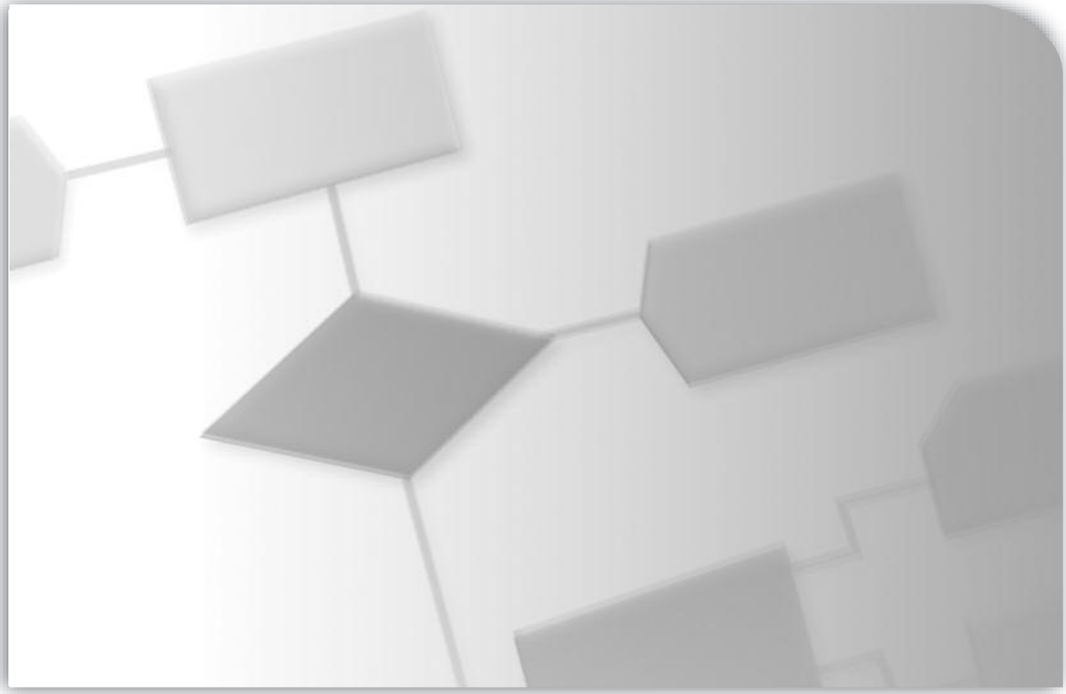
LISTEN.
THINK.
SOLVE.®

Arena®



USER'S GUIDE

PUBLICATION ARENA-UM001F-EN-P–March 2010
Supersedes Publication ARENA-UM001E-EN-P
PN-71363

Allen-Bradley • Rockwell Software

Rockwell Automation

Contact Rockwell Automation Customer Support Telephone — 1.440.646.3434
Online Support — http://www.rockwellautomation.com/support/

Warranty This product is warranted in accordance with the product license. The product’s performance may be affected by system configuration, the application being performed, operator control, maintenance, and other related factors. Rockwell Automation is not responsible for these intervening factors. The instructions in this document do not cover all the details or variations in the equipment, procedure, or process described, nor do they provide directions for meeting every possible contingency during installation, operation, or maintenance. This product’s implementation may vary among users.

This document is current as of the time of release of the product; however, the accompanying software may have changed since the release. Rockwell Automation, Inc. reserves the right to change any information contained in this document or the software at anytime without prior notice. It is your responsibility to obtain the most current information available from Rockwell when installing or using this product.

Version: 14.50.00
Modified: June 25, 2013 6:10:39 PM

Contents

A• Statistical Distributions 151

B• Library Shapes and Symbols 169

C• Using Visio and Arena 173

Index 183

1 Welcome

What is Arena software?

Arena® software allows you to bring the power of modeling and simulation to your business. It is designed for analyzing the impact of changes involving significant and complex redesigns associated with supply chain, manufacturing, processes, logistics, distribution and warehousing, and service systems. Arena software provides the maximum flexibility and breadth of application coverage to model any desired level of detail and complexity.

Typical scenarios include:

- detailed analysis of any type of manufacturing system, including material-handling components
- analysis of complex customer service and customer management systems
- analysis of global supply chains that include warehousing, transportation, and logistics systems
- predicting system performance based on key metrics such as costs, throughput, cycle times, and utilizations
- identifying process bottlenecks such as queue build ups and over-utilization of resources
- planning staff, equipment, or material requirements

Rockwell Automation offers a full suite of products to provide enterprise-wide simulation, optimization, and 3D model animation.

Intended audience

Arena software is designed for business consultants, business analysts, and industrial or system engineers. It can be widely deployed as a desktop tool and can serve as a natural extension to those who use various flowcharting packages such as Visio® to help demonstrate, predict, and measure system strategies.

Arena offers several editions of the software, ranging from the Arena Standard Edition up to the Professional Edition, which isthe ultimate in functionality and flexibility. All versions of the software build upon basic concepts familiar to users of the Microsoft® Windows® operating system and applications. A familiarity with the concepts and terms used in simulation is also helpful.

Your Arena license

Your software purchase of any commercial or academic edition includes a license that identifies the features that will be accessible for your use when your software is activated and run.

Arena Standard and Professional Edition (which also encompasses Academic licensing) software include module panels and templates that allow you to create simulations of more complex systems requiring material handling, advanced modeling concepts or models requiring more detailed analysis and flexibility.

New to Arena is Arena Visual Designer, the visualization application inside Arena for quickly and easily designing dashboards for advanced business graphic reporting and 3D animation in customizable display windows. Your software license will determine which features of Arena Visual Designer will be available to enhance your model's visualization. All licenses include the ability to enhance models with 3D animation in addition to dashboards.

Consult your Rockwell Automation sales representative or global representative to discuss configuration options that are best suited to your needs.

Where can I go for help?

Our commitment to your success starts with the suite of learning aids and assistance we provide for Arena. Whether you're new to simulation or a seasoned veteran putting a new tool to use, you'll quickly feel at home with the Arena software.

Refer to the user's guides

This *Getting Started with Arena* manual covers the product basics with an easy, "click-by-click" tutorial for modeling and analyzing a discrete process. *Getting Started with Arena* is a manual for all editions of the software. As such, chapters that are not applicable to a particular edition will be marked with a notation. The set also has a separate *Variables Guide* reference that provides complete descriptions of Arena variables found in the Arena product templates. Depending upon the software license, other user manuals may be included that are specific to different software and template components.

DOCUMENT CONVENTIONS

Throughout the guides, a number of style conventions are used to help identify material. New terms and concepts may be emphasized by use of italics or bold; file menu paths are in bold with a (>) separating the entries (for example, go to **Help >**

Arena Help); text you are asked to type is shown in Courier Bold (for example, in this field, type **Work Week**), and dialog box and window button names are shown in bold (for example, click **OK**).

Explore our examples

Arena is accompanied by a number of sample models that illustrate many of the commonly used approaches for capturing the essence of a variety of systems. For a list of Arena's examples, go to **Help > Arena Help**. On the **Contents** tab, choose **Model Building Basics**, and then select **Viewing Arena Example Models**.

Get help

Online help is always at your fingertips! Arena incorporates the latest in help features, including *What's This?* help that displays a brief description of fields in dialog boxes, context-sensitive help on menu and toolbar buttons, and a help button on each of Arena's modules. See the Arena Help table of contents and index for a list of all help topics.

Use the Smarts library

As you craft models of your own processes, use our Smarts library to explore how to best use Arena. This suite of tutorial models covers topics ranging from modeling resources to animation techniques. The library is organized into categories to help you find the right model with ease. When you're wondering how to take the next step in your model, browse the Smarts library for a ready-made solution. For a list of categories and their related Smarts, go to **Help > Arena Help**. On the **Contents** tab, first click **Model Building Basics**, and then **Learning Arena with Smart Files**.

Get phone support

Rockwell Automation provides full support for the entire Arena family of products. Questions concerning installation, how modules work, the use of the model editor, and the use of the software are handled by technical support.

Arena technical support includes:

- (for users on active maintenance) a technical support hotline and e-mail address staffed by full-time, experienced professionals
- help with installation problems or questions related to the software's requirements
- troubleshooting
- limited support regarding the interaction of Arena with other programs

- support of the Arena Object Model, which is used in Microsoft Visual Basic for Applications

If you call the support line (1.440.646.3434 (options 3 & 7) for calls originating in the U.S/Canada), be at your computer and be prepared to give the following information:

- the product serial number
- the product version number
- the operating system you are using
- the exact wording of any messages that appeared on your screen
- a description of what happened and what you were doing when the problem occurred
- a description of how you tried to solve the problem

International technical support is provided by the global representatives. For contact information on the representative nearest you, visit the Global Partners page on www.ArenaSimulation.com.

Get Web support

In addition to phone support, the Rockwell Automation Customer Support Center offers extensive online knowledgebases of technical notes and frequently asked questions for support of non-urgent issues. These databases are updated daily by our support specialists. The knowledgebase contains numerous Rockwell Automation products. Remember to search using Arena and any other appropriate descriptors as the search field criteria. Go to http://rockwellautomation.custhelp.com/ to sign up for online support.

If you can’t find the answer you need, contact your local representative or Arena technical support.

Refer to the Arena Web site

The Arena Web site provides a collection of brief “how-to” videos and FAQ topics that may be of assistance to you. This material and more is available through the Tools and Resources tab of www.ArenaSimulation.com.

Get training

Do you need training? Rockwell Automation offers a standard training course consisting of lectures and hands-on workshops designed to introduce you to the fundamental concepts of modeling with Arena.

We also offer customized training courses designed to meet your specific needs. These courses can be held in our offices or yours, and we can accommodate one to ten attendees. You design the course that's right for you! Simply contact our consulting services group to discuss how we can help you achieve success in your simulation efforts.

Get consulting services

Rockwell Automation provides expert consulting and turnkey implementation of the entire Arena product suite. Please contact our offices for more information.

Contact us

We strive to help all of our customers become successful in their process improvement efforts. Toward this objective, we invite you to contact your local representative or Rockwell Automation at any time that we may be of service to you.

Support E-mail: Arena-Support@ra.rockwell.com
Support Phone: 1.440.646.3434, options 3 & 7
General E-mail: Arena-Info@ra.rockwell.com
U.S. Sales Phone: 1.724.741.4000
URL: www.ArenaSimulation.com
URL: www.RockwellAutomation.com

2 Getting Started

Introduction

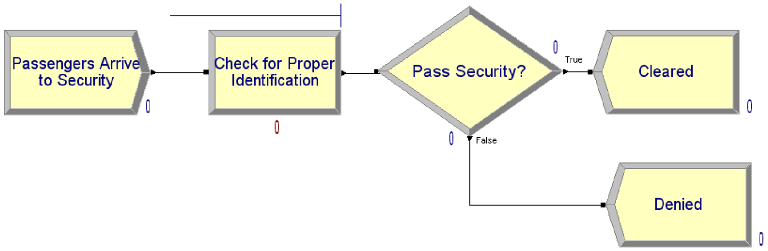
While you may not realize it quite yet, you now have the power to transform your business. Whenever you and others in your organization are wondering “what if…?,” you can look into the future to find the answer.

With Arena, you can:

- **model** your processes to define, document and communicate
- **simulate** the future performance of your system to understand complex relationships and identify opportunities for improvement
- **visualize** your operations with dynamic animation graphics
- **analyze** how your system will perform in its “as-is” configuration and under a myriad of possible “to-be” alternatives so that you can confidently choose the best way to run your business.

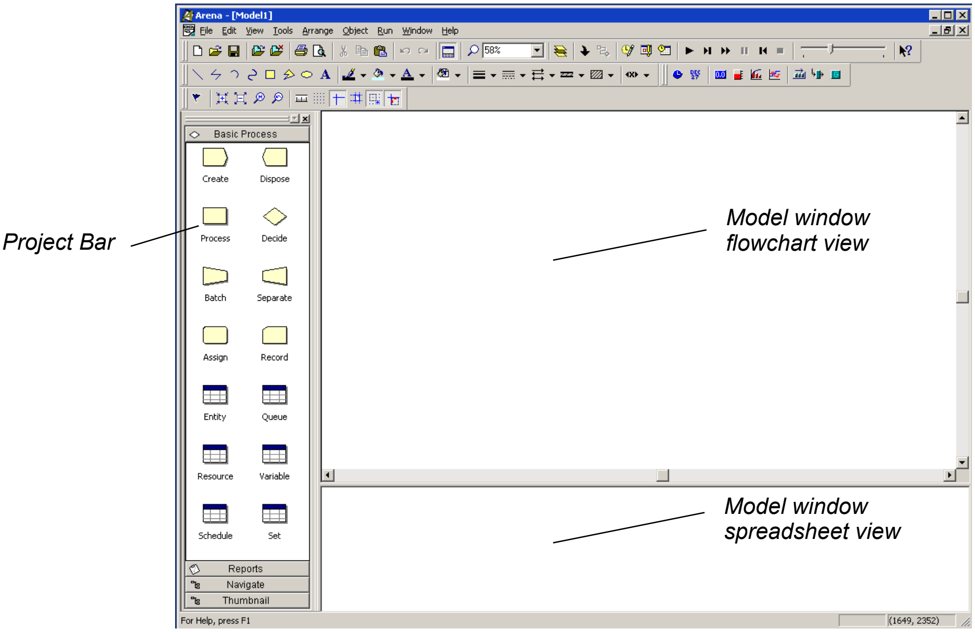
Our task: Analyze an airport security line

In this chapter, we will examine a simple airport security line to illustrate how you can model, simulate, visualize and analyze with Arena. To begin, we’ll look at the process of passengers arriving at security for identification inspection. We will build the flowchart shown below, introducing you to the process of modeling and simulating with Arena.


Passengers Arrive to Security
0
Check for Proper Identification
0
Pass Security?
0
True
0
False
Cleared
0
Denied
0

The Arena modeling environment

If Arena is not already running, start it from the Windows Start menu and navigate to **Programs > Rockwell Software > Arena**. The Arena modeling environment will open with a new model window, as shown below.



To model your process in Arena, you'll work in two main regions of the application window: the Project Bar and the model window.

The Project Bar usually is docked at the left of the Arena application window, but can tear off or dock at another position just like any toolbar.

The *Project Bar* hosts panels with the primary types of objects that you will work with:

- **Basic Process panel**: Contains the modeling shapes, called *modules*, that you'll use to define your process.
- **Reports panel**: Contains the reports that are available for displaying results of simulation runs.
- **Navigate panel**: Allows you to display different views of your model, including navigating through hierarchical submodels and displaying a model thumbnail.

The model window contains two main regions: the flowchart view and the spreadsheet view. The *flowchart view* contains all of your model graphics, including the process flowchart, animation, and other drawing elements. The lower, *spreadsheet view* displays model data, such as times, costs, and other parameters.

As we model the airport security process, we'll work in all of these regions of Arena.

Map your process in a flowchart

Let's start by examining what we're going to do: **Build a flowchart**. The word itself—*flowchart*—suggests two of the main concepts behind modeling and simulation. We'll be building a *chart*—also referred to as a *process map* or a *model*—that describes a *flow*.

This raises a key question in process modeling: **What exactly is it that will *flow* through the chart?**

We're modeling the process of reviewing passengers' identifications at the security checkpoint. These passengers with their identification are the items, referred to as *entities,* that will move through the process steps in our model. They are the data that are associated with our passengers' requests to pass through security into the airport terminal. As we build the flowchart, it's helpful to think of the process from the perspective of the *entity* (the identification review), asking questions like:

- Where do the passengers with their identification enter the process?
- What happens to them at each step?
- What resources are needed to complete work?

First, we'll draw the flowchart representing the security review process. See the Airport Security Process flowchart (shown previously) so you'll know what we'll be creating.

