td>
<td>A location along a path at which vehicles can stop and receive instructions.</td>
</tr>
<tr>
<td>work location</td>
<td>A location at which a vehicle can find work.</td>
</tr>
<tr>
<td>park location</td>
<td>A location at which a vehicle can park when idle.</td>
</tr>
<tr>
<td>block</td>
<td>A barrier to limit how many vehicles can be in a physical area at one time.</td>
</tr>
<tr>
<td>vehicle scheduling lists</td>
<td>Lists that schedule a vehicle’s actions, such as where it can work and where it can park.</td>
</tr>
<tr>
<td>transfer</td>
<td>A connection between two sections of path.</td>
</tr>
</tbody>
</table>

Vehicle motion

Vehicles are controlled in three ways:

- Vehicle scheduling lists
- Vehicle procedures and functions
- Job scheduling

This tutorial discusses how to create and use vehicle scheduling lists; however, it does not teach vehicle procedures and functions or job scheduling, which give you greater control over routing vehicles in the simulation. For more information about using vehicle procedures and functions and job scheduling, see the “Vehicle Scheduling” chapter in the AutoMod User’s Guide.
Vehicles traveling on the same path avoid collisions by automatically decelerating and accumulating behind preceding vehicles. Every time a vehicle encounters congestion or reaches a destination, it adjusts its velocity to meet the circumstances. Vehicles decelerate to stop at destination control points, work locations, and park locations, but do not slow down to claim intermediate control points or blocks. Vehicles accelerate to their defined velocity whenever possible.

For a vehicle to avoid colliding with other vehicles at an intersection, it must claim strategically-placed blocks; blocks limit the number of vehicles that can travel in the bounded (blocked) area at the same time. Once paths are defined and the appropriate blocks are placed, AutoMod automatically calculates the shortest route between control points.

**System description**

You will expand the model by adding an AGV system, in which vehicles sequentially transport loads from the conveyor system to an inspection area, a repair area (if they fail inspection), a labeling area, and a storage area. Stored loads delay temporarily and then leave the system.
Assembled loads are picked up by AGVs at one of two pickup points. Loads are first delivered here for inspection; each load has a 10 percent chance of being scrapped and a 15 percent chance of needing repair. After inspection and/or repair, loads are delivered here for labeling. After labeling, loads are delivered for storage here, where they temporarily delay before leaving the system.
Adding an AGV system to the model

To define an AGV system in the model
1. Click System > New.
   The Create A New System dialog box opens.
2. From the System Type list, select Path Mover.
3. Name the path mover system Agv, then press ENTER.
4. Click Create.
   The Path Mover palette opens:

   ![Path Mover palette]

   The Path Mover palette allows you to accomplish several tasks: draw paths, place control points, define vehicles, and create scheduling lists. The palette also contains a Select tool.

Adjusting the grid

Before you draw a path, change the grid spacing so that the minor lines are 10 feet apart.

To change the grid spacing
1. In the AutoMod window, click the grid control icon.
   The Grid Control dialog box opens.
2. Edit the grid spacing values to:
   - Grid Spacing: 10 (press **ENTER**)
   - Minor line every: 1 (press **ENTER**)
   - Major line every: 10 (press **ENTER**)

3. Close the Grid Control dialog box.

---

**Drawing paths**

The process of drawing a path is similar to that of drawing a conveyor.

**To begin drawing a path**

1. On the Path Mover palette, click **Single Line**.
   
   The Single Line dialog box opens.

   ![Single Line dialog box]

   - **Name**: `path1`
   - **Orthogonal**
   - **Attributes**
   - **Snap to End**
   - **Snap to Path**

   ![Single Line dialog box]

**Note** Paths have attributes. Each path segment can be assigned a direction of travel, as well as a vehicle travel option. You will use the default path attributes in this tutorial.

   Straight paths can be placed at any angle and can be of any length.

2. If necessary, select the **Orthogonal** check box to draw lines that are perpendicular to each other and the grid.

3. Click the measurement icon, then select **Track Mouse**. Use this dialog box to draw the path to scale.

4. Zoom in on the drawing.
5. Draw four lines to represent the skeleton of your path (paths are drawn using the same procedure as conveyor sections). Use the grid to place them the correct distance from each other (see the illustration below).

**Note**
- Make sure the direction of the paths is correct. If you need to change the direction of a path, use the Select tool on the Path Mover palette to select the desired path, then select Change Direction from the Edit menu.
- If you need to make the grid larger, refer to “Expanding the drawing grid” on page 23 for information on how to resize the grid.

**Editing and deleting paths**

If you need to edit or delete a path, use the Select tool to select the path. Then, using the right mouse button, use the Edit menu to change the path’s direction, edit its length, or delete the path.

**Filleting paths**

The fillet tool connects two straight lines by drawing a curved section of path. Use this tool to connect the individual paths.
To use the fillet tool

1. On the Path Mover palette, click **Fillet**.

2. Drag a box over the end of the top path and the beginning of the right path (as shown below) or click each path to select it individually.

The Fillet dialog box opens.

The **Trim** option extends or shortens the endpoint of the section to match the endpoint of the arc.

3. Click **OK**.

The two paths are now connected.

- **Note** The box that appears between two sections is called a *transfer*. Transfers indicate that the paths are joined and that vehicles can move from one path to the other.

4. Fillet the remaining paths until the system looks like the following illustration.

- **Note** It is not important what the paths are named or what order they are drawn in for this system.
5. In the Measurement dialog box, clear the **Snap** check box.

6. Draw the straight lines of the inner loops as shown below (do not draw the parking area yet).

7. Fillet the new paths (do the inspection paths last).

   **Tip** When using the fillet tool on the last inspection path, you may need to decrease the fillet path’s radius; the inspection path should connect to the outer path (not the inner labeling area path).

8. Draw the parking loop’s straight lines as shown below:
9. Fillet those paths to complete the path.

Defining blocks

Now that you have drawn the path system, you need to define and place blocks to prevent vehicle collisions. A block is a boundary that you place on a path to limit the number of vehicles that can travel in the bounded (blocked) section of path at the same time. Blocks are necessary at intersections where vehicles merge from one path to another. In the system that you have drawn, there are four intersections where paths merge that require blocks.

To define the blocks

1. Open the process system.
   
   The Process System palette opens.

2. On the Process System palette, click Blocks.
   
   The Blocks dialog box opens.

3. Click New to define a new block.
   
   The Define A Block dialog box opens.

4. Name the block B_Merge, then press ENTER.

5. Change the Number of Blocks to 4, then press ENTER.

6. The capacity of the blocks is already set to the default value of one (to limit the number of vehicles that can travel in each block to one vehicle), so click OK to close the Define A Block dialog box.

You are now ready to place the block graphics.

Placing block graphics

The size of the block graphic determines the area of the path that is blocked. The default block graphic size is 1x1x1 feet. You will define larger blocks and place them on intersections where paths merge in the system.
To place the block graphics

1. In the Blocks dialog box, click **Edit**.
   The Edit A Block dialog box opens.
2. In the Edit A Block dialog box, click **Edit Graphic**.
   The Edit Block Graphics dialog box opens.
3. Select the first block, B_Merge(1).
4. Click **Place**, then place the graphic anywhere in the AutoMod window (after changing the size of the block, you will move it to the correct location on the path).
5. In the Edit Block Graphics dialog box, select the **Scale All** check box.
6. Change the scale value to 2.5, then press **ENTER**.

Notice that the X, Y, and Z scale boxes all change to 2.5, making the block scale 2.5x2.5x2.5 and more than doubling the size of the block graphic in the AutoMod window.

Now that you have scaled and placed the block graphic, you can move it into the correct position on the path.

7. Click **Move** in the Edit Block Graphics dialog box.
8. In the AutoMod window, drag the block's graphic so that its top edge is aligned with the transfer where the outer parking loop merges with the inner loop, as shown here:

![Diagram showing block alignment](image)

9. Place, scale, and move the remaining blocks so that they are aligned with the transfers where the parking, inspection and labeling loops merge with the outer path, as shown in the preceding graphic.

10. Click **Done** to close the Edit Block Graphics dialog box.
11. Click **OK** to close the Edit A Block dialog box.

**Hiding block graphics**

Because blocks are logical entities, block graphics are often hidden during animation (blocks are used to prevent collisions in the model and are not visible in a real-world system).

- **To hide block graphics**
  1. Click **Model > Display**.
     
     The Display dialog box opens.
  2. Clear the **Visible** check boxes for **Block** and **Block Names**.
  3. Click **OK** to close the Display dialog box.
     
     The block graphics disappear in the AutoMod window.
  4. Export your model.
Review

You have created an AGV system. In this system, you have drawn paths on which vehicles travel.
You have connected multiple paths using the fillet tool.
You have learned what transfers are and that they get created automatically when you connect paths.
You have placed blocks to prevent vehicle collisions when merging.
Section 7: Defining and placing control points

In this section, you will learn:
- What a control point is
- How to name and place control points
- How to add queues to the model

Control points

A control point is a location on the guidepath at which a vehicle can stop and receive instructions about where to work and park; vehicles can only search lists and execute procedures when stopped at a control point.

In addition to allowing vehicles to receive instructions, control points can also be used to limit the number of vehicles that can travel on a path at the same time. Each control point has a user-defined capacity that specifies the number of vehicles that can simultaneously claim the control point as a destination. The default capacity for control points is infinite, which allows an unlimited number of vehicles to travel to a control point at the same time. To limit the number of vehicles that can travel on the path leading to a control point, change the control point capacity to an integer value (for example, a capacity of two allows only two vehicles to travel on the path leading to the control point).

How vehicles claim control points

Each stopped vehicle claims at least one control point. A moving vehicle may have multiple control points claimed at once: the control point to which it is currently traveling, and one or more control points ahead of the vehicle (on the route to its destination). For example, for a vehicle to move from control point A to control point B, it must claim B before leaving A. If B has a capacity of one, this ensures that two vehicles are not traveling on the same segment of path at the same time. To avoid stopping at control point B, the vehicle attempts to claim control point C before reaching the place where it must begin decelerating to stop at control point B.