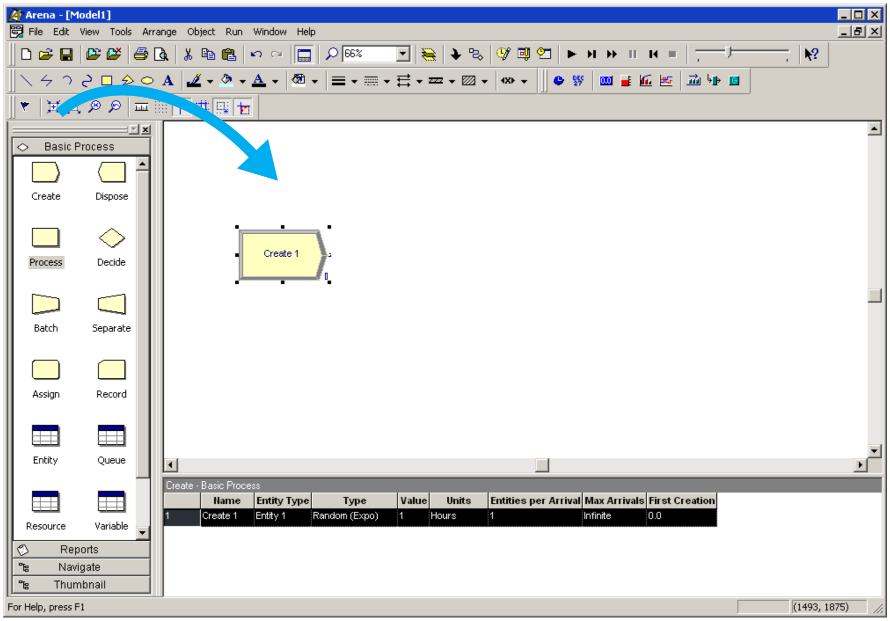
Create the security identification review entities

We'll start the flowchart using a *Create module*, from the Basic Process panel. This is the starting point for the flow of entities through the model.

1. Drag the Create module from the Basic Process panel into the model window.

   A default name, Create 1, is given to the module when it is placed. We'll return later to provide a more meaningful description as well as some data to support the simulation.

Every process flow starts with a Create module. When you simulate the flowchart, individual entities will be created according to timing information you supply in the Create module properties. After it's created, each entity moves from the Create module to the next shape in the process flow.



	Name	Entity Type	Type	Value	Units	Entities per Arrival	Max Arrivals	First Creation
1	Create 1	Entity 1	Random (Expo)	1	Hours	1	Infinite	0.0

Process the identification review

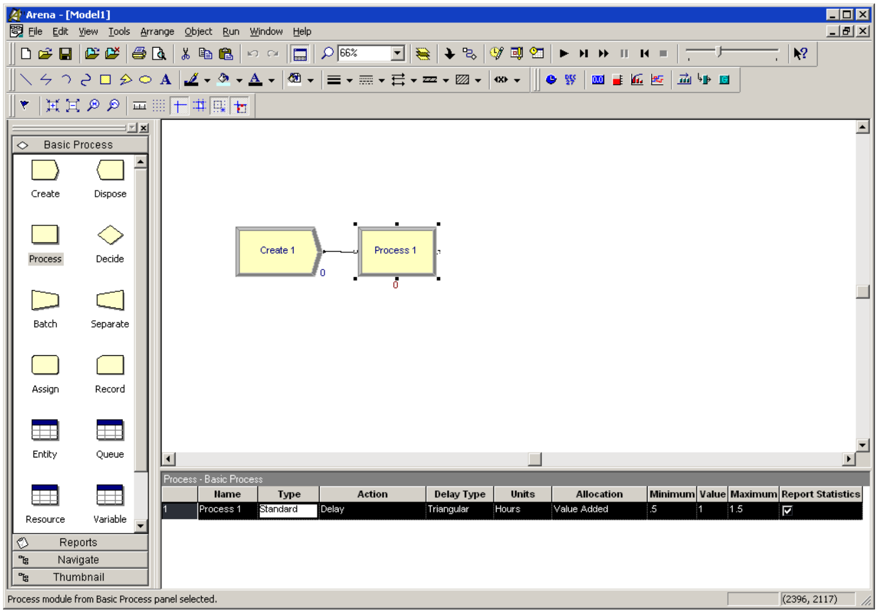
Next we add a *Process module* from the Basic Process panel to represent the Identification Review step.

1. Select the Create module so that Arena will automatically connect the Process module to the Create module.

2. Drag a Process module from the Basic Process panel into the flowchart region of the model window, placing it to the right of the Create module. Arena automatically connects the two modules.

   As with the Create module, the Process module has a default name that we'll replace later.

If your Create and Process were not connected automatically when you placed the Process, check the Object > Auto-Connect menu to verify that it is checked. If it is not, select it to turn on this option.



Process - Basic Process

	Name	Type	Action	Delay Type	Units	Allocation	Minimum	Value	Maximum	Report Statistics
1	Process 1	Standard	Delay	Triangular	Hours	Value Added	.5	1	1.5	☑

Process module from Basic Process panel selected.

(2396, 2117)

***Note:** If no connection appears between Create and Process, click the **Object > Connect** menu item or the **Connect** toolbar button to draw a connection. Your cursor will change to a cross hair. Start the connection by clicking the exit point (▸) of the Create module, then click the entry point (■) of the Process module to complete the connection. If you need to make multiple connections, simply double-click the **Connect** button or select **Object > Connect** twice (the Connect button will remain depressed to indicate it is in multi-connect mode). Then draw as many connections as desired. A valid connection target (for example, entry point, exit point, or operand object) is now highlighted during a connection session if the pointer is hovered over it. To end the multi-connection session, click the **Connect** option again, press **Esc**, or right-click.*

How do I use Snap and Grid?

If your flowchart shapes aren't lining up properly, you can use Arena's snap and grid features to straighten them out. First, check the **Snap** option on the **View** menu so that newly placed shapes will be positioned at regular snap points. To realign the shapes you've already placed, select the main module shapes (the yellow boxes) by holding the **Ctrl** key and clicking each shape. Then, click the **Arrange > Snap to Grid** menu option to adjust their positions to align with grid points.

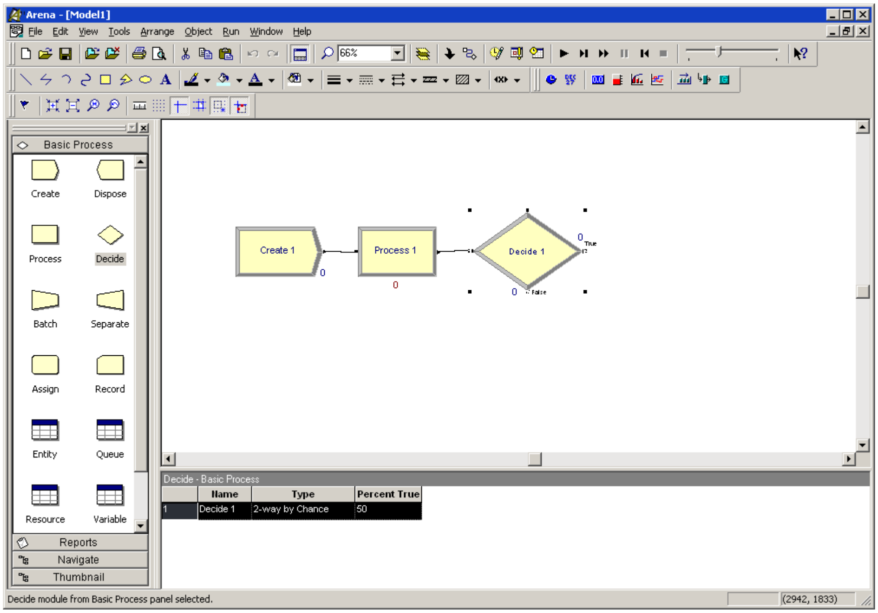
You can display the grid by checking the **Grid** option on the **View** menu. Both snap and grid are turned off by clicking the menu option again, clearing the check box.

Decide whether identification review is complete

After the Process module, we need to place a *Decide module* to determine whether the identification review is complete.

1. If you are using the Auto-Connect feature (that is, it is checked on the **Object > Auto-Connect** menu), be sure that the Process module is selected so that the Decide module will be connected to it.

2. Drag a Decide module to the right of the Process module.



Decide - Basic Process

	Name	Type	Percent True
1	Decide 1	2-way by Chance	50

Decide module from Basic Process panel selected.

(2942, 1833)

If the passenger identification review has a complete set of information, the review will exit the Decide module from the right side of the diamond shape, representing the True condition. Incomplete identifications (False result to the Decide test) will exit from the bottom connection.

Dispose of the reviews to terminate the process

Next we'll place the *Dispose module* from the Basic Process panel to represent accepted identification reviews, connecting to the True (right) output from the Decide module. Then, we'll complete the flowchart with another Dispose module for rejected identification reviews.

Entity flow always begins with a Create module and terminates with a Dispose module. You may have as many of each of these modules as you need to generate entities into the model and to remove them when their processing is complete.

1. Select the Decide module so that our first Dispose module will connect automatically.
2. Drag a Dispose module to the right of the Decide module. Arena will connect it to the primary (True) exit point of the Decide module.
3. To add the second Dispose module, once again select the Decide module so that Arena will automatically connect its False exit point to the new Dispose module.
4. Drag and drop another Dispose module, placing it below and to the right of the Decide module, completing the process flowchart.

What is a module?

In Arena, *modules* are the flowchart and data objects that define the process to be simulated. All information required to simulate a process is stored in modules.

For now, we're working with flowchart modules—those that are placed in the model window to describe the process. In the Basic Process panel, these are the first eight shapes:

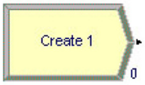
- **Create:** The start of process flow. Entities enter the simulation here.
- **Dispose:** The end of process flow. Entities are removed from the simulation here.
- **Process:** An activity, usually performed by one or more resources and requiring some time to complete.
- **Decide:** A branch in process flow. Only one branch is taken.
- **Batch:** Collect a number of entities before they can continue processing.
- **Separate:** Duplicate entities for concurrent or parallel processing or separating a previously established batch of entities.
- **Assign:** Change the value of some parameter (during the simulation), such as the entity's type or a model variable.
- **Record:** Collect a statistic, such as an entity count or cycle time.

Simulation settings are defined in the **Run > Setup > Replication Parameters** dialog box. There is also a set of *data modules* for defining the characteristics of various process elements, such as resources and queues.

Define model data

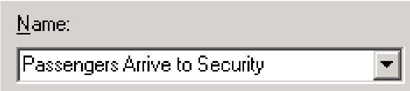
Now that we've drawn the basic flowchart for our passenger identification review process, let's define the data associated with the modules, including the name of the module and information that will be used when we simulate the process.

Initiate identification review (Create module)



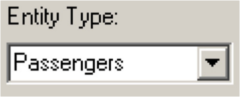
First, let's rename the Create module to *Passengers Arrive to Security*. Its data will include the type of entity to be created—in our case, a passenger identification review. We also need to define how often identification reviews are initiated. We'll use an average of 2 minutes between passenger arrivals for review as a starting point for our model, and we'll make this a random activity to represent the natural variation in the timing of passenger reviews being submitted.

1. Double-click the Create module to open its property dialog box.
2. In the Name field, type **`Passengers Arrive to Security`**.



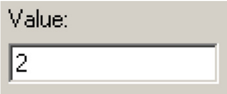
Name:
Passengers Arrive to Security

3. For the Entity Type, name our entities by typing **`Passengers`**.



Entity Type:
Passengers

4. Type **2** in the Value field of the Time Between Arrivals section.



Value:
2

5. Click **OK** to close the dialog box.

For now, we'll leave the default values for the other Create module properties. Feel free to explore their purposes through online help or the *Arrivals* models in the Smarts library.

What are entities?

Entities are the items—customers, documents, parts—that are being served, produced, or otherwise acted on by your process. In business processes, they often are documents or electronic records (such as checks, contracts, applications, purchase orders). In service systems, entities usually are people (the customers being served in a restaurant, hospital, airport, etc.). Manufacturing models typically have some kind of part running through the process, whether it's raw material, a subcomponent, or finished product. Other models might have different types of entities, such as data packets in network analysis or letters and boxes in package-handling facilities.

You may have different types of entities in the same model. For example, customers moving through a check-in counter at an airport might be separated into regular, first-class, and priority entity types. In some cases, entity types might be of an altogether different form rather than classifications of some basic type. For instance, in a pharmacy, prescriptions would be modeled as entities, running through the process of being filled. At the same time, customers might be competing for the pharmacist's attention with medical inquiries; they would also be modeled as entities.

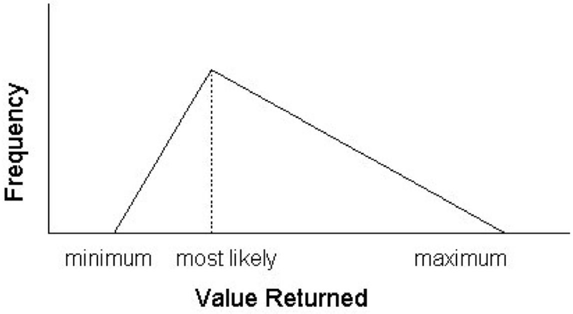
Review passenger identification (Process module)

Remember that as we create the flowchart, we're looking at the process from the perspective of the entity. The Create module is a starting point for an entity's flow through the system being modeled. Next, in our case, the passenger identification will be reviewed for completeness by a *Secutiry Agent*. Because this will take some amount of time, holding the entity at this point in the flowchart for a *delay* and requiring a *resource* to perform the activity, we use a Process module. We'll call this process *Check for Proper Identification*.

Appendix A describes the distributions available in Arena.

For the time delay, we also want to capture the natural variability that exists in most processes. Very often, for work done by people or equipment, a triangular distribution provides a good approximation. You specify the *minimum* time in which the work could be done, the *most likely value* for the time delay, and the *maximum* duration of the process.

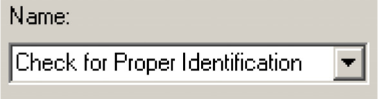
During the simulation run, each time an entity enters the process, Arena will calculate a sample from the distribution information you've provided—in our case, a triangular distribution. Over the course of a long simulation run where thousands of individual samples are taken, the times will follow the profile illustrated next.



For our Check for Proper Identification process, we'll use a minimum time of .75 minute, most likely value of 1.5 minutes, and a maximum of 3 minutes. We will assign a *resource*, the *Transportation Security Officer*, to perform this process.

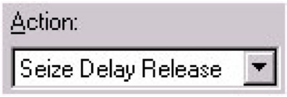
If more than one resource is required for a process to be performed, add as many as are necessary in the Process dialog's Resources list. An entity won't commence its process delay until all listed resources are available.

1. Double-click the Process module to open its property dialog box.
2. In the Name field, type **`Check for Proper Identification`**.



Name: Check for Proper Identification

3. To define a resource to perform this process, pull down the **Action** list and select **Seize Delay Release**.

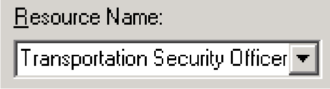


Action: Seize Delay Release

   Arriving entities will wait their turn for the resource to be available. When its turn comes, each entity will *seize* the resource, *delay* for the process time, and then *release* the resource to do other work.

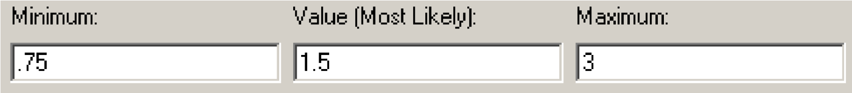
4. A list of resources will appear in the center of the dialog box. To add a resource for this process, click **Add**.

5. In the Resource Name field of the Resource dialog box, type **`Transportation Security Officer`**.



Resource Name:
Transportation Security Officer

6. Click **OK** to close the Resource dialog box.
7. Define the process delay parameters in the Minimum, Value (Most Likely), and Maximum fields as **`.75`**, **`1.5`**, and **`3`**. (Note that the default delay type is triangular and the default time units are in minutes.)

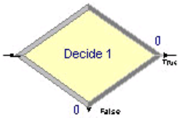


Minimum: .75
Value (Most Likely): 1.5
Maximum: 3

8. Click **OK** to close the dialog box.

   For now, we'll leave the default values for the other Process module properties. Feel free to explore their purposes through online help or the *Basic Concepts* and *Resources* models in the Smarts library.

Pass Security? (Decide module)

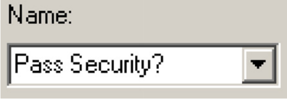


After each passenger's identification has been reviewed, we determine whether to pass or fail the passenger through security. In Arena, whenever an entity selects among branches in the process logic, taking just one of the alternatives, a Decide module is used.

For the airport security review process, we'll use a simple probability to determine the outcome of the decision, with 93% of passenger reviews accepted as passed.

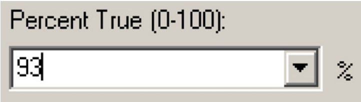
When you use a 2-way Decide module, the entity that enters the module leaves via one of the two exit points. If you want to make copies of an entity to model parallel processes, use a Separate module.

1. Double-click the Decide module to open its property dialog box.
2. In the Name field, type **`Pass Security?`**.



Name:
Pass Security?

3. For the Percent True field, type **93** to define the percentage of entities that will be treated with a “True” decision (that is, will depart through the exit point at the right of the Decide module).

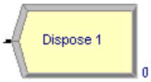


Percent True (0-100):

93 %

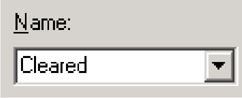
4. Click **OK** to close the dialog box.

Cleared, Denied (Dispose module)



In our simple process for reviewing passengers’ identifications at airport security, all the work that we’re interested in is done. Now, we’ll remove the passengers from the model, terminating the process with a Dispose module. Because there are two possible outcomes of the security passenger identification process—passengers can be cleared or denied—we’re using two Dispose modules that will count the number of applications under each outcome.

1. Double-click the first Dispose module (connected to the True condition branch of the Decide module) to open its property dialog box, and in the Name field, type **`Cleared`**.

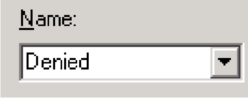


Name:

Cleared

2. Click **OK** to close the dialog box.

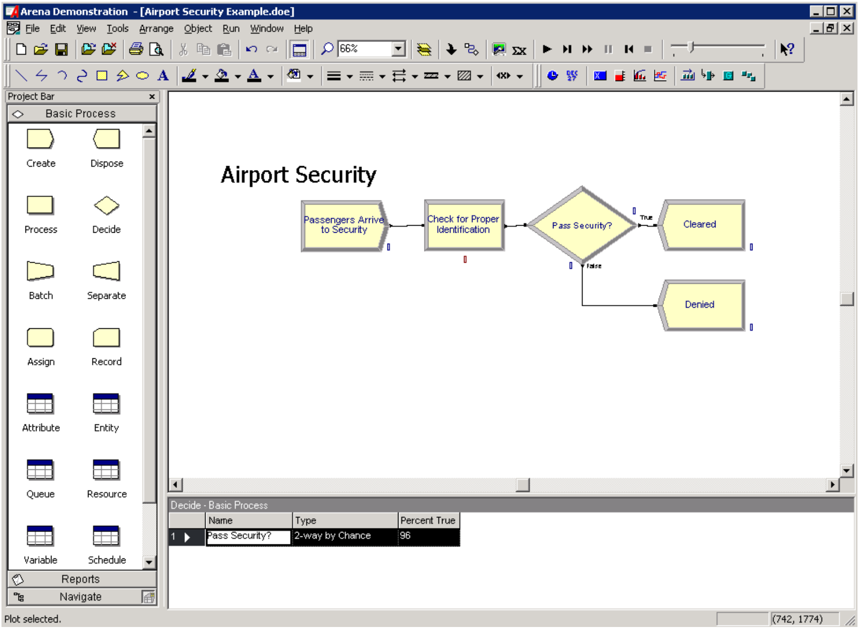
3. Double-click the other Dispose module to open its property dialog box. In the Name field, type **`Denied`**.



Name:

Denied

4. Click **OK** to close the dialog box.



	Name	Type	Percent True
1	Pass Security?	2-way by Chance	96

Transportation Security (Resource module)

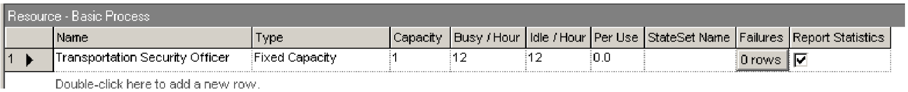
Along with our flowchart, we also can define parameters associated with other elements of our model, such as resources, entities and queues. For the airport security process, we'll simply define the cost rate for the security officer so that our simulation results will report the cost associated with performing this process. The officer's costs are fixed at $12 per hour.

To provide these parameters to the Arena model, you'll enter them in the Resources spreadsheet.

You can edit the fields for any module using Arena's spreadsheet, including flowchart modules. Just click on the icon in the Basic Process panel to display its spreadsheet.

1. In the Basic Process panel, click the Resource icon to display the Resources spreadsheet.

2. Because we defined the security officer as the resource in the Check for Proper Identification process, Arena has automatically added a resource with this name in the Resources spreadsheet. Click in the Busy/Hour cell and define the cost rate when the agent is busy by typing **12**. Click in the Idle/Hour cell and assign the idle cost rate by typing **12**.



Resource - Basic Process

	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures	Report Statistics
1 ▶	Transportation Security Officer	Fixed Capacity	1	12	12	0.0		0 rows	☑

Double-click here to add a new row.

Prepare for the simulation

To make the model ready for simulation, we'll specify general project information and the duration of the simulation run. Since we're just testing our first-cut model, we'll perform a short, 20-day run.

1. Open the Project Parameters dialog box by using the **Run > Setup** menu item and clicking the **Project Parameters** tab. In the Project Title field, type **Airport Security Analysis.** We'll leave the Statistics Collection check boxes as the defaults, with Entities, Queues, Resources and Processes checked, and we'll also check the Costing box.

2. Next, click the **Replication Parameters** tab within the same Run Setup dialog box. In the Replication Length field, type **20**; and in the Time Units field directly to the right of Replication Length, select **days** from the drop-down list. Click **OK** to close the dialog box.

Save the simulation model

Now that you've done some work on your model, it seems like a good time to save it.

1. Click **Save** on the Standard toolbar or select the **File > Save** menu item.

2. When Arena prompts you for a destination folder and file name, browse to the target folder in which you want to save the model (for example, C:\My Documents) and type a name (for example, **Airport Security**) in the file name field.

Arena's model files store all of the model definition, including the flowchart, other graphics you've drawn, and the module data you entered. When you perform a simulation run, the results are stored in a database using the same name as the model file.

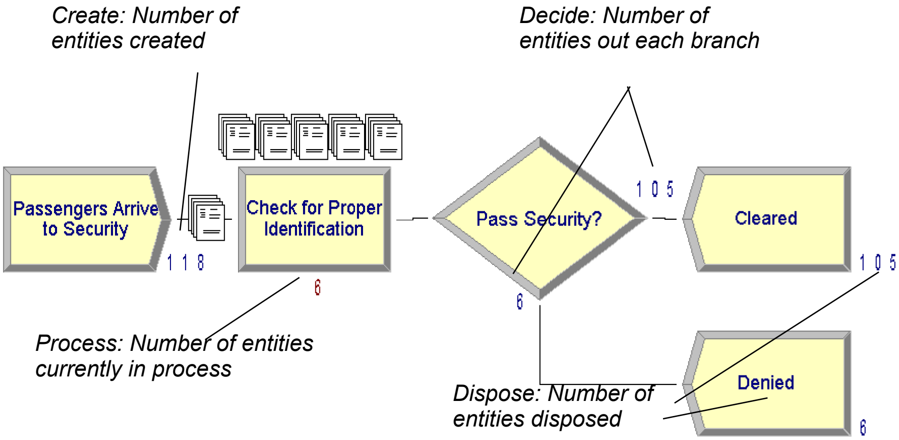
Simulate the process

With these few, short steps, we are ready to predict the future! The airport security passenger review model contains all of the information needed to run the simulation.

Start the simulation run by clicking the **Go** button or clicking the **Run** > **Go** menu item. Arena first will check to determine whether you've defined a valid model, then will start the simulation.

If Arena displays an error message, you can use the ***Find*** *button in the error window to locate the source of the problem. You can change between the error and model windows by selecting them from the Window menu.*

As the simulation progresses, you'll see small entity pictures resembling report documents moving among the flowchart shapes. Also, a variety of variables change values as entities are created and processed, as illustrated below.


Create: Number of entities created
Decide: Number of entities out each branch
Passengers Arrive to Security
118
Check for Proper Identification
6
Pass Security?
105
6
Cleared
105
Denied
6
Process: Number of entities currently in process
Dispose: Number of entities disposed

If the animation is moving too fast, you can slow it down by adjusting the animation scale factor. To do this, you have these choices:

The animation scale factor is the amount of simulated time between successive screen updates. Smaller values provide smoother, slower animation.

- Open the Run Setup dialog box by selecting the **Run > Speed > Animation Speed Factor** and then enter a smaller value (for example, `0.005`) for the scale factor.
- Use the less-than (<) key during the run to decrease the scale factor by 20%. Be sure that the model window is active—not the Navigate panel—or the less-than (<) and greater-than (>) keys won't take effect. Pressing these repeatedly is an easy way to fine tune the animation speed. The greater-than (>) key speeds up animation by 20%.
- Use the slider bar in the main toolbar. Move the slider to the left to slow down the animation and to the right to speed it up.

To pause the simulation, click the **Pause** button or press the **Esc** key. With the automatic flowchart animation, you can see how many entities have been created, are currently in the Check for Proper Identification process, have left each branch of our Decide module, and have left the model at each of our terminating Dispose modules. These variables can be helpful in verifying the model. For example, if the probability in the Decide module was entered incorrectly (for example, if you typed 9 as the denied probability instead of 93), the variables would show that many more applications were leaving the Denied branch.

*If the run finishes before you have a chance to explore these controls, answer No when you're asked if you want to view the results. Then click **Start Over** on the **Run** toolbar to begin the run again.*

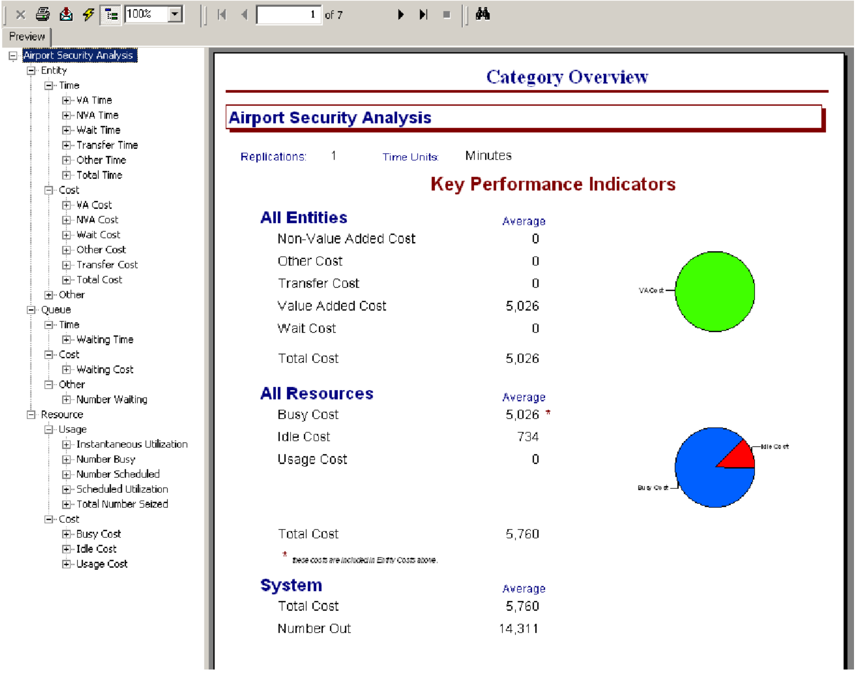
You also can step through the simulation one event at a time. To do this, **Pause** the simulation and then click the **Step** button or press the **F10** key. Each time you step the simulation, an entity is moved through the flowchart. Usually, you'll see animation of the entity's movement, though sometimes no visual change will take place (for example, when the next event is creating a new entity). When this occurs, just step again to move forward to the next event.

View simulation reports

After you’ve watched some of the animated flowchart, you can quickly run to the end of the simulation to view reports. To do this, **Pause** the simulation and then click the **Fast Forward** button to run the simulation without updating the animation.

At the end of the run, Arena will ask whether you’d like to view reports. Click **Yes** and the default report (the Category Overview Report) will be displayed in a report window, as shown below.

Each of Arena’s reports is displayed in its own window within the Arena application. You can use the standard window options (maximize, minimize, etc.) by clicking on the window control buttons or by pulling down the window menu.



- Airport Security Analysis
  - Entity
    - Time
      - VA Time
      - NVA Time
      - Wait Time
      - Transfer Time
      - Other Time
      - Total Time
    - Cost
      - VA Cost
      - NVA Cost
      - Wait Cost
      - Other Cost
      - Transfer Cost
      - Total Cost
    - Other
  - Queue
    - Time
      - Waiting Time
    - Cost
      - Waiting Cost
    - Other
      - Number Waiting
  - Resource
    - Usage
      - Instantaneous Utilization
      - Number Busy
      - Number Scheduled
      - Scheduled Utilization
      - Total Number Seized
    - Cost
      - Busy Cost
      - Idle Cost
      - Usage Cost

Category Overview

Airport Security Analysis

Replications: 1 Time Units: Minutes

Key Performance Indicators

All Entities	Average
Non-Value Added Cost	0
Other Cost	0
Transfer Cost	0
Value Added Cost	5,026
Wait Cost	0
Total Cost	5,026

All Resources	Average
Busy Cost	5,026 *
Idle Cost	734
Usage Cost	0
Total Cost	5,760

* these costs are included in Entity Costs above.

System	Average
Total Cost	5,760
Number Out	14,311

On the left side of each report window is a tree listing the types of information available in the report. The project name (in our case, Airport Security Analysis) is listed at the top of the tree, followed by an entry for each category of data. This report summarizes the results across all replications (although, in this model, we have only one replication). Other reports provide detail for each replication.

By clicking on the entries inside the category sections, you can view various types of results from the simulation run. The following table illustrates some of the questions you could answer from the Category Overview Report on our simple airport security analysis process simulation.

Question	Report Section	Answer
On average, how long did passengers spend in the modeled process?	Total Time (Entity), Average column	8.17 min
What was the average cost of reviewing a passenger's identification?	Total Cost (Entity), Average column	$0.35
What was the longest time a passenger spent in the process?	Total Time (Process), Maximum column	50.57 min
What was the maximum number of passengers waiting for identification check?	Number Waiting (Queue), Maximum column	31 passengers
What proportion of time was the security officer busy?	Utilization (Resource), Average column	87%

After you've browsed the Category Overview Report, you can close it by clicking on the window icon to the left of the **File** menu and clicking **Close**. You can look at other reports by clicking on their icons in the Project Bar. Each report will be displayed in its own window. To return to the model window, close all of the report windows or select the model file from the Window menu.

After you have viewed the reports and returned to the model window, end the Arena run session by clicking the **End** button.

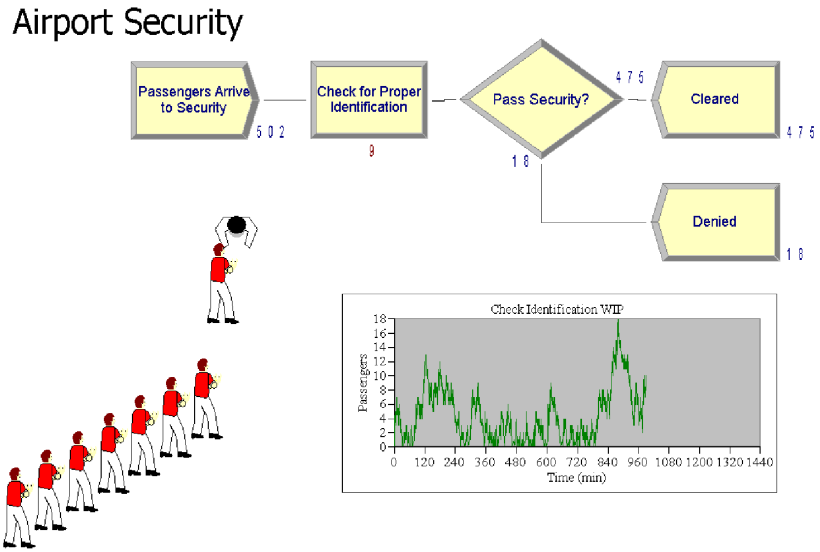
Enhance the 2D animation

In this section, we will demonstrate 2D animation, and in the Visual Designer section, we will learn about even more exciting animation and presentation capabilities.