Defining control points

Now add control points for each process area. These points allow vehicles to stop at the processing areas and park when idle.

- To add control points
  1. Open the AGV system.
  2. On the Path Mover palette, click Control Point.
The Control point dialog box opens.

3. Name the control point `cpin1`, then press ENTER.

You do not need to edit the control point attributes. By default, the control point capacity is set to infinite, which allows an unlimited number of vehicles to claim and travel to a control point at the same time.

4. To place `cpin1`, drag the point into position below `staout1` of the conveyor system (refer to the illustration below).

You have now placed control point `cpin1`.

5. Place control point `cpin2` under conveyor station `staout2`.

6. Name the next point `cppark`, then place it on the parking loop (refer to the previous illustration).

7. Place the points `cpinspect`, `cplabelin`, `cplabelout`, `cpstore`, `cprepairin` and `cprepairout`, as shown in the previous illustration.

8. Export your model.

---

Adding queues to the model

The assembled loads need a way to get from the conveyor exit stations into the AGV system. You will define an array of queues and place a queue at the end of each conveyor section to hold loads while they wait for a vehicle to pick them up.
To define an array of queues
1. Open the process system.
2. On the Process System palette, click Queues.
   The Queues dialog box opens.
3. Click New.
   The Define A Queue dialog box opens.
4. Name the queue Q_Wait, then press ENTER.
5. Change the Number of Queues to 2, then press ENTER.
   The queue Q_Wait is now an array of two.
6. Click OK.
   The Queues dialog box opens.
7. Select Q_Wait, then click Edit.
   The Edit A Queue dialog box opens.
8. Click Edit Graphic.
   The Edit Queue Graphics dialog box opens.
9. Select Q_Wait(1) in the right list.
10. Click Place. Place the queue between the end of the conveyor and the path as shown in the next illustration (you may need to clear the Snap option in the Measurement dialog box before placing the queue).
11. Select Q_Wait(2), then place the graphic.
12. Click Done, then OK.
13. Export your model.
You have placed control points on the path where vehicles can stop and pick up or set down loads. The points all have an infinite capacity, allowing multiple vehicles to claim and travel to a control point at the same time.

Q_Wait(1) and Q_Wait(2) provide an area for loads to wait while transferring from the conveyor system to the AGV system.
Section 8: Defining vehicles

In this section, you will learn how to do the following:
- Define a vehicle.
- Define vehicle starting locations.

Defining a vehicle

In this section you will edit the default vehicle type, DefVehicle, and set the number of vehicles in the system to three. Each vehicle requires five seconds to pick up and drop off loads.

To edit the default vehicle type
1. Open the AGV system.
2. On the Path Mover palette, click Vehicle.
The Vehicles dialog box opens.
3. Click Edit to edit the default vehicle type’s characteristics.
The Edit A Vehicle Definition dialog box opens:
Vehicles have the following attributes.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle Type</td>
<td>The name of the vehicle you are currently editing.</td>
</tr>
<tr>
<td>Vehicle Capacity</td>
<td>The number of loads a vehicle can transport at one time. The default is one load. If the capacity is greater than one, the vehicles are multi-load vehicles. Multi-load vehicle scheduling is based on the concept of closest task for both picking up and dropping off loads.</td>
</tr>
<tr>
<td>Number of Vehicles</td>
<td>The number of vehicles of this type in the system.</td>
</tr>
<tr>
<td>Vehicle Start List</td>
<td>By default, vehicles can start at any location in the system. Those locations are chosen randomly at the beginning of the run. It is also possible to force your vehicle to originate from only one or from several control points.</td>
</tr>
<tr>
<td>Edit Graphics</td>
<td>Opens the Edit Vehicle Graphics dialog box, where you can place and translate the vehicle graphic. The graphic itself is defined at the segment level.</td>
</tr>
<tr>
<td>Initial Segments</td>
<td>Opens the Initial Segments dialog box, where you specify the vehicle’s initial segments and their order.</td>
</tr>
<tr>
<td>Specifications by Load Type</td>
<td>In addition to defining the vehicles themselves, you may describe the vehicles when they are carrying different load types. For example, vehicles might be required to carry very heavy or very wide loads, which can change characteristics such as speed. The standard load types are Default and Empty.</td>
</tr>
</tbody>
</table>

4. Change the **Number of Vehicles** to 3, then press **ENTER** to create 3 vehicles.

5. Click **Done**.

You have now edited all of the necessary vehicle attributes.

### Defining starting locations for vehicles

By default, vehicles can start at any location in the system. Those locations are chosen randomly at the beginning of the run. In this simulation, however, you want the vehicles to start at the parking location `cppark`. To do this, you will create a list that includes only that point and assign it as the vehicle start list. You create the list using a named list.

A named list is a list of points that can be used in another scheduling list, such as a work list or park list (you will learn about these other lists in the next section).