Now that we've completed the basic steps for analyzing the passenger identification process, we can return to our model and embellish the graphical animation to gain further insight into the process dynamics. Animation also can be of great benefit in enticing others in the organization to be interested in process improvement.

We'll add two animation components to the airport security model. First, we'll show our security officer at the security checkpoint, either busy or idle. To gain a better sense of how many passengers are waiting in the Check for Proper Identification process over time, we'll also add a dynamic plot of the work-in-process (WIP) simulation variable. Our Arena model will appear as shown below after we add these two objects.

*Note: You can toggle between the split view (flowchart and spreadsheet) and a full-screen view of either area by clicking the **Split Screen** toolbar button or selecting the **View > Split Screen** menu item. When in full-screen view, clicking the icons on the Basic Process panel displays the appropriate view (flowchart for flowchart modules and spreadsheet for data-only modules).*



Animate the Security Officer resource

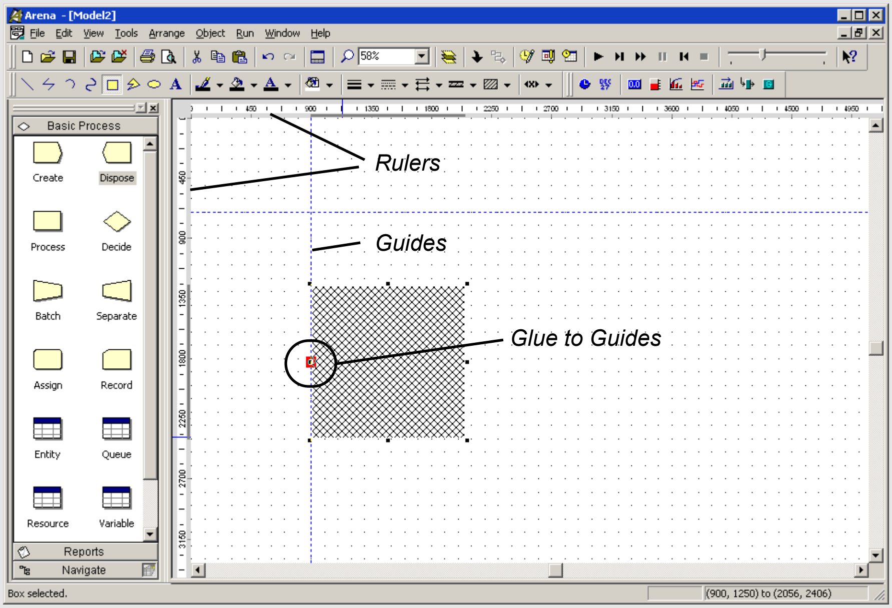
During the simulation run, our Transportation Security Officer resource can be in one of two states. If no passenger entity is in-process, the resource is *idle*. We'll use a top down view of a person to depict idleness. When an entity seizes the resource, the security officer's state is changed to *busy*, in which case our picture will show the officer with the passenger.

1. Click the **Resource** button on the **Animate** toolbar. The Resource Placement dialog box appears.
2. Select the **Transportation Security Officer** from the drop-down list in the Identifier field so that this object animates the security officer.
3. Open the People picture library by clicking the **Open** button, then browsing to the *People.plb* file in the Arena application folder (for example, C:\Program Files\Rockwell Software\Arena) and double-clicking it.
4. To change the idle picture:
   - Click the **Idle** button in the table on the left.
   - From the picture library table on the right, select the picture of the top-down person.
   - Click the **Transfer** button between the tables to use the worker picture for the Idle resource state.
5. To change the busy picture:
   - Click the **Busy** button in the table on the left.
   - From the picture library table on the right, select the picture of the top-down person.
   - Select the Seize Area check box within the Effects area of the resource editor.
   - Click the **Transfer** button between the tables to use the selected picture when the security officer is busy.
6. To further edit the resource, double-click the Idle or Busy button. Within this editor you can add objects to the picture, rotate the image and more. Close the editor to exit.
7. Click **OK** to close the dialog box. (All other fields can be left with their default values.)

8. The cursor appears as a cross hair. Move it to the model window and click to place the security officer resource animation picture.

9. If you'd like to have the security officer appear a bit larger, select the picture and use the resize handles to enlarge it.

How do I use Rulers and Guides?



Box selected.

(900, 1250) to (2056, 2406)

Selecting the **Rulers** command on the **View** menu toggles horizontal and vertical rulers at the top and left edges of each drawing window. The ruler labels display the (+/-) distance from the origin in Arena world units. The spacing between the ruler labels is based on the snap spacing, with the labels rounded to the nearest snap point if **Snap to Grid** is enabled.

When the **Guides** command (**View** menu) is activated, you may add guides into the application window to aid in positioning shapes precisely. You may either drag them from the horizontal or vertical ruler edges independently or drag them simultaneously from the origin corner. If you select a guide, you may drag it to a new position, or you may remove it by pressing **Delete**.

Activating the **Glue to Guides** command **(View** menu) allows you to drag a shape (module or drawing object) to a guide until the shape selection handle turns red (as shown in the circled call-out of the example), which indicates that the shape is "glued" to the guide. The guide may then be dragged to a new location and all shapes glued to the guide will maintain their alignment and will move with the guide.

Plot the number of passengers in-process

Our second animation enhancement is a plot of how many passengers are under review as the simulation progresses. This will give us a sense of the dynamics of the workload, which can vary quite a bit when the random nature of processes is incorporated into a simulated model.

1. Click the **Plot** button on the **Animate** toolbar.

   We'll plot a single expression, the work-in-process (WIP) at the Check for Proper Identification process.

2. To add a data series to a plot, go to the **Data Series** tabbed page and then click the **Add** button.

You can plot many expressions on the same set of axes by adding multiple expressions in the Plot dialog. Each can be color-coded so that you can readily compare data such as workloads in processes or waiting customers.

3. In the properties grid for the data series (by default it will be named 'Series1'), go to the *Source Data\Expression* property. This property defines the simulation expression to monitor and plot. Type in the expression `Check for Proper Identification.WIP`.

4. Now go to the **Axes** tabbed page and select the Left (Y) Value axis. Specify the value `2` for the *Scale\MajorIncrement* property, so that major tick marks on the y-axis are displayed over 2 units. (When setting the axis values, leave the default values for any property not specifically mentioned here.)

5. Now select the horizontal Time (X) axis. Specify the value `1440` for the *Scale\Maximum* property. This sets the horizontal time axis of the plot to be 1,440 minutes (or 24 hours) of simulated time.

6. Specify the value `120` for the *Scale\MajorIncrement* property, so that the major tick marks on the *x*-axis are displayed every 120 hours (or 2 days).

7. Explore the other properties of the graph to add titles, change colors and more.

8. Click **OK** to close the Plot dialog box. The cursor changes to a cross hair.

9. Draw the plot in the model window by clicking to locate the top-left and bottom-right corners of the plot below the flowchart and to the right of the resource.

With the edits now complete, you can save them by clicking **Save** or pressing **Ctrl+S.**

What is a model thumbnail?

The Navigate panel on the Project Bar provides a model navigation tree for maneuvering through the different named views and hierarchical levels defined in the model. Included as part of the Navigate panel is a model thumbnail that helps users orient and change the current view within the active drawing window.

The thumbnail displays the world space of the current drawing window. A colored, translucent rectangle represents the current view's boundaries. Dragging the rectangle in the thumbnail moves the drawing window's current view left, right, up, or down; however, the rectangle may not be dragged outside the boundaries of the world space.

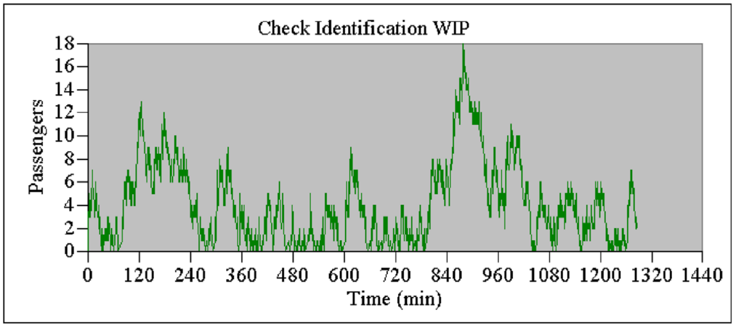
You may also re-center the rectangle, and thus the world space, or change the level of zoom. For details on model thumbnail, see the Navigate panel topic in online help.

Rerun the simulation

Now that we've made our animation more interesting and valuable, let's run the simulation again. Because we did not change any of the process parameters (that is, data in the modules), the simulation will provide the same results.

Click **Run** or press the **F5** key to start the simulation. As the simulation progresses, you'll notice the security officer's picture changes from idle to busy (engaging with a passenger) and back again, as passenger entities move through the Check for Proper Identification process.

The plot shows some significant peaks in the number of passengers who are in line with the security officer, caused by the combination of the variation in the time between arrivals of passengers (defined in the Create module) and the time to check for proper identification (Process module).



Next steps

You've succeeded in modeling, simulating, visualizing and analyzing a simple airport passenger identification screening process. To further explore Arena's capabilities, try solving a few of these extensions to the process.

1. Add a baggage check screening process following the identification check.

   Once passengers clear the identification check, they proceed to the X-ray baggage screening. At a minimum, it takes 1.5 minutes per passenger for the X-ray process to complete. Typically, this process takes 2.5 minutes. At the most, this process can last 7 minutes. There are two X-ray machines. What is the maximum time that a passenger had to wait in line for the X-ray machine? What is the utilization of the X-ray machines?

2. Enhance with a random security check for passengers.

   To increase security measures, it has been decided that a more extensive security check of passengers is necessary after the baggage scan. Every 15th passenger will go through a full-body scan and manual baggage review. This inspection usually takes 5 minutes. At the least, it will take 3 minutes and at the most 10 minutes. Assume there is unlimited availability of resources to model this process. What is the overall average cycle time of passengers (from the time they enter the system until they are through all security points) who are selected for this check?

   Hint 1: Use the Assign module and define a variable to keep track of the number of passengers through the process.

   Hint 2: Use the Assign module in combination with the Record to calculate the average cycle time.

   This problem uses concepts that have not yet been introduced. Proceeding through this entire guide and coming back to this problem is recommended!

To view the completed Arena models for the main tutorial and this extension, browse the Examples folder for *Airport Security Example.doe*, *Airport Security Example Extension 1.doe* and *Airport Security Example Extension 2.doe*.

Licensed users of Arena Standard and Professional Edtions can further enhance their visualization by using the Arena Visual Designer tool to create 3D animations of their models.

3 The Basic Process Panel

This chapter describes the flowchart and data modules that you'll use to model your process. Additionally, this chapter includes example uses of each module.

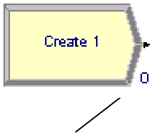
The Basic Process panel is accessible to all licensed editions of Arena.

Flowchart modules

Flowchart modules are the set of objects that are placed in the model window to describe the simulation process.

Create module

Description



Number of entities created

This module is intended as the starting point for entities in a simulation model. Entities are created using a schedule or based on a time between arrivals. Entities then leave the module to begin processing through the system. The entity type is specified in this module.

Typical uses

- The start of a part's production in a manufacturing line
- A document's arrival (for example, order, check, application) into a business process
- A customer's arrival at a service process (for example, retail store, restaurant, information desk)

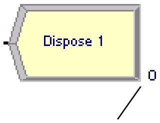
Prompts

Prompt	**Description**
Name	Unique module identifier displayed on the module shape.
Entity Type	Name of the entity type to be generated.
Type	Type of arrival stream to be generated. Types include **Random** (uses an exponential distribution, user specifies mean), **Schedule** (uses an exponential distribution, mean determined from the specified Schedule module), **Constant** (user specifies constant value; for example, 100), or **Expression** (drop-down list of various distributions).

Prompt	Description
Value	Determines the mean of the exponential distribution (if *Random* is used) or the constant value (if *Constant* is used) for the time between arrivals. Applies only when Type is Random or Constant.
Schedule Name	Identifies the name of the schedule to be used. The schedule defines the arrival pattern for entities arriving to the system. Applies only when Type is Schedule.
Expression	Any distribution or value specifying the time between arrivals. Applies only when Type is Expression.
Units	Time units used for interarrival and first creation times. Does not apply when Type is Schedule.
Entities per Arrival	Number of entities that will enter the system at a given time with each arrival.
Max Arrivals	Maximum number of entities that this module will generate. When this value is reached, the creation of new entities by this module ceases.
First Creation	Starting time for the first entity to arrive into the system. Does not apply when Type is Schedule.

Dispose module

DESCRIPTION



Number of entities that have been disposed

This module is intended as the ending point for entities in a simulation model. Entity statistics may be recorded before the entity is disposed of.

TYPICAL USES

- Parts leaving the modeled facility
- The termination of a business process
- Customers departing from the store

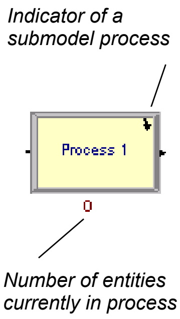
Prompts

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Record Entity Statistics	Determines whether or not the incoming entity’s statistics will be recorded. Statistics include value-added time, non-value-added time, wait time, transfer time, other time, total time, value-added cost, non-value-added cost, wait cost, transfer cost, other cost, and total cost.

Process module

Description

This module is the main processing method in the simulation. Options for seizing and releasing resource constraints are available. Additionally, there is the option to use a submodel and specify hierarchical user-defined logic. The process time is allocated to the entity and may be considered to be value added, non-value added, transfer, wait, or other. The associated cost will be added to the appropriate category.


Indicator of a submodel process
Process 1
0
Number of entities currently in process

Typical uses

- Machining a part
- Reviewing a document for completeness
- Fulfilling orders
- Serving a customer

Prompts

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Type	Method of specifying logic within the module. **Standard** processing signifies that all logic will be stored within the Process module and defined by a particular Action. **Submodel** indicates that the logic will be hierarchically defined in a submodel that can include any number of logic modules.

Prompt	Description
Action	Type of processing that will occur within the module. **Delay** simply indicates that a process delay will be incurred with no resource constraints. **Seize Delay** indicates that a resource(s) will be allocated in this module and delay will occur, but that resource release will occur at a later time. **Seize Delay Release** indicates that a resource(s) will be allocated followed by a process delay and then the allocated resource(s) will be released. **Delay Release** indicates that a resource(s) has previously been allocated and that the entity will simply delay and release the specified resource(s). Applies only when Type is Standard.
Priority	Priority value of the entity waiting at this module for the resource(s) specified if one or more entities are waiting for the same resource(s) anywhere in the model. Not visible when Action is Delay or Delay Release or when Type is Submodel.
Resources	Lists the resources or resource sets used for entity processing. Does not apply when Action is Delay, or when Type is Submodel.
Delay Type	Type of distribution or method of specifying the delay parameters. **Constant** and **Expression** require single values, while **Normal**, **Uniform**, and **Triangular** require several parameters.
Units	Time units for delay parameters.
Allocation	Determines how the processing time and process costs will be allocated to the entity. The process may be considered to be **Value Added**, **Non-Value Added**, **Transfer**, **Wait**, or **Other** and the associated cost will be added to the appropriate category for the entity and process.
Minimum	Parameter field for specifying the minimum value for either a uniform or triangular distribution.
Value	Parameter field for specifying the **mean** for a normal distribution, the **value** for a constant time delay, or the **mode** for a triangular distribution.
Maximum	Parameter field for specifying the maximum value for either a uniform or triangular distribution.

Prompt	Description
Std Dev	Parameter field for specifying the standard deviation for a normal distribution.
Expression	Parameter field for specifying an expression whose value is evaluated and used for the processing time delay.
Report Statistics	Specifies whether or not statistics will be automatically collected and stored in the report database for this process.

Process module — Resource dialog box

PROMPTS

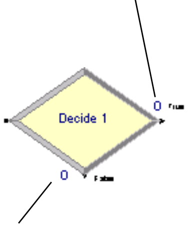
Prompt	Description
Type	Specification of a particular resource, or selecting from a pool of resources (that is, a resource set).
Resource Name	Name of the resource that will be seized and/or released. Applies only when Type is Resource.
Set Name	Name of the resource set from which a member will be seized and/or released. Applies only when Type is Set
Quantity	Number of resources of a given name or from a given set that will be seized/released. For sets, this value specifies only the number of a selected resource that will be seized/released (based on the resource’s capacity), not the number of members of a set to be seized/released.
Selection Rule	Method of selecting among available resources in a set. Cyclical will cycle through available members (for example, 1st member–2nd member–3rd member–1st member–2nd member–3rd member). **Random** will randomly select a member. **Preferred Order** will always select the first available member (1st member, if available; then 2nd member, if available; then 3rd member, etc.). **Specific Member** requires an input attribute value to specify which member of the set (previously saved in the Save Attribute field). **Largest Remaining Capacity** and **Smallest Number Busy** are used for resources with multiple capacity. Applies only when Type is Set.

Prompt	Description
Save Attribute	Attribute name used to save the index number into the set of the member that is selected. This attribute can later be referenced with the Specific Member selection rule. Does not apply when Selection Rule is Specific Member. If Action is specified as Delay Release, the value specified defines which member (the index number) of the set to be released. If no attribute is specified, the entity will release the member of the set that was *last* seized.
Set Index	The index number into the set of the member requested. Applies only when Selection Rule is Specific Member. If Action is specified as Delay Release, the value specified defines which member (the index number) of the set is to be released.

Decide module

DESCRIPTION

Number of entities that have taken the True branch



Number of entities that have taken the False branch

This module allows for decision-making processes in the system. It includes options to make decisions based on one or more conditions (for example, if entity type is Gold Card) or based on one or more probabilities (for example, 75%, true; 25%, false). Conditions can be based on attribute values (for example, Priority), variable values (for example, Number Denied), the entity type, or an expression (for example, NQ(ProcessA.Queue)).

There are two exit points out of the Decide module when its specified type is either 2-way by Chance or 2-way by Condition. There is one exit point for “true” entities and one for “false” entities. When the N-way by Chance or by Condition type is specified, multiple exit points are shown for each condition or probability and a single “else” exit. The number of entities that exit from each type (true/false) is displayed for
2-way by Chance or by Condition modules only.

TYPICAL USES

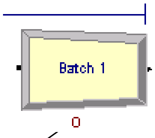
- Dispatching a faulty part for rework
- Branching accepted vs. rejected checks
- Sending priority customers to a dedicated process

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Type	Indicates whether the decision is based on a condition (if X>Y) or by chance/percentage (for example, 60%, yes; 40%, no). The type can be specified as either 2-way or N-way. 2-way allows for one condition or probability (plus the "false" exit). N-way allows for any number of conditions or probabilities to be specified as well as an "else" exit.
Conditions	Defines one or more conditions used to direct entities to different modules. Applies only when Type is N-way by Condition.
Percentages	Defines one or more percentages used to direct entities to different modules. Applies only when Type is N-way by Chance.
Percent True	Value that will be checked to determine the percentage of entities sent out a given True exit.
If	Types of conditions that are available for evaluation: Variable, Variable Array (1D), Variable Array (2D), Attribute, Entity Type, Expression.
Named	Specifies the name of the variable, attribute, or entity type that will be evaluated when an entity enters the module. Does not apply when Type is Expression.
Is	Evaluator for the condition. Applies only to Attribute and Variable conditions.
Row	Specifies the row index for a variable array. Applies only when Type is N-way by Condition or 2-way by Condition and Variable is Array 1-D or Array 2-D.
Column	Specifies the column index for a variable array. Applies only when Type is N-way by Condition or 2-way by Condition and Variable is Array 1-D or Array 2-D.
Value	Expression that will be either compared to an attribute or variable or that will be evaluated as a single expression to determine if it is true or false. Does not apply to Entity Type condition. If Type is Expression, this value must also include the evaluator (for example, Color<>Red).

Batch module

DESCRIPTION



Number of entities waiting to be batched

This module is intended as the grouping mechanism within the simulation model. Batches can be permanently or temporarily grouped. Temporary batches must later be split using the Separate module.

Batches may be made with any specified number of entering entities or may be matched together based on an attribute. Entities arriving at the Batch module are placed in a queue until the required number of entities has accumulated. Once accumulated, a new representative entity is created.

TYPICAL USES

- Collect a number of parts before starting processing
- Reassemble previously separated copies of a form
- Bring together a patient and his record before commencing an appointment

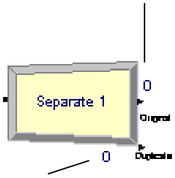
PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Type	Method of batching entities together.
Batch Size	Number of entities to be batched.
Save Criterion	Method for assigning representative entity’s user-defined attribute values.
Rule	Determines how incoming entities will be batched. **Any Entity** will take the first “Batch Size” number of entities and put them together. **By Attribute** signifies that the values of the specified attribute must match for entities to be grouped. For example, if Attribute Name is Color, all entities must have the same Color value to be grouped; otherwise, they will wait at the module for additional incoming entities.
Attribute Name	Name of the attribute whose value must match the value of the other incoming entities in order for a group to be made. Applies only when Rule is By Attribute.
Representative Entity	The entity type for the representative entity.

Separate module

Description

The original leaves on this branch



Duplicates leave on this branch

This module can be used to either copy an incoming entity into multiple entities or to split a previously batched entity. Rules for allocating costs and times to the duplicate are also specified. Rules for attribute assignment to member entities are specified as well.

When splitting existing batches, the temporary representative entity that was formed is disposed and the original entities that formed the group are recovered. The entities proceed sequentially from the module in the same order in which they originally were added to the batch.

When duplicating entities, the specified number of copies is made and sent from the module. The original incoming entity also leaves the module.

Typical uses

- Send individual entities to represent boxes removed from a container
- Send an order both to fulfillment and billing for parallel processing
- Separate a previously batched set of documents

Prompts

Prompt	**Description**
Name	Unique module identifier displayed on the module shape.
Type	Method of separating the incoming entity. **Duplicate Original** will simply take the original entity and make some number of identical duplicates. **Split Existing Batch** requires that the incoming entity be a temporarily batched entity using the Batch module. The original entities from the batch will be split.
Percent Cost to Duplicates	Allocation of costs and times of the incoming entity to the outgoing duplicates. This value is specified as a percentage of the original entity's costs and times (between 0-100). The percentage specified will be split evenly between the duplicates, while the original entity will retain any remaining cost/time percentage. Visible only when Type is Duplicate Original.
# of Duplicates	Number of outgoing entities that will leave the module, in addition to the original incoming entity. Applies only when Type is Duplicate Original.

Prompt	Description
Member Attributes	Method of determining how to assign the representative entity attribute values to the original entities.These options relate to six of the special-purpose attributes (Entity.Type, Entity.Picture, Entity.Sequence, Entity.Station, Entity.Jobstep, and Entity.HoldCostRate) and all user-defined attributes. Applies only when Type is Split Existing Batch.
Attribute Name	Name of representative entity attribute(s) that are assigned to original entities of the group. Applies only when Member Attributes is Take Specific Representative Values.

Assign module

DESCRIPTION

This module is used for assigning new values to variables, entity attributes, entity types, entity pictures, or other system variables. Multiple assignments can be made with a single Assign module.

TYPICAL USES

- Accumulate the number of subassemblies added to a part
- Change an entity’s type to represent the customer copy of a multi-page form
- Establish a customer’s priority

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Assignments	Specifies the one or more assignments that will be made when an entity executes the module.
Type	Type of assignment to be made. Other can include system variables, such as resource capacity or simulation end time.
Variable Name	Name of the variable that will be assigned a new value when an entity enters the module. Applies only when Type is Variable, Variable Array (1D), or Variable Array (2D).
Row	Specifies the row index for a variable array.
Column	Specifies the column index for a variable array.

Prompt	Description
Attribute Name	Name of the entity attribute that will be assigned a new value when the entity enters the module. Applies only when Type is Attribute.
Entity Type	New entity type that will be assigned to the entity when the entity enters the module. Applies only when Type is Entity Type.
Entity Picture	New entity picture that will be assigned to the entity when the entity enters the module. Applies only when Type is Entity Picture.
Other	Identifies the special system variable that will be assigned a new value when an entity enters the module. Applies only when Type is Other.
New Value	Assignment value of the attribute, variable, or other system variable. Does not apply when Type is Entity Type or Entity Picture.

Record module

Description

This module is used to collect statistics in the simulation model. Various types of observational statistics are available, including time between exits through the module, entity statistics (such as time or costing), general observations, and interval statistics (from some time stamp to the current simulation time). A count type of statistic is available as well. Tally and Counter sets can also be specified.

Typical uses

- Collect the number of jobs completed each hour
- Count how many orders have been late being fulfilled
- Record the time spent by priority customers in the main check-out line

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Type	Type of observational (tally) or count statistic to be generated. **Count** will increase or decrease the value of the named statistic by the specified value. **Entity Statistics** will generate general entity statistics, such as time and costing/duration information. **Time Interval** will calculate and record the difference between a specified attribute's value and current simulation time. **Time Between** will track and record the time between entities entering the module. **Expression** will record the value of the specified expression.
Attribute Name	Name of the attribute whose value will be used for the interval statistics. Applies only when Type is Interval.
Value	Value that will be recorded to the observational statistic when Type is Expression or added to the counter when Type is Count.
Tally Name	This field defines the symbol name of the tally into which the observation is to be recorded. Applies only when Type is Time Interval, Time Between, or Expression.
Counter	This field defines the symbol name of the counter to *Name* increment/decrement. Applies only when Type is Counter.
Record into Set	Check box to specify whether or not a tally or counter set will be used.
Tally Set Name	Name of the tally set that will be used to record the observational-type statistic. Applies only when Type is Time Interval, Time Between, or Expression.
Counter Set Name	Name of the counter set that will be used to record the count-type statistic. Applies only when Type is Count.
Set Index	Index into the tally or counter set.

Data modules

Data modules are the set of objects in the spreadsheet view of the model that define the characteristics of various process elements, such as resources and queues.

Attribute module

DESCRIPTION

This data module is used to define an attribute's dimension, data type and initial value(s). An attribute is a characteristic of all entities created, but with a specific value that can differ from one entity to another. Attributes can be referenced in other modules (for example, the Decide module), can be reassigned a new value with the Assign module, and can be used in any expression. Attribute values are unique for each entity, as compared to Variables which are global to the simulation module.

There are three methods for manually editing the Initial Values of an Attribute module:

- Using the standard spreadsheet interface. In the module spreadsheet, right-click on the **Initial Values** cell and select the **Edit via spreadsheet** menu item. The values for two-dimensional arrays should be entered one column at a time. Array elements not explicitly assigned are assumed to have the last entered value.
- Using the module dialog box. In the module spreadsheet, right-click on any cell and select the **Edit via dialog** menu item. The values for two-dimensional arrays should be entered one column at a time. Array elements not explicitly assigned are assumed to have the last entered value.
- Using the two-dimensional (2-D) spreadsheet interface. In the module spreadsheet, click on the **Initial Values** cell.

TYPICAL USES

- Due date of an order (entity)
- Priority of an order (entity)
- Color of a part (entity)

PROMPTS

Prompt	Description
Name	The unique name of the attribute being defined.
Rows	Number of rows in a one- or two-dimensional attribute.
Columns	Number of columns in a two-dimensional attribute.
Data Type	The data type of the values stored in the attribute. Valid types are **Rea**l and **String**. The default type is **Real**.
Initial Values	Lists the initial value or values of the attribute. You can assign new values to the attribute by using the Assign module.
Initial Value	Entity attribute value when entity is created and enters the system.

Entity module

DESCRIPTION

This data module defines the various entity types and their initial picture values in a simulation. Initial costing information and holding costs are also defined for the entity.

TYPICAL USES

- Items being produced or assembled (parts, pallets)
- Documents (forms, e-mails, faxes, reports)
- People moving through a process (customers, callers)

PROMPTS

Prompt	Description
Entity Type	The name of the entity type being defined. This name must be unique.
Initial Picture	Graphical representation of the entity at the start of the simulation. This value can be changed during the simulation using the Assign module.
Holding Cost/Hour	Hourly cost of processing the entity through the system. This cost is incurred when the entity is anywhere in the system.

Prompt	Description
Initial VA Cost	Initial cost value that will be assigned to the value-added cost attribute of the entity. This attribute accrues the costs incurred when an entity is spending time in a value-added activity.
Initial NVA Cost	Initial cost value that will be assigned to the non-value-added cost attribute of the entity. This attribute accrues the costs incurred when an entity is spending time in a non-value-added activity.
Initial Waiting Cost	Initial cost value that will be assigned to the waiting-cost attribute of the entity. This attribute accrues the costs incurred when an entity is spending time in a wait activity; for example, waiting to be batched or waiting for resource(s) at a Process module.
Initial Transfer Cost	Initial cost value that will be assigned to the transfer cost attribute of the entity. This attribute accrues the costs incurred when an entity is spending time in a transfer activity.
Initial Other Cost	Initial cost value that will be assigned to the other cost attribute of the entity. This attribute accrues the costs incurred when an entity is spending time in another activity.
Report Statistics	Specifies whether or not statistics will be collected automatically and stored in the report database for this entity type.

Queue module

Description

This data module may be utilized to change the ranking rule for a specified queue. The default ranking rule for all queues is First In, First Out unless otherwise specified in this module. There is an additional field that allows the queue to be defined as shared.

Typical uses

- Stack of work waiting for a resource at a Process module
- Holding area for documents waiting to be collated at a Batch module

Prompts

Prompt	Description
Name	The name of the queue whose characteristics are being defined. This name must be unique.
Type	Ranking rule for the queue, which can be based on an attribute. Types include **First In, First Out**; **Last In, First Out**; **Lowest Attribute Value** (first); and **Highest Attribute Value** (first). A low attribute value would be 0 or 1, while a high value may be 200 or 300.
Attribute Name	Attribute that will be evaluated for the Lowest Attribute Value or Highest Attribute Value types. Entities with lowest or highest values of the attribute will be ranked first in the queue, with ties being broken using the First In, First Out rule.
Shared	Check box that determines whether a specific queue is used in multiple places within the simulation model. Shared queues can only be used for seizing resources (for example, with the Seize module from the Advanced Process panel).
Report Statistics	Specifies whether or not statistics will be collected automatically and stored in the report database for this queue.

Resource module

Description

This data module defines the resources in the simulation system, including costing information and resource availability. Resources may have a fixed capacity that does not vary over the simulation run or may operate based on a schedule. Resource failures and states can also be specified in this module.

Typical uses

- Equipment (machinery, cash register, phone line)
- People (clerical, order processing, sales clerks, operators)

PROMPTS

Prompt	Description
Name	The name of the resource whose characteristics are being defined. This name must be unique.
Type	Method for determining the capacity for a resource. **Fixed Capacity** will not change during the simulation run. **Based on Schedule** signifies that a Schedule module is used to specify the capacity and duration information for the resource.
Capacity	Number of resource units of a given name that are available to the system for processing. Applies only when Type is Fixed Capacity.
Schedule Name	Identifies the name of the schedule to be used by the resource. The schedule defines the capacity of a resource for a given period of time. Applies only when type is Schedule.
Schedule Rule	Dictates when the actual capacity change is to occur when a decrease in capacity is required for a busy resource unit. Applies only when Type is Schedule.
Busy/Hour	Cost per hour of a resource that is processing an entity. The resource becomes busy when it is originally allocated to an entity and becomes idle when it is released. During the time when it is busy, cost will accumulate based on the busy/hour cost. The busy cost per hour is automatically converted to the appropriate base time unit specified within the Replication Parameters page of the **Run > Setup** menu item.
Idle/Hour	Cost per hour of a resource that is idle. The resource is idle while it is not processing an entity. During the time when it is idle, cost will accumulate based on the idle/hour cost. The idle cost per hour is automatically converted to the appropriate base time unit specified within the Replication Parameters page of the **Run > Setup** menu item.
Per Use	Cost of a resource on a usage basis, regardless of the time for which it is used. Each time the resource is allocated to an entity, it will incur a per-use cost.
StateSet Name	Name of states that the resource may be assigned during the simulation run.