**To define a starting location for vehicles**

1. On the Path Mover palette, click **Named List**.
   
   The Named Lists dialog box opens.

2. Click **New**. The New Named List dialog box opens.

3. Type **Start**, then press **ENTER**. This is the name of the list.
4. Click **Create**.

   The Edit Named List dialog box opens.

   There are two areas in the Edit Named List dialog box: **List Items** (on the left) and **Location Selection List** (on the right). The Location Selection List displays all available locations in the system. You select names from the list on the right and add them to the list on the left, which becomes your named list.

5. In the **Location Selection List**, select `cppark`.

6. Click **Add After**.

   The location `cppark` appears in the list on the left.

7. Click **Done**.

   The Named Lists dialog box opens.

8. On the Path Mover palette, click **Vehicle**.

   The Vehicles dialog box opens.

9. Click **Edit**.

   The Edit A Vehicle Definition dialog box opens.

10. Click the **Vehicle Start List** button (it currently says **Random**). The Vehicle Start List dialog box opens.

11. Select **Start**, then click **OK** and **Done**.

    The vehicle type `DefVehicle` now has a start list that starts all vehicles at `cppark`.

---

**Defining a vehicle graphic**

You are now ready to import a graphic to represent the vehicles in the simulation.

► **To import a graphic**

1. On the Path Mover palette, click **Segment**.
2. Click **Edit**.

   The Edit A Segment Definition dialog box opens.

3. Click **Edit Graphic**.

   The Edit Segment Graphics dialog box opens.

4. From the **Shape Definition** list, select **Import**.

5. Navigate to the demos/graphics/cell directory in the software installation directory.

6. Double-click `agvk.cel`. The picture is imported into the AGV system.

7. Click **Done** to close the Edit Segment Graphics dialog box.

8. Click **Done** to close the Edit A Segment Definition dialog box.

► **To place a graphic**

1. On the Path Mover palette, click **Vehicle**.
2. Click **Edit**.

   The Edit A Vehicle Definition dialog box opens.

3. Click **Edit Graphic**.

   The Edit Vehicle Graphics dialog box opens.
4. Click **Place**.
5. Click anywhere in the AutoMod window to place the graphic.
6. Click **Hide**, then click **Yes** to confirm.
   - The graphic you just placed is hidden from view until you run the model.
7. Click **Done** to close the Edit Vehicle Graphics dialog box.
8. Click **Done** to close the Edit A Vehicle Definition dialog box.
9. Export the model.

---

**Review**

In this section, you edited the default settings for the vehicles in the system and defined `cppark` as their starting location. You also imported a graphic for vehicle segments.
Section 9: Scheduling vehicles

In this section, you will learn how to do the following:
- How to control vehicles through scheduling lists
- The definition of a work list
- The definition of a park list
- How to sort lists
- How to move loads from the conveyor to the AGV system

Scheduling vehicles

In the AutoMod software, you can schedule vehicles by creating scheduling lists at locations in the vehicles’ movement system. Scheduling lists are lists of locations where vehicles can search for their next task. Vehicle tasks include:
- Delivering loads
- Picking up (retrieving) loads
- Moving to park
- Parking (idle)
The time that vehicles spend in each of these states is reported in the vehicle statistics.

**Note** When moving, vehicles automatically search for the shortest route to their destination.

Scheduling lists

There are five types of scheduling lists that you can define for a location in a movement system:
- Work lists
- Park lists
- Load activation lists
- Vehicle activation lists
- Load search lists

Named lists, which you learned about in the previous section, can be used in any of these other lists.

**Note** In this tutorial, you will only define work lists and park lists. For more information about the other scheduling lists, see the "Vehicle Scheduling" chapter in the *AutoMod User’s Guide*.

Work and park lists

A *work list* is a list of locations at which a vehicle looks for work (a load to pick up). When a vehicle becomes idle or is awakened, the first thing it does is look for work at the current location (this does not require a work list). If you want the vehicle to look for work at other locations, list those locations in a work list.
For example, if a vehicle is idle at cp1, it checks to see if any loads are waiting to be picked up at its current location (cp1). If none are found, it searches all control points on cp1’s work list until it finds a task. If there isn’t any work, then the vehicle searches cp1’s park list.

A park list is a list of locations where an empty vehicle can park. The vehicle searches the list if it fails to find work. When a parking location with available capacity is found, the vehicle claims the point (reducing its available capacity) and moves to park at the location. If no parking location is found, the vehicle parks at its current control point.

Scheduling lists are always searched from the top down.

Building the system in phases

You have drawn all paths and placed all control points in the system. In this section, you will begin routing loads and scheduling vehicles in the system. To get started, you will schedule vehicles to pick up loads at the conveyor and take them to the storage area. Once that is working, you will implement the other areas: inspection, rejection, labeling, and repair.

Creating work lists

All locations where vehicles can park or set down loads in the system require a work list. Because you are not yet implementing all of the processing areas, you can ignore locations other than the model’s pickup, storage, and parking areas. Within these areas, there are two control points where you need to create work lists: the parking location (cppark) and the storage location (cpstore).