Prompt	Description
Initial State	Initial state of a resource. If specified, the name must be defined within the repeat group of state names. This field is shown only when a StateSet Name is defined.
Failures	Lists all failures that will be associated with the resource. ■ *Failure Name*–Name of the failure associated with the resource. ■ *Failure Rule*–Behavior that should occur when a failure is to occur for a busy resource unit.
Report Statistics	Specifies whether or not statistics will be collected automatically and stored in the report database for this resource.

Variable module

DESCRIPTION

This data module is used to define a variable's dimension and values. You can reference variables in other modules (for example, the Decide module), reassign new values to variables with the Assign module, and use variables in any expression.

You can use an external data file to specify variable values, and you can specify the variable's initial values in the Variable module. If you use both methods, the values are read at different times, depending on the options you specify, including the File Read Time, the Clear Option, and the replication parameters you specify in the Run Setup dialog box. For more information, see the online Help.

There are three methods for manually editing the Initial Values of a Variable module:

- Using the standard spreadsheet interface. In the module spreadsheet, right-click on the Initial Values cell and select the Edit via spreadsheet menu item. The values for two-dimensional arrays should be entered one column at a time. Array elements not explicitly assigned are assumed to have the last entered value.
- Using the module dialog box. In the module spreadsheet, right-click on any cell and select the Edit via dialog menu item. The values for two-dimensional arrays should be entered one column at a time. Array elements not explicitly assigned are assumed to have the last entered value.
- Using the two-dimensional (2-D) spreadsheet interface. In the module spreadsheet, click on the Initial Values cell.

TYPICAL USES

- Number of documents processed per hour
- Serial number to assign to parts for unique identification
- Space available in a facility

PROMPTS

Prompt	Description
Name	The unique name of the variable being defined.
Rows	Number of rows in a one- or two-dimensional variable.
Columns	Number of columns in a two-dimensional variable.
Report Statistics	Check box for determining whether or not statistics will be collected. This field is visible when the rows and columns are not specified (that is, for single variables).
Data Type	The data type of the values stored in the variable. Valid types are **Rea**l and **String**. The default type is **Real**.
Clear Option	Defines the time (if at all) when the value(s) of the variable is reset to the initial value(s) specified. Specifying **Statistics** resets this variable to its initial value(s) whenever statistics are cleared. Specifying **System** resets this variable to its initial value(s) whenever the system is cleared. Specifying **None** indicates that this variable is never reset to its initial value(s), except prior to the first replication.
File Name	Name of the file from which to read the variable's value or values. You can use any file access type supported by Arena except sequential text files and Lotus spreadsheet (.wks) files. If the file name you specify has not been created yet, Arena will create it, but you must edit the file to specify the file access type, path, and recordset (if required).
Recordset	Name of the recordset in the specified file from which to read values. This field is available only if you specify a File Name for a file that has been set up with a file access type, path, and recordset. Arena uses the Rows and Columns properties to determine the amount of data to read from the recordset. A recordset is required for all file types except .xml. The recordset size must be equal to or greater than the number of rows and columns specified for the variable.

Prompt	Description
File Read Time	Specifies when to read the values from the file into the variable. If you select **PreCheck**, the values for the variable are read while the model is still in Edit mode (prior to the model being checked and compiled). If you select **BeginSimulation**, values are read when the model is compiled, prior to the first replication. If you select **BeginReplication**, values are read prior to each replication.
Initial Values	Lists the initial value or values of the variable. You can assign new values to the variable at different stages of the model by using the Assign module.
Initial Value	Variable value at the start of the simulation.

Schedule module

DESCRIPTION

This data module may be used in conjunction with the Resource module to define an operating schedule for a resource or with the Create module to define an arrival schedule. Additionally, a schedule may be used and referenced to factor time delays based on the simulation time.

TYPICAL USES

- Work schedule for staff, including breaks
- Breakdown patterns for equipment
- Volume of customers arriving at a store
- Learning-curve factors for new workers

PROMPTS

Prompt	Description
Name	The name of the schedule being defined. This name must be unique.
Type	Type of schedule being defined. This may be **Capacity-**related (for resource schedules), **Arrival-**related (for the Create module), or **Other** (miscellaneous time delays or factors).

Prompt	Description
Time Units	Time units used for the time-duration information.
Scale Factor	Method of scaling the schedule for increases or decreases in Arrival/Other values. The specified Value fields will be multiplied by the scale factor to determine the new values. Not available for Capacity-type schedules.
Durations	Lists the value and duration pairs for the schedule. Values can be capacity, arrival, or other type values, while the duration is specified in time units. Schedule pairs will repeat after all durations have been completed, unless the last duration is left blank (infinite). Schedule data can be entered graphically using the graphical schedule editor or manually using the Value/ Duration fields.
Value	Represents either the capacity of a resource (if Type is Capacity), arrival rate (if Type is Arrival), or some other value (if Type is Other). Examples of Other may be a factor that is used in a delay expression to scale a delay time during various parts of the day.
Duration	Time duration for which a specified Value will be valid.

Set module

Description

This data module defines various types of sets, including resource, counter, tally, entity type, and entity picture. Resource sets can be used in the Process modules (and Seize, Release, Enter, and Leave of the Advanced Process and Advanced Transfer panels). Counter and Tally sets can be used in the Record module. Queue sets can be used with the Seize, Hold, Access, Request, Leave, and Allocate modules of the Advanced Process and Advanced Transfer panels.

Typical uses

- Machines that can perform the same operations in a manufacturing facility
- Supervisors, check-out clerks in a store
- Shipping clerks, receptionists in an office
- Set of pictures corresponding to a set of entity types

PROMPTS

Prompt	Description
Name	The unique name of the set being defined.
Type	Type of set being defined.
Members	Repeat group that specifies the resource members with the set. The order of listing the members within the repeat group is important when using selection rules such as Preferred Order and Cyclical.
Resource Name	Name of the resource to include in the resource set. Applies only when Type is Resource.
Tally Name	Name of the tally within the tally set. Applies only when Type is Tally.
Counter Name	Name of the counter within the counter set. Applies only when Type is Counter.
Entity Type	Name of the entity type within the entity type set. Applies only when Type is Entity.
Picture Name	Name of the picture within the picture set. Applies only when Type is Entity Picture.

4 The Advanced Process Panel

This chapter presents the modules included on the Advanced Process panel, where you will find additional functionality for modeling your process. Example uses of each module are included in this section.

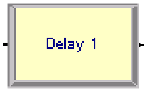
The Advanced Process panel is accessible to licensed editions of Arena Standard and Professional.

Flowchart modules

Flowchart modules are the set of objects that are placed in the model window to describe the simulation process.

Delay module

Description



The Delay module delays an entity by a specified amount of time.

When an entity arrives at a Delay module, the time delay expression is evaluated and the entity remains in the module for the resulting time period. The time is then allocated to the entity's value-added, non-value added, transfer, wait, or other time. Associated costs are calculated and allocated as well.

Typical uses

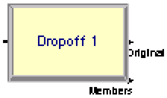
- Processing a check at a bank
- Performing a setup on a machine
- Transferring a document to another department

Prompts

Prompt	**Description**
Name	Unique module identifier displayed on the module shape.
Allocation	Type of category to which the entity's incurred delay time and cost will be added.
Delay Time	Determines the value of the delay for the entity.
Units	Time units used for the delay time.

Dropoff module

DESCRIPTION



The Dropoff module removes a specified number of entities from the entity's group and sends them to another module, as specified by a graphical connection. Group user-defined attribute value and internal attributes may be given to the dropped-off entities based on a specified rule.

TYPICAL USES

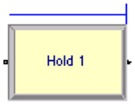
- Loading shelves with product
- Separating a form for use in various departments

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Quantity	Number of entities that will be dropped off from an incoming representative grouped entity.
Starting Rank	Starting rank of the entities to be dropped off, based on the entities in the group.
Member Attributes	Method of determining how to assign the representative entity attribute values (other than costs/times) to the dropped-off original entities.
Attribute Name	Name of representative entity attribute(s) assigned to dropped-off original entities of the group.

Hold module

DESCRIPTION



This module will hold an entity in a queue to wait for a signal, wait for a specified condition to become true (scan), or be held infinitely (to be removed later with the Remove module).

If the entity is holding for a signal, the Signal module is used elsewhere in the model to allow the entity to move on to the next module. If the entity is holding for a given condition to be true, the entity will remain at the module (either in a defined or internal queue) until the condition(s) becomes true. When the entity is in an infinite hold, the Remove module is used elsewhere in the model to allow the entity to continue processing.

TYPICAL USES

- Waiting for a traffic light to turn green
- Holding a part for authorization
- Checking the status of a machine or operator to continue a process

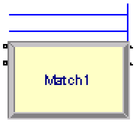
PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Type	Indicates the reasoning for holding the entity within a specified or internal queue. Wait for Signal will hold the entity until a signal of the same value is received. Scan for Condition will hold the entity until the specified condition becomes true. Infinite Hold will hold the entity until it is removed from the queue by a Remove module.
Wait for Value	Signal code for the waiting entity. Applies only when Type is Wait for Signal.
Limit	Maximum number of waiting entities that will be released upon receipt of a signal. Applies only when Type is Wait for Signal.
Condition	Specifies the condition that will be evaluated to hold the entity at the module. If the condition is evaluated to true, the entity leaves the module immediately. If the condition is false, the entity will wait in the associated queue until the condition becomes true. Applies only when Type is Scan for Condition.
Queue Type	Determines the type of queue used to hold the entities. If Queue is selected, the queue name is specified. If Set is selected, the queue set and member in the set are specified. If Internal is selected, an internal queue is used to hold all waiting entities. Attribute and Expression are additional methods for defining the queue to be used.
Queue Name	This field is visible only if Queue Type is Queue, and it defines the symbol name of the queue.
Set Name	This field is visible only if Queue Type is Set, and it defines the queue set that contains the queue being referenced.

Prompt	Description
Set Index	This field is visible only if Queue Type is Set, and it defines the index into the queue set. Note that this is the index into the set and not the name of the queue in the set. For example, the only valid entry for a queue set containing three members is an expression that evaluates to 1, 2, or 3.
Attribute	This field is visible only if Queue Type is Attribute. The attribute entered in this field will be evaluated to indicate which queue is to be used.
Expression	This field is visible only if Queue Type is Expression. The expression entered in this field will be evaluated to indicate which queue is to be used.

Match module

DESCRIPTION



The Match module brings together a specified number of entities waiting in different queues. The match may be accomplished when there is at least one entity in each of the desired queues. Additionally, an attribute may be specified such that the entities waiting in the queues must have the same attribute values before the match is initiated.

When an entity arrives at the Match module, it is placed in one of up to five associated queues, based on the entry point to which it is connected. Entities will remain in their respective queues until a match exists.

Once a match exists, one entity from each queue is released to be matched. The matched entities are then synchronized to depart from the module.

TYPICAL USES

- Assembling a part
- Gathering various products for a customer order
- Synchronizing a customer exit with a filled order

Prompts

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Number to Match	Number of matching entities that must reside in different queues before a match may be completed.
Type	Method for matching the incoming entities. If Type is Any Entities, one entity must reside in each queue for a match to be made. If Type is Based on Attribute, one entity must reside in each queue with the same attribute value.
Attribute Name	Attribute name that is used for identifying an arriving entity's match value. Applies only when Type is Based on Attribute.

Pickup module

Description

The Pickup module removes a number of consecutive entities from a given queue starting at a specified rank in the queue. The entities that are picked up are added to the end of the incoming entity's group.

Typical uses

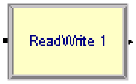
- Gathering an order from various queue locations
- Gathering completed forms for an office order
- Picking up students at a bus stop for school

Prompts

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Quantity	Number of entities to pick up.
Queue Name	Name of the queue from which the entities will be picked up, starting at the specified rank.
Starting Rank	Starting rank of the entities to pick up from the queue, Queue Name.

ReadWrite module

DESCRIPTION



The ReadWrite module is used to read data from an input file or the keyboard and assign the data values to a list of variables or attributes (or other expression). This module is also used to write data to an output device, such as the screen or a file.

When reading from or writing to a file, the ReadWrite logic varies according to the *Type* of read or write action performed and according to the *Access Type* of the *Arena File Name* (the *Access Type* of a file is specified in the File module).

Sequential File or LOTUS Spreadsheet Access Types. When an entity arrives at the ReadWrite module, the specified file is examined to see if it is open (active). If not, the file is automatically opened. The values of the attributes, variables, or expressions (other) listed are read or written according to the format specified.

If you plan to import Lotus (*.wks*) files into Excel, see the online help for guidelines.

Microsoft Excel, Microsoft Access, and ActiveX Data Objects Access Types. When an entity arrives at the ReadWrite module, the specified file is examined to see if an ADO connection to the file is open (active). If the file is currently closed, then an ADO connection is automatically opened to the data source using either the Microsoft Jet OLE DB Provider (if the *Access Type* is *Microsoft Excel, Microsoft Excel 2007, Microsoft Access,* or *Microsoft Access 2007*) or the specified connection string (if the *Access Type* is *ActiveX Data Objects*).

The *Recordset ID* is then examined to see if that recordset in the file is open. A recordset refers to a table of data in the data store that is organized in columns (fields) and rows (records).

If the recordset is currently closed, then it is automatically opened using the recordset definition specified in the File module (for example, the specified named range or table name). The recordset then remains open until the file is closed.

The values of the attributes, variables, or expressions (other) listed in the Assignments repeat group are then read from or written to the specified *Record Number* (that is, row) in the recordset. The first entry in *Assignments* is read from or written to the first field in the record, the second entry corresponds to the second field, and so on.

Extensible Markup Language Access Type. When an entity arrives at the ReadWrite module, the specified file is examined to see if it is open (active). If not, the file is automatically opened into a single ADO recordset.

The values of the attributes, variables, or expressions (other) listed are then read from or written to the specified *Record Number* (row) in the recordset.

Typical uses

- Reading in airline arrival/departure information
- Prompting an end user for model configuration data (menu)
- Writing order information to a data file, such as order arrival, order fill time, order completion time

Prompts

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Type	Method of reading or writing used. Data may be read from a file or from the keyboard. Data may be written to a file or to the screen.
Arena File Name	Name used to identify the file within the File module. Applies only when reading from or writing to a file.
Overriding File Format	Format for writing or reading the data. This format will override any format specified in the structure field of the File module (for files of the Sequential File Access Type). Enter a FORTRAN-style or C-style format (enclosed in double quotes) to exactly describe the type and location of each field. Or enter the keyword FREE to specify a list-directed format whereby each field will be assumed to be separated by a comma or one or more blanks. A FORTRAN or C format (enclosed in double quotes) may be used to describe the type and location of each field.
Recordset ID	Identifies the recordset in Arena File Name that will be accessed. Applies only when the file *Access Type* (specified in the File module) is Microsoft Excel, Microsoft Excel 2007, Microsoft Access, Microsoft Access 2007, or ActiveX Data Objects.
Record Number Access Type	Record number (that is, row) in Recordset ID to read from or write to. Applies only when the file (specified in the File module) is Microsoft Excel, Microsoft Excel 2007, Microsoft Access, Microsoft Access 2007, ActiveX Data Objects, or Extensible Markup Language.
Variable Type	Type of information that will be read or written.
Attribute Name	Defines the symbol name of the attribute to be written or read.

Prompt	Description
Variable Name	Defines the symbol name of the variable to be read or written.
Row	Specifies the row index number for a variable array.
Column	Specifies the column index number for a variable array.
Other	Defines the expression for reading or writing other types of information.

Release module

DESCRIPTION

The Release module is used to release units of a resource that an entity previously has seized. This module may be used to release individual resources or may be used to release resources within a set. For each resource to be released, the name and quantity to release are specified.

When the entity enters the Release module, it gives up control of the specified resource(s). Any entities waiting in queues for those resources will gain control of the resources immediately.

TYPICAL USES

- Finishing a customer order (release the operator)
- Completing a tax return (release the accountant)
- Leaving the hospital (release the doctor, nurse, hospital room)

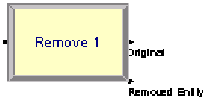
PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Type	Type of resource for releasing, either specifying a particular resource, or selecting from a pool of resources (that is, a resource set). The resource name may also be specified by an expression or attribute value.
Resource Name	Name of the resource that will be released.
Set Name	Name of the resource set from which a member will be released.

Prompt	Description
Attribute Name	Name of the attribute that specifies the resource name to be released.
Expression	Name of the expression that specifies the name of the resource to be released.
Quantity	Number of resources of a given name or from a given set that will be released. For sets, this value specifies only the number of a selected resource that will be released (based on the resource's capacity), not the number of members to be released within the set.
Release Rule	Method of determining which resource within a set to release. Last Member Seized and First Member Seized will release the last/first member from within the set that was seized. Specific member indicates that a member number or attribute (with a member number value) will be used to specify the member to release.
Set Index	Member index of the resource set that the entity will release.

Remove module

DESCRIPTION



The Remove module removes a single entity from a specified position in a queue and sends it to a designated module.

When an entity arrives at a Remove module, it removes the entity from the specified queue and sends it to the connected module. The rank of the entity signifies the location of the entity within the queue. The entity that caused the removal proceeds to the next module specified and is processed before the removed entity.

TYPICAL USES

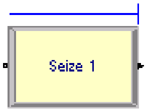
- Removing an order from a queue that is due to be completed next
- Calling a patient from a waiting room for an examination
- Retrieving the next order to be processed from a pile of documents

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Queue Name	Name of the queue from which the entity will be removed.
Rank of Entity	Rank of the entity to remove from within the queue.

Seize module

DESCRIPTION



The Seize module allocates units of one or more resources to an entity. The Seize module may be used to seize units of a particular resource, a member of a resource set, or a resource as defined by an alternative method, such as an attribute or expression.

When an entity enters this module, it waits in a queue (if specified) until all specified resources are available simultaneously. Allocation type for resource usage is also specified.

TYPICAL USES

- Beginning a customer order (seize the operator)
- Starting a tax return (seize the accountant)
- Being admitted to hospital (seize the hospital room, nurse, doctor)

PROMPTS

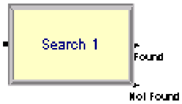
Prompt	Description
Name	Unique module identifier displayed on the module shape.
Allocation	Determines to which category the resource usage cost will be allocated for an entity going through the Seize module.
Priority	Priority value of the entity waiting at this module for the resource(s) specified if one or more entities from other modules are waiting for the same resource(s).
Type	Type of resource for seizing, either specifying a particular resource, or selecting from a pool of resources (that is, a resource set). The name of the resource may also be specified as an attribute value or within an expression.

Prompt	Description
Resource Name	Name of the resource that will be seized.
Set Name	Name of the resource set from which a member will be seized.
Attribute Name	Name of the attribute that stores the resource name to be seized.
Expression	Expression that evaluates to a resource name to be seized.
Quantity	Number of resources of a given name or from a given set that will be seized. For sets, this value specifies only the number of a selected resource that will be seized (based on the resource's capacity), not the number of members to be seized within the set.
Selection Rule	Method of selecting among available resources in a set. **Cyclical** will cycle through available members (for example, 1-2-3-1-2-3). **Random** will randomly select a member. **Preferred Order** will always select the first available member (for example, 1, if available; then 2, if available; then 3). **Specific Member** requires an input attribute value to specify which member of the set (previously saved in the Save Attribute field). **Largest Remaining Capacity** and **Smallest Number Busy** are used for resources with multiple capacity.
Save Attribute	Attribute name used to store the index number into the set of the member that is chosen. This attribute can later be referenced with the Specific Member selection rule.
Set Index	Index value into the set that identifies the number into the set of the member requested. If an attribute name is used, the entity must have a value for the attribute before utilizing this option.
Resource State	State of the resource that will be assigned after the resource is seized. The resource state must be defined with the Resource module.
Queue Type	Determines the type of queue used to hold the entities while waiting to seize the resource(s). If Queue is selected, the queue name is specified. If Set is selected, the queue set and member in the set are specified. If Internal is selected, an internal queue is used to hold all waiting entities. Attribute and Expression are additional methods for defining the queue to be used.
Queue Name	This field is visible only if Queue Type is Queue, and it defines the symbol name of the queue.

Prompt	Description
Set Name	This field is visible only if Queue Type is Set, and it defines the queue set that contains the queue being referenced.
Set Index	This field is visible only if Queue Type is Set, and it defines the index into the queue set. Note that this is the index into the set and not the name of the queue in the set. For example, the only valid entries for a queue set containing three members is an expression that evaluates to 1, 2, or 3.
Attribute	This field is visible only if Queue Type is Attribute. The attribute entered in this field will be evaluated to indicate which queue is to be used.
Expression	This field is visible only if Queue Type is Expression. The expression entered in this field will be evaluated to indicate which queue is to be used.

Search module

DESCRIPTION



The Search module searches a queue, a group (batch), or an expression to find the entity rank (for entities in a queue or group) or the value of the global variable *J* that satisfies the specified search condition. When searching a queue or group, the value of the global system variable *J* is set to the rank of the first entity that satisfies Search Condition, or to 0 if Search Condition is not satisfied. When searching an expression, the global system variable *J* is set to the value of the first index value that satisfies the search condition or to zero if no value of *J* in the specified range satisfies the search condition.

When an entity arrives at a Search module, the index *J* is set to the starting index and the search condition is then checked. If the search condition is satisfied, the search ends and the current value of *J* is retained. Otherwise, the value of *J* is increased or decreased and the condition is rechecked. This process repeats until the search condition is satisfied or the ending value is reached. If the condition is not met or there are no entities in the queue or group, *J* is set equal to 0.

TYPICAL USES

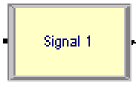
- Looking for a particular order number in a queue
- Searching a group for a certain part type
- Determining which process to enter based on availability of resources (search an expression)

Prompts

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Type	Determination of what will be searched. Search options include entities in a queue, entities within a group (batch) or some expression(s).
Queue Name	Name of the queue that will be searched. Applies only when the Type is Search a Queue.
Starting Value	Starting rank in the queue or group or starting value for *J* in an expression.
Ending Value	Ending rank in the queue or group or ending value for *J* in an expression.
Search Condition	Condition containing the index *J* for searching expressions or containing an attribute name(s) for searching queues or batches.

Signal module

Description



The Signal module sends a signal value to each Hold module in the model set to Wait for Signal and releases the maximum specified number of entities.

When an entity arrives at a Signal module, the signal is evaluated and the signal code is sent. At this time, entities at Hold modules that are waiting for the same signal are removed from their queues. The entity sending the signal continues processing until it encounters a delay, enters a queue, or is disposed.

Typical uses

- Analyzing traffic patterns at an intersection (signal when the light turns green)
- Signaling an operator to complete an order that was waiting for a component part

Prompts

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Signal Value	Value of the signal to be sent to entities in Hold modules.
Limit	Maximum number of entities that are to be released from any Hold modules when the signal is received.

Store module

DESCRIPTION

The Store module adds an entity to storage. The Unstore module may then be used to remove the entity from the storage.

When an entity arrives at the Store module, the storage specified is incremented, and the entity immediately moves to the next module in the model.

Storages are useful for displaying entity animation while an entity undergoes processing in other modules. Additionally, statistics may be kept on the number of entities in storage.

TYPICAL USES

- Animating a part through a number of delay operations (load, setup, process, unload)
- Tracking the number of customers within a grocery store (place in storage upon entry)

PROMPTS

Prompt	**Description**
Name	Unique module identifier displayed on the module shape.
Type	Method of specifying the storage name as a **Storage**, **Set**, **Attribute**, or **Expression**.
Storage Name	Name of the storage to which the entity will be added. Applies only when the Type is Storage.
Set Name	Name of the storage set from which the storage is to be selected. Applies only when the Type is Set.
Set Index	Index into the defined storage set that contains the desired storage name. Applies only when the Type is Set.
Attribute	Name of the attribute whose value contains the storage. Applies only when the Type is Attribute.
Expression	Expression that is evaluated to the storage into which the entity is placed. Applies only when the Type is Expression.

Unstore module

Description

The Unstore module removes an entity from storage. When an entity arrives at the Unstore module, the storage specified is decreased and the entity immediately moves to the next module in the model.

Typical uses

- Removing the entity from an animation location when processing is complete
- Tracking the number of customers within a grocery store (unstore upon exit)

Prompts

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Type	Method of specifying the storage name as a **Storage**, **Set**, **Attribute**, or **Expression**. Default will remove an entity from the last storage that it entered.
Storage Name	Name of the storage to which the entity will be added. Applies only when the Type is Storage.
Set Name	Name of the storage set from which the storage is to be selected. Applies only when the Type is Set.
Set Index	Index into the defined storage set that contains the desired storage name. Applies only when the Type is Set.
Attribute	Name of the attribute whose value contains the storage. Applies only when the Type is Attribute.
Expression	Expression that is evaluated to the storage into which the entity is placed. Applies only when the Type is Expression.

Adjust Variable module

Description

The Adjust Variable module adjusts a variable to a target value at a specified rate. This module might be used with the *Rotate By Expression* option in Global and Resource pictures to present smooth animated rotations of pictures. It might also be used to approximate/ animate a continuous increase or decrease to a variable value over time.

When an entity enters an Adjust Variable module, the *Variable Name* is adjusted to the *To Value* at the specified *Rate*. The entity is held in the module until the adjustment is completed.

The *Update Interval* specifies the time interval between actual assignment updates of the variable value over the time period of the change. A smaller update interval produces smoother animation in plots or rotating pictures that reference the variable, as well as more accurate time-persistent statistics collected on the variable. A larger update interval produces faster run speeds.

TYPICAL USES

- Rotate a resource picture to 180 degrees at a rate of 5 degrees per second

PROMPTS

Prompt	**Description**
Name	Unique module identifier. This name is displayed on the module shape.
Variable Type	Specifies the array characteristics of the variable being adjusted.
Row	Specifies the row index if the variable is an array.
Column	Specifies the column index if the variable is a 2-D array.
Variable Name	Specifies the name of the variable to adjust.
To Value	Specifies the target value that the variable will be adjusted to.
Rate	Specifies the rate at which the adjustment will be made.
Units	The time units of the rate.
Update Interval	Specifies the interval between updates of the variable’s value as the variable is adjusted at the specified rate.
Units	The time units of the update interval.
Allocation	Type of category to which the entity’s incurred delay time and cost in the module will be added.

Data modules

Data modules are the set of objects in the spreadsheet view of the model that define the characteristics of various process elements, such as resources and queues.

Advanced Set module

DESCRIPTION

The Advanced Set module specifies queue sets, storage sets, and other sets and their respective members. A set defines a group of similar elements that may be referenced from a common name and a set index. The elements that make up the set are referred to as the members of the set.

Queue sets may be specified within a Seize or material-handling-type module. Storage sets may be used in the Store and Unstore module. Other sets may include sets of attributes or other miscellaneous elements.

TYPICAL USES

- Various checkout queues in a grocery store (queues)
- Animation locations for performing operations (storages)

PROMPTS

Prompt	Description
Name	The name of the advanced set whose members are being defined. This name must be unique.
Set Type	Type of set that is being defined, which may include **Queue**, **Storage**, or **Other** types of sets.
Queue Name	Name of the queues that are included within a queue set.
Storage Name	Name of the storages that are included within a storage set.
Other	Name of the members that are included within an “other” type set.

Expression module

DESCRIPTION

The Expression module defines expressions and their associated values. Expressions are referenced in the model by using their name. Expressions can optionally be specified as one- or two-dimensional arrays.

You can specify expression values within the Expression module, or you can use the Expression module to specify a file from which to obtain expression values. The file allows you to read values from an outside source.

An expression value may be formed using combinations of integers, real numbers, symbol names, statistical distributions (for example, NORM(10,2)), arithmetic operators (for example, +, *), parentheses, logical operators, (for example, .GT. or >), attributes, and variables. An expression value may be referenced in another expression, but recursive calls to the same expression are not allowed.

TYPICAL USES

- Distribution for an entity arrival
- Entity picture values
- Complex expression for order entry times

PROMPTS

Prompt	Description
Name	The name of the expression whose characteristics are being defined. This name must be unique.
Rows	Maximum number of rows in the defined expression.
Columns	Maximum number of columns in the defined expression. Visible only when the number of rows is specified.
Data Type	The data type of the values stored in the expression.: Real, String, or Native.
File Name	Name of the file from which to read the expression's value or values. You can use any file access type supported by Arena except sequential text files and Lotus spreadsheet (.wks) files. If the file name you specify has not been created yet, Arena will create it, but you must edit the file to specify the file access type, path, and recordset (if required).

Prompt	Description
Recordset	Name of the recordset in the specified file from which to read values. This field is available only if you specify a File Name with a file access type, path, and recordset. Arena uses the Rows and Columns properties to determine the amount of data to read from the recordset. A recordset is required for all file types except .xml. The recordset size must be equal to or greater than the number of rows and columns specified for the expression.
Expression Values	Lists the value or values of the expression. This property is not available if you specify a File Name from which to read expression values.
Expression Value	Expression value associated with the expression name.

Failure module

Description

The Failure module is designed for use with resources. When a failure occurs, the entire resource (regardless of its capacity) is failed. Failures are designed to be used with single-capacity resources or with multiple-capacity resources whose individual resource units all fail at the same time.

Typical uses

- Breakdown information for a machine
- Cash register tape refill every "x" customers
- Random computer shutdowns or restarts

Prompts

Prompt	Description
Name	The name of the failure associated with one or more resources.
Type	Determines if the failure is **time-based** or **count-based**.
Count	Defines the number of resource releases for count-based failures. Valid when the Type is Count.
Up Time	Defines the time between failures for time-based failures. Valid when the Type is Time.

Prompt	Description
Up Time Units	Time units for the time between failures (Up Time) for time-based failures.
Down Time	Defines the duration of the failure.
Down Time Units	Time units for the duration of the failure (Down Time).
Uptime in this State only	Defines the state that should be considered for the time between failures (only for time-based failures). If state is not specified, then all states are considered (that is, the time between failures does not depend on the time spent in a specific state, but rather on the total simulation time). For example, you might want to define a failure to be based only on the state Busy, and therefore, the time between downtimes would be based on the amount of time that a resource is busy, not simulated clock time.

File module

DESCRIPTION

Use the File module to access external files for the ReadWrite module, Variable module, and Expression module. The File module identifies the system file name and defines the access type and operational characteristics of the file.

TYPICAL USES

- File containing predefined airline flight data
- File specifying customer order times and relevant information
- File to write user model configuration data from menu input

PROMPTS

Prompt	Description
Name	The name of the file whose characteristics are being defined. This name must be unique.
Access Type	The file type.
Operating System File Name	Name of the actual file that is being read from or to which it is being written.

Prompt	Description
Connecting String	Connection string used to open ADO connection to the data source.
Structure	File structure, which can be unformatted, free format, or a specific C or FORTRAN format.
End of File Action	Type of action to occur if an end of file condition is reached.
Initialize Option	Action to be taken on file at beginning of each simulation replication.
Comment Character	Character indicating comment record.
Recordset Name	Name used to identify the recordset in the Expression, ReadWrite, and Variable modules. This name must be unique within the file. This field is available for Microsoft Excel, Microsoft Excel 2007, Microsoft Access, Microsoft Access 2007, and ActiveX Data Objects files.
CommandText	Text of the command that will be used to open the recordset (for example, SQL statement, procedure name, table name.) This field is available for ActiveX Data Object files only.
CommandType	Type of command entered in the CommandText.
Named Range	The named range in the Excel workbook to which the recordset refers.
Table Name	The name of the table in the Access database to which the recordset refers.

StateSet module

DESCRIPTION

The StateSet module is used to define states for a resource or number of resources. The states may be associated with an autostate or may be new states for the resource. The Resource module in the Basic Process panel references the stateset, if any, that a given resource will use.

TYPICAL USES

- States for an operator, such as busy with customer, busy refilling shelves, busy on phone
- States for a machine, such as random failure, scheduled failure, busy with setup, busy with processing
- States for a doctor, such as scheduled appointments, hospital rounds, paperwork, telephone calls

PROMPTS

Prompt	Description
StateSet Name	Name of set of states that the resource may be assigned during the simulation run.
State Name	Name of the user-defined state.
AutoState or Failure	Used to associate the state name with an autostate (such as **Idle**, **Busy**, **Inactive**, or **Failed**) or a particular Failure Name. If not used, the state must be assigned by using either the Assign module or the Resource State within the Seize or Process modules.