Vehicles begin the simulation at cppark and will return to this location when idle (that is, when they can’t find work on a work list). Therefore, cppark needs a list of locations where parked vehicles can look for work. After picking up loads, vehicles deliver them to cpstore and then must look for more work, so cpstore also needs a work list.

Tip  Vehicles will be more efficient if they search for work at pickup locations in the order of the locations’ distance to the vehicles’ current control point. Consequently, you should add control points to work lists in order of distance, with the closest control point listed first.

The two locations in the system where loads wait to get on a vehicle are cpin1 and cpin2 (the ends of the conveyors); these are the locations where work is available. Therefore, the work lists at cppark and cpstore must include these locations. Without a work list, the vehicles would remain parked during the entire simulation.

To create the work lists, you will define a work list for point cppark, and then copy the same list for point cpstore. Because both points are closest to cpin1, vehicles should search this location for work first, then cpin2.

To define a work list at point cppark

1. On the Path Mover palette, click Work List.
   The Work Lists dialog box opens.
2. Click New.
   The New Work List dialog box opens.
3. Select cppark, then click **New**.
   The Edit Work Lists dialog box opens.

4. Click **Add After**.
   The Add Work List Locations dialog box opens.

**Note**: Each location specified within a list must have a sorting option associated with it; sorting options are useful when adding groups of locations to a list. In this tutorial, you will add only single locations and will use the default sorting option **at**.

5. Select cpin1. This is the first location where vehicles look for work.
6. Click **Add**.
   Location cpin1 is added to the work list.
7. Click **Add After**.
   The Add Work List Locations dialog box opens.
8. Select cpin2, then click **Add**.
9. Click **Done**.
You have now defined the required work list for cppark. If vehicles do not find work at cpin1, they will look at cpin2.

**To copy the work list for point cpstore**
1. In the Work Lists dialog box, select cppark, then click **Copy**.
   The Copy Work Lists dialog box opens.
2. Select cpstore, then click **Copy**.
   The work list for cppark is copied for cpstore.

Now that you have created the required work lists in the system, you need to create a park list to instruct vehicles to travel to cppark when idle.

---

**Creating a park list**

All locations where vehicles can set down a load in the system require a park list. After dropping off a load, if a vehicle does not find any work, the empty vehicle searches the current location’s park list to determine where to travel so that it is not blocking the path.

In this section, vehicles will only drop off loads at cpstore. From there, send the vehicles to cppark.

**To create a park list**
1. On the Path Mover palette, click **Park List**.
   The Park Lists dialog box opens.
2. Click **New**.
   The New Park List dialog box opens.
3. Select cpstore, then click **New**.
   The Edit Park Lists dialog box opens.
4. Click **Add After**.
   The Add Park List Locations dialog box opens.
5. Select cppark in the Add Park List Locations dialog box.
6. Click Add. The location cppark is added to the list.
7. Click Done.
8. Export your model.

Modifying the model

Now that you have created the required scheduling lists in the path mover system, you need to edit the process system and route loads through the path mover system. You must make the following changes to the model:

- Add a queue for the storage area.
- Move loads from process P_Out to a new process, P_Store, which simulates the storage area. The loads will move from the conveyor system to the AGV system. You must then adjust the cycle time statistic to include the time spent moving the loads to the storage area.

Adding the storage queue

- **To add the queue for the storage area**
  1. Open the process system.
  2. On the Process System palette, click Queues. The Queues dialog box opens.
  3. Define a new queue named Q_Store using the default queue attributes.
  4. Place the default queue picture below cpstore on the bottom section of the path (you may need to turn Snap off in the Measurement dialog box):
Adding a new storage process

- To route loads from the conveyor system to the storage area
  
  Define a new process called \texttt{P\_Store} that uses the default process settings.
  
  You are now ready to define the process’ arriving procedure.

Defining the storage process’ arriving procedure

Currently, the last process that loads execute in the model is the \texttt{P\_Out} process. You will need to edit this process to send loads to the process \texttt{P\_Store}. You will also need to move the cycle time and throughput calculations to the \texttt{P\_Store} arriving procedure so the calculations will include the time that loads spend in the AGV system.

- To define the arriving procedure
  
  1. Edit the source file mycode.m.
  2. Define a new arriving procedure for \texttt{P\_Store} at the end of the file:
     
     \begin{verbatim}
     begin P\_Store arriving procedure
     end
     \end{verbatim}
  3. Edit the \texttt{P\_Out} arriving procedure and cut the cycle time and throughput calculations:
     
     \begin{verbatim}
     set A\_Time to ac - A\_Time /* Calculate cycle time */
     print this load, “Time in system = ” A\_Time to message
     inc V\_Numdone by 1 /* Count throughput */
     send to die
     \end{verbatim}
  4. Paste the cycle time information into the \texttt{P\_Store} procedure.
     
     To move loads into the AGV system, you will edit the \texttt{P\_Out} arriving procedure to move loads first into a queue (where they can wait for a vehicle) and then into one of the two pickup control points (either \texttt{cpin1} or \texttt{cpin2}). You will use \texttt{procindex} in the procedure to align a conveyor exit station with an arrayed queue and control point.
  5. Add the following bold lines to the \texttt{P\_Out} arriving procedure:
     
     \begin{verbatim}
     begin P\_Out arriving procedure
     travel to Conv\_staout(procindex)
     move into Q\_Wait(procindex) /* Location to wait for a vehicle */
     move into Agv\_cpin(procindex) /* Get on vehicle */
     send to P\_Store
     end
     \end{verbatim}

    The two \texttt{move} lines cause the loads to move into the appropriate queue and AGV location based on their arrayed process. The new \texttt{send} line sends loads to the process \texttt{P\_Store}.
6. Edit P_Store’s arriving procedure and insert the following bold lines to move loads through the AGV system and into the storage queue:

```mod
begin P_Store arriving procedure
travel to Agv.cpstore
/* Travel onboard vehicle to cpstore */
move into Q_Store /* Get off vehicle */
wait for 20 sec
set A_Time to ac - A_Time
print this load, “Time in system =” A_Time to message
inc V_Numdone by 1
send to die
end
```

7. Click **File > Save**, then **File > Exit**.

8. Export your model.

---

**Verifying the model**

Now run the model to verify that loads are moving through the AGV system correctly. Vehicles pick up loads at Q_Wait, then take them to Q_Store. When idle, the vehicles park at cppark.

▶ **To run and verify the model**

1. Run the model with the animation on.
2. Turn off the animation and let the simulation run to completion.
3. Click **View > Variables**.

   V_Numdone should show approximately 236 loads completed (your model may vary slightly depending on path distances and control point locations).

---

**Review**

You have changed the processing flow so that loads no longer leave the simulation in P_Out; instead they are delivered to the storage area by a vehicle. The storage area consists of a process (P_Store) and a queue (Q_Store).

Vehicles start at cppark. The vehicles search cpin1 and cpin2 for work, and when loads are waiting to be picked up, the vehicles transport them to cpstore. After dropping off loads at cpstore, the vehicles look for more work at cpin1 and cpin2. If no work is found, the vehicles return to park at cppark.

You have also moved the code that tracks cycle time and throughput from P_Out to P_Store, so that the statistics include the time that loads spend in the AGV system.

In the next section, you will expand the model to include the inspection, labeling, and repair processes.
Section 10: Adding remaining processes

In this section, you will review how to do the following:
- Create processes, queues, and resources.
- Place graphics.
- Edit work and park lists.

Model description

In this section you will add three new processes: inspection, labeling, and repair. In addition, some loads will be rejected and scrapped from the model in a fourth process: reject.

These new processes use resources and queues. The new processes also require some changes to the work and park lists in the model.
Adding queues

Before adding the new processes, add three new queues to your model: one for the labeling area, one for the repair area, and one for the rejection process.

**To add these queues**

1. Define the following three new queues with a capacity of infinite:
   - Q_Repair
   - Q_Label
   - Q_Reject

2. Place the queue graphics (refer to the following illustration). You may need to turn Snap off in the Measurement dialog box.

3. Export the model.

Adding resources

Now you will add three new resources to your model: one for inspecting the loads, one for labeling the loads, and one for repairing the loads.

**To add these new resources**

1. Define three new resources:
   - R_Inspect
   - R_Label
   - R_Repair
   
   Use the default resource attributes for each resource.

2. Edit the graphic for R_Inspect.

3. Select Import from the Shape Definition list in the lower left corner of the dialog box.
4. Navigate to the demos/graphics/cell directory in the software installation directory (AutoMod by default).

5. Double-click the file man.cel.

6. Place the operator next to \( Q_{\text{Reject}} \), as shown below:

![Diagram showing the placement of resources]

7. Repeat this process to place graphics for the other resources. You will need to rotate the graphics around the Z axis, as shown in the table below:

<table>
<thead>
<tr>
<th>Resource</th>
<th>Z rotation amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>R_Label</td>
<td>180 degrees</td>
</tr>
<tr>
<td>R_Repair</td>
<td>270 degrees</td>
</tr>
</tbody>
</table>

### Creating the new processes

Add four new processes to accommodate for the labeling, repairing, storage, and inspection/rejection areas in the model:

- \( P_{\text{Label}} \)
- \( P_{\text{Repair}} \)
- \( P_{\text{Reject}} \)
- \( P_{\text{Inspect}} \)

### Defining the process arriving procedures

Instead of sending loads from the conveyor system directly to storage, loads must now first travel to the inspection process. You will edit the model logic to reroute loads and define the procedures for each of the new processing areas.
To reroute loads and define the procedures

1. Edit the source file mycode.m.

2. Edit the P_Out arriving procedure and edit the last line to send loads to P_Inspect instead of P_Store:
   ```
   begin P_Out arriving procedure
   travel to Conv.staout(procindex)
   move into Q_Wait(procindex)
   /* Location to wait for a vehicle */
   move into Agv.cpin(procindex) /* Get on vehicle */
   send to P_Inspect
   end
   ```