Statistic module

DESCRIPTION

The Statistic module is used to define additional statistics that are to be collected during the simulation and also to specify output data files. While summary statistics (for example, Average and Maximum) are automatically generated for each statistic, if you specify an output file, then each individual observation (for example, each tally recorded) is written out to an output file. The types of statistics that may be defined in the Statistic module are time-persistent, tallies (observational data), count-based, outputs, and frequency-based.

You can also specify custom periodic patterns for statistical data collection. this allows you to collect and track performance statistics by hour, shift, day, or week. You can specify either a repeating or specific (non-repetitive) time-period during a simulation run. For example:

- You could use a **repeating** time-period to gather statistics during an 8-hour work shift or every 1st hour during an 8-hour work shift.
- You could use a **specific** time-period to gather statistics on the third day only during a week-long simulation run.

TYPICAL USES

- Gathering observational data to compare two simulation configurations (save data to file for a tally)
- Collecting additional statistical information such as number in queue plus number resources busy
- Generating statistics on various states of an operator or machine
- Tracking the number of customers in a grocery store using NSTO function and Store/Unstore modules
- Collecting and tracking performance statistics for a given period of time during a simulation

PROMPTS

Prompt	Description
Name	The name of the schedule being defined. This name must be unique.
Type	Type of statistic being defined. Type may be **Time-Persistent** (Dstat), observational **Tally-**, **Counter-**, **Output-** or **Frequency-**based.
Tally Name	Defines the symbol name for the tally. This name appears in the report as the tally identifier. Applies only when Type is Tally.
Counter Name	Defines the symbol name for the counter. This name appears in the report as the counter identifier. Applies only when Type is Counter.
Frequency Type	Indicates whether statistics are being collected on the state of a resource (State) or on an expression (Value). Applies only when Type is Frequency.
Expression	Name of the time-persistent statistic, value of the output statistic or expression for the frequency. For frequencies, if the Frequency Type is State, this field is a resource name indicating that statistics are to be collected on the state of a resource. If Frequency Type is Value, this field simply specifies the expression on which statistics are to be collected. Applies only when Type is Time-Persistent, Output, or Frequency.

Prompt	**Description**
Collection Period	Defines whether to collect statistics for the entire replication or for a different period. If you select User Specified, fill in the options under User Specified Collection Parameters to specify when to collect statistics. This property is available for Frequency and Time-Persistent statistics only.
Start Time	Defines the beginning of the statistic collection (sampling) period, in units of time after the start of the replication. This property is available only if you specify a collection period of User Specified.
Units	Specifies the units of the start time value. This property is available only if you specify a collection period of User Specified.
Duration	Defines the duration of the statistic sampling period. This property is available only if you specify a collection period of User Specified.
Units	Specifies the units of the duration value. This property is available only if you specify a collection period of User Specified.
Repeat Statistics	Specifies whether to collect statistics for additional periods of time. This property is available only if you specify a collection period of User Specified.
Repeat Interval	Defines the intervals (since the Start Time) after which the statistic collection will start again. If the Repeat Interval is shorter than the Duration time, the simulation will continue to collect data without interruption when the Duration time ends. Unless you want this behavior to occur, define a Repeat Interval that is longer than the Duration. This property is available only if you specify a collection period of User Specified.
Units	Specifies the units of the Repeat Interval value. This property is available only if you specify a collection period of User Specified.

Prompt	Description
Report Label	Defines the symbol name for the expression. This name appears as the report label associated with the expression in the reports. If a statistic has already been defined through a different module, the Report Label field will automatically appear. Applies only when Type is Time-Persistent, Output, or Frequency.
Limit	Defines the terminating condition for a simulation replication. When the counter reaches or exceeds the specified limit, the simulation replication ends and the report is generated. Applies only when Type is Counter.
Initialization Option	Indicates if the counter will be initialized (reset to zero) between simulation replications. If you select **No** and multiple replications are performed, then the value of the counter at the end of a replication will be retained as the initial value at the beginning of the next replication. If you select **Yes**, the counter value is set to zero at the beginning of each replication. If you select **Replicate**, the counter value will be cleared when the other statistics (for example, tallies, time-persistent) are cleared as specified in Setup.
Tally Output File	Specifies the system-specific file name to which the raw statistical data are written. Applies only when Type is Tally.
Counter Output File	Specifies the system-specific file name to which the raw statistical data are written. Applies only when Type is Counter.
Output File	Specifies the system-specific file name to which the raw statistical data are written. Applies only when Type is Frequency, Output, or Time-Persistent. This property is not available for Time-Persistent and Frequency statistics if you specify a collection period other than Entire Replication.
Categories	Lists the category or categories of data values to collect for this statistic. This property is available only for the Frequency Type.
Constant or Range	Specifies whether statistics are being collected based on the Expression Value being a Constant value (or state of a resource) or a Range of values.

Prompt	Description
Value	Specifies the value when collecting statistics for a Constant value. Therefore, statistics will be collected against this category whenever the Expression Value equals this Value (or whenever the resource is in this state). If collecting statistics for a Range value, this field specifies the lowest value in the range. Therefore, statistics will be collected against this category whenever the Expression Value is greater than Value and less than or equal to High Value.
High Value	Specifies the highest value in the range, if collecting statistics for a Range value. Statistics will be collected against this category whenever the Expression Value is greater than Value and less than or equal to High Value.
Category Name	Defines the symbol name for this category. This name appears in the summary report as the category identifier. If left blank, the identifier will simply be the Value or Range.
Category Option	This field indicates whether or not a particular category is to be excluded from the Restricted Column in the summary report.

Storage module

DESCRIPTION

The Storage module defines the name of a storage. Storages are automatically created by any module that references the storage so that this module is seldom needed. The only time this module is needed is when a storage is defined as a member of a storage set or specified using an attribute or expression.

TYPICAL USES

- Defining an animate storage for a set of storages

PROMPTS

Prompt	Description
Name	The name of the storage set being defined. This name must be unique.

5 The Advanced Transfer Panel

This chapter presents the modules from the Advanced Transfer panel, which includes the modules needed for modeling the movement of entities from one location to another. Example uses of each module are included in this section.

The Advanced Transfer panel is accessible to licensed editions of Arena Standard and Professional

General flowchart modules

Flowchart modules are the set of objects that are placed in the model window to describe the simulation process.

Enter module

Description

The Enter module defines a station (or a set of stations) corresponding to a physical or logical location where processing occurs. When an entity arrives at an Enter module, an unloading delay may occur and any transfer device used to transfer the entity to the Enter module's station may be released.

The station (or each station within the defined set) has a matching Activity Area that is used to report all times and costs accrued by the entities in this station. This Activity Area's name is the same as the station. If a parent Activity Area is defined, then it also accrues any times and costs by the entities in this station.

Typical uses

- The start of a part's production in a series of parallel processes where the part's forklift needs to be released
- The start of a document's processing after the document has been created where the mail clerk resource needs to be released

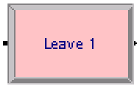
Prompts

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.

Prompt	Description
Station Type	Type of station, either a single **Station** or station **Set**.
Station Name	Name of the individual station. A given station can only exist once within a model.
Parent Activity Area	Name of the Activity Area's parent.
Associated Intersection	Name of the intersection associated with this station in a guided transporter network.
Report Statistics	Specifies whether or not statistics will automatically be collected and stored in the report database for this station and its corresponding activity area.
Set Name	Name of the station set. A given station set can only exist once within a model.
Save Attribute	Specifies the attribute to be used to store the index into the station set for an entity entering this module.
Set Members	This repeat group permits you to define the individual stations that are to be members of the specified station set. A station set must have at least one member station. Active when Station Type is Set.
Station Name	This field indicates the name of a station that is to be a member of this station set. A given station can only exist within a model once. Therefore, an individual station can only be the member of one station set, and that individual station may not be the name of a station in another module.
Parent Activity Area	Name of the Activity Area's parent for the station set member.
Associated Intersection	Name of the intersection associated with this station set in a guided transporter network.
Report Statistics	Specifies whether or not statistics will automatically be collected and stored in the report database for this station set member and its corresponding activity area.
Allocation	Type of category to which the entity's incurred delay time and cost will be added.
Delay	This field defines the delay that will be experienced by entities immediately upon arrival at the station.

Prompt	Description
Units	Time units used for the delay time.
Transfer In	If a resource, transporter, or conveyor was used to transfer the entity to this station, this can be used to release, free, or exit the device. If **Release Resource** is selected, the specified resource is released. If **Free Transporter** is selected, the specified transporter is freed. If **Exit Conveyor** is selected, the specified conveyor is exited.
Transporter Name	Name of the transporter to be freed upon arrival to the station. Active when Transfer Name is Free Transporter.
Unit Number	Unit number of the transporter if the transporter is multi-capacity.
Conveyor Name	Name of the conveyor to exit upon arrival to the station.
Resource Type	Type of allocation, either single **Resource** or resource **Set**.
Resource Name	Name of the resource to release. Active when Transfer Name is Release Resource.
Set Name	Name of the resource set from which the resource is to be released.
Release Rule	Determines which member of the set is to be released, either the **Last Member Seized**, **First Member Seized**, or **Specific Member**.
Set Index	Index into the set that determines which member of the set is to be released.
Attribute Name	Name of the attribute that determines the instance number of the resource to release.
Expression	Expression value that determines the instance number of the resource to release.

Leave module



Description

The Leave module is used to transfer an entity to a station or module. An entity may be transferred in two ways. It can be transferred to a module that defines a station by referencing the station and routing, conveying, or transporting to that station, or a graphical connection can be used to transfer an entity to another module. When an entity arrives at a Leave module, it may wait to obtain a transfer device (resource, transporter, or conveyor). When the transfer device has been obtained, the entity may experience a loading delay. Finally, the entity is transferred from this module to a destination module or station.

TYPICAL USES

- The end of a part's production in a series of parallel processes where the part needs a forklift to be transferred to shipping

PROMPTS

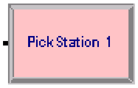
Prompt	**Description**
Name	Unique name of the module that will be displayed in the flowchart.
Allocation	Type of category to which the entity's incurred delay time and cost will be added.
Delay	Specifies a load time incurred after getting a transfer device.
Units	Time units used for the delay time.
Transfer Out	Determines whether a resource (**Seize Resource**), transporter (**Request Transporter**), or conveyor (**Access Conveyor**) is required prior to transferring the entity out of this module.
Priority	Indicates the priority of the module when either seizing a resource or requesting a transporter when there are entities waiting for that resource/transporter from other modules. This field is not visible when the Transfer Type is None or Access Conveyor.
Transporter Name	Name of the transporter to request.
Queue Type	Type of queue, either a single **Queue**, queue **Set**, **Internal** queue, **Attribute**, or **Expression**.
Queue Name	Name of the individual queue.
Queue Set Name	Name of the queue set.
Set Index	Defines the index into the queue set. Note that this is the index into the set and not the name of the queue in the set.

Prompt	Description
Queue Attribute Name	The attribute name that will be evaluated to indicate which queue is to be used.
Queue Expression	The expression that will be evaluated to indicate which queue is to be used.
Selection Rule	Method of selecting among available transporters in a set. Cyclical will cycle through available members. **Random** will randomly select a member. **Preferred Order** will always select the first available member. **Specific Member** requires an input attribute value to specify which member of the set (previously saved in the Save Attribute field). **Largest Distance** selects the transporter farthest away, and **Smallest Distance** selects the closest transporter.
Save Attribute	Attribute name used to store the index number into the set of the member that is chosen. This attribute can later be referenced with the **Specific Member** selection rule. Active when Transfer Out is Request Transporter.
Index Set	Attribute name whose value identifies the index number into the set of the member requested. The entity must have a value for the attribute before utilizing this option.
Resource Type	Type of resource for seizing, either specifying a particular **Resource**, selecting from a pool of resources (that is, a resource **Set**), **Attribute**, or **Expression**.
Resource Name	Name of the resource to seize.
Conveyor Name	Name of the conveyor to access.
# of Cells	Number of contiguous cells the entity requires.
Connect Type	Determines if the entity is to **Route**, **Convey**, or **Transport** to another station or **Connect** to another module.
Move Time	Time to route from this module to the destination station.
Units	Time units used for the move time.
Station Type	The entity’s destination station type either an individual **Station**, a station based on an **Attribute** or **Expression** value, or **By Sequence.**
Station Name	Name of the individual destination station.
Attribute Name	The attribute name that will be evaluated to indicate the station.

Prompt	Description
Expression	The expression that will be evaluated to indicate the station.

PickStation module

DESCRIPTION



The PickStation module allows an entity to select a particular station from the multiple stations specified. This module picks among the group of stations based on the selection logic defined with the module. The entity may then route, transport, convey, or connect to the station specified. If the method chosen is connect, the selected station is assigned to an entity attribute. The station selection process is based on the minimum or maximum value of a variety of system variables and expressions.

TYPICAL USES

- A part sent to a processing station based on machine’s availability at each station
- A loan application sent to a set of loan officers based on the number sent to each officer
- A customer selecting among cashier lines based on the least number waiting in each line

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Test Condition	Test condition to use for the station selection process, either **Minimum** or **Maximum**.
Number En Route to Station	The number of entities transferring to the station is considered in the station selection process.
Number in Queue	The number of entities in the queue at the station is considered in the station selection process.
Number of Resources Busy	The number of busy resources at the station is considered in the station selection process.

Prompt	Description
Expression	Determines if an additional user-defined expression is considered in the station selection process.
Transfer Type	Determines how an entity will be transferred out of this module to its next destination station—either **Route**, **Convey**, **Transport**, or **Connect**.
Save Attribute	Defines the name of the attribute that will store the station name that is selected, visible when the transfer method is Connect.
Route Time	Move time of the entity from its current station to the station determined through this module.
Units	Time units for route-time parameters.

PickStation module — Stations dialog box

PROMPTS

Prompt	Description
Station Name	Station name for the selection process. All potential station names are required even though the Number En Route to station may not be a consideration in the PickStation evaluation.
Queue Name	Visible only if Number in Queue is checked, it defines the name of the queue corresponding to the station defined.
Resource Name	Visible only if Number of Resources Busy is checked, it defines the resource name corresponding to the station defined.
Expression	Visible only if Expression is checked, it defines an additional expression that will be considered in the station selection process.

Route module

DESCRIPTION

The Route module transfers an entity to a specified station or the next station in the station visitation sequence defined for the entity. A delay time to transfer to the next station may be defined.

When an entity enters the Route module, its Station attribute (Entity.Station) is set to the destination station. The entity is then sent to the destination station, using the route time specified.

If the station destination is entered as By Sequence, the next station is determined by the entity’s sequence and step within the set (defined by special-purpose attributes Entity.Sequence and Entity.Jobstep, respectively).

TYPICAL USES

- Send a part to its next processing station based on its routing slip
- Send an account balance call to an account agent
- Send restaurant customers to a specific table

PROMPTS

Prompt	**Description**
Name	Unique name of the module that will be displayed in the flowchart.
Route Time	Travel time from the entity’s current location to the destination station.
Units	Time units for route-time parameters.
Destination Type	Method for determining the entity destination location. Selection of **By Sequence** requires that the entity has been assigned a sequence name and that the sequence itself has been defined.
Station Name	Name of the individual destination station.
Attribute Name	Name of the attribute that stores the station name to which entities will route.
Expression	Expression that is evaluated to the station name where entities will route.

Station module

DESCRIPTION

The Station module defines a station (or a set of stations) corresponding to a physical or logical location where processing occurs. If the Station module defines a station set, it is effectively defining multiple processing locations.

The station (or each station within the defined set) has a matching Activity Area that is used to report all times and costs accrued by the entities in this station. This Activity Area's name is the same as the station. If a parent Activity Area is defined, then it also accrues any times and costs by the entities in this station.

Typical uses

- Defining a lathe area
- Defining a set of toll booths
- Defining a food preparation area

Prompts

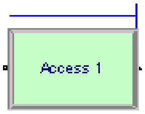
Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Station