3. Define a new arriving procedure for the inspection process P_Inspect:
   ```
   begin P_Inspect arriving procedure
   travel to Agv.cpinspect /* Travel to the inspection area */
   use R_Inspect for 30 sec /* Inspect while onboard the vehicle */
   send to oneof(10:P_Reject, 15:P_Repair, 75:P_Label)
   end
   ```
   The inspection procedure sends loads to the control point cpinspect. There the loads are inspected while still on the vehicle. The inspection process sends loads to one of three processes using a oneof distribution (each load has a 10 percent chance of being scrapped, a 15 percent chance of needing repair, and a 75 percent chance of passing inspection and going directly to the labeling area).

4. Define the arriving procedure for the rejection process P_Reject:
   ```
   begin P_Reject arriving procedure
   move into Q_Reject /* Get off vehicle */
   wait for 30 sec
   set A_Time to ac - A_Time
   /* Calculate cycle time for rejected loads */
   print this load, ”was rejected. Total time in system was ”
   A_Time to message
   send to die
   end
   ```
   Rejected loads get off the vehicle in the inspection area and delay temporarily in Q_Reject before leaving the simulation. Because rejected loads leave the simulation, their cycle time must be calculated and printed in the P_Reject arriving procedure.

5. Define the arriving procedure for the repair process P_Repair:
   ```
   begin P_Repair arriving procedure
   travel to Agv.cprepairin /* Travel to repair area */
   move into Q_Repair /* Get off vehicle */
   use R_Repair for normal 6,1 min /* Repair delay */
   move into Agv.cprepairout /* Get on vehicle */
   send to P_Label
   end
   ```
   Loads that require repair travel by vehicle to cprepairin. Loads get off the vehicle and are repaired in queue Q_Repair for a time that is normally distributed with a mean of six minutes and a standard deviation of one minute.

Note For more information about the normal distribution, see the AutoMod syntax Help online.
After repair, loads wait to be picked up by a vehicle at cprepairout and then travel to the labeling process.

6. Define the arriving procedure for the labeling process P_Label:

```plaintext
begin P_Label arriving procedure
  travel to Agv.cplabelin /* Travel to labeling area */
  move into Q_Label /* Get off vehicle */
  use R_Label for 2 min /* Labeling delay */
  move into Agv.cplabelout /* Get on vehicle */
  send to P_Store
end
```

Loads get off the vehicle at cplabelin and delay for two minutes while being labeled. The loads then wait to be picked up by a vehicle at cplabelout and travel to the storage area where they leave the simulation.

7. Click File > Save, then File > Exit.

8. Export the model.

---

**Modifying the work and park lists**

Now you should create work and park lists to cause vehicles to service the new areas.

<table>
<thead>
<tr>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inspection</td>
<td>After picking up a load from the conveyor system, each vehicle travels directly to the inspection area. Inspection takes place onboard the vehicle. Vehicles that drop off rejected loads in the inspection area must look for work or a parking location at the other points in the system.</td>
</tr>
<tr>
<td>Repair</td>
<td>Vehicles carrying loads that need repair drop off loads at cprepairin. The vehicles must then look for additional work or a parking location.</td>
</tr>
<tr>
<td>Label</td>
<td>Vehicles carrying loads that need labeling drop off loads at cplabelin. The vehicles must then look for additional work or a parking location.</td>
</tr>
</tbody>
</table>

Because there are new locations where work becomes available, you should edit the model’s existing work lists to cause loads to look for work at these locations. You will also create new work and park lists at the new setdown locations.

The locations where vehicles set down loads or park are:

<table>
<thead>
<tr>
<th>Location</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>cppark</td>
<td>where vehicles park when idle</td>
</tr>
<tr>
<td>cpinspect</td>
<td>where rejected loads get off the vehicle to leave the system</td>
</tr>
<tr>
<td>cplabelin</td>
<td>where loads get off the vehicle to be labeled</td>
</tr>
<tr>
<td>cpstore</td>
<td>where loads get off the vehicle to be stored</td>
</tr>
<tr>
<td>cprepairin</td>
<td>where loads get off the vehicle to be repaired</td>
</tr>
</tbody>
</table>
The locations where work becomes available are:

<table>
<thead>
<tr>
<th>Location</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpin1</td>
<td>where loads transfer from the conveyor system</td>
</tr>
<tr>
<td>cpin2</td>
<td>where loads transfer from the conveyor system</td>
</tr>
<tr>
<td>cplabelout</td>
<td>where labeled loads get back on a vehicle</td>
</tr>
<tr>
<td>cprepairout</td>
<td>where repaired loads get back on a vehicle</td>
</tr>
</tbody>
</table>

### Editing work lists

- **To edit the existing work lists so vehicles look for work in the labeling and repair areas**
  1. Open the AGV system.
  2. Edit the work list for cppark and append the labeling and repair pickup points, as shown below:

    **Work list for cppark**
    - At cpin1
    - At cpin2
    - At cplabelout
    - At cprepairout

    **Note** For optimal efficiency, locations are added to the vehicle list in order of their distance to the current control point, with closest locations listed first. Vehicles search locations on the list from the top down.

  3. Edit the work list for cpstore and insert the repair and labeling pickup points, in the order shown below:

    **Work list for cpstore**
    - At cprepairout
    - At cpin1
    - At cpin2
    - At cplabelout

### Defining work lists

To cause vehicles that set down loads in the inspection, labeling, and repair processes to look for work, you must define new work lists.

- **To define new work lists**
  1. Define a work list for cpinspect, as shown below:

    **Work list for cpinspect**
    - At cplabelout
    - At cprepairout
    - At cpin1
    - At cpin2
2. Copy the cpinspect work list for the control point cplabelin. Vehicles look for work at cplabelin in the same order of locations as vehicles at cpinspect.

3. Define a work list for cprepairin, as shown below:

   Work list for cprepairin
   At cprepairout
   At cpin1
   At cpin2
   At cplabelout

Defining park lists

The only location at which idle vehicles are allowed to park in the system is cppark. You have already created a park list for cpstore that causes vehicles to look for parking at this location.

- To create the park lists for the remaining locations where vehicles set down loads
  1. Copy the park list of cpstore for cpinspect, cplabelin, and cprepairin.
  2. Export the model.

- To verify the model
  1. Run the model. After picking up loads from the conveyor system, vehicles travel through the inspection area and to the other three processes.
  2. Watch the Message dialog box. Reject messages appear, stating how long loads were in the system before being rejected.

Tutorial summary

You have created a simple model that uses conveyors and vehicles to move loads through several processing steps. You moved the loads through the system using process arriving procedures.

You also learned how to read standard statistics, as well as create your own variables and attributes to track custom statistics.
Index

A
AGVS. See vehicles
aligning the grid 24
animation step 35
arriving procedures 16
   editing 33
   writing 17
axis 22
   displaying 22
drawing
   conveyors 7
   paths 56

B
blocks 52–53
   defining 60
   hiding graphics 62
   placing 60

C
centering views 2
closing models 20
commands, keyboard 26
configuration, saving 26
control points 52, 64
   defining 64
conveyors, drawing 7
creating
   a conveyor system 5
   a new model 4
   a path mover system 55
loads 28
cycle time 43

defining
blocks 60
   control points 64
load attributes 44
loads 13
processes 11
queues 31, 66
resources 29
run control 46
stations 39
vehicles 68
views 23
displaying
   graphics 25
   grid 25

drawing
   conveyors 7
   paths 56

E
editing
   park lists 82
   sections 9
   work lists 82
exporting 5

F
filleting paths 57
friction 22

generation limit 14

G
graphics
   blocks 60
   displaying 25
   hiding 62
   loads 15
   queues 32
   resources 30
   vehicles 70
grid
   sizing 23
   spacing 25

H
hiding block graphics 62

K
keyboard commands 26

L
lines
   major 25
   minor 25
lists
   editing 82
   scheduling 52, 72
loads
   attributes 44
   creation rate 28
   defining 13
   graphics 15
locations
  park 52
  vehicle starting location 69
  work 52

M
  Measurement icon 8
models
  building 47
  closing 20
  opening 21
  running 19, 47
  saving. See exporting
  statistics 48
mouse
  buttons 2
  tracking 8
moving
  blocks 61
  stations 10

O
  opening
    models 21
      the AutoMod software 4
origin and orientation 24

P
  park lists
    creating 74
    editing 82
    park locations 52
path mover system
  creating 55
  deleting paths 57
  drawing paths 56
  editing paths 57
  filleting paths 57
perspective 22
placing
  blocks 60
  queues 32
  resources 30
  stations 9
procedures
  arriving 16–17
  editing 33
processes
  arrays 40
  defining 11
procindex 42

Q
  queues
    defining 31, 66
    graphics 32

R
  reports 51
  resources
    defining 29
    graphics 30
  rotating views 22
  running
    models 19, 47
    run control 46

S
  saving
    configurations 26
    models. See exporting
    views 26
  scaling
    views 22
  scheduling lists 52, 72
  editing 82
  scheduling vehicles 52, 72
  screen 22
  sections
    drawing 7
    editing 9
  segments 69
  snap 46
  solid views 22
  source files, defining 17
  spacing, grid
    major line 25
    minor line 25
  starting location (vehicles) 69
  startup configuration 26
  stations
    defining 39
    moving 10
    placing 9
  statistics 48
  step, animation 35
  systems
    conveyor 5, 7
    path mover 55
    process 5

T
  text, displaying 25
  tracking the mouse 8
  transfers 52
  translating views 22

U
  using the mouse 2

V
  vehicles
    claiming control points 64
    defining 68
    graphics 70
    segments 69
    starting location 69
views
   rotating  22
   saving  23, 26
   view control  22

W
windows, keeping on top  22
wireframe views  22
work lists
   creating  73
   editing  82
   work locations  52
world view  22

Z
zooming  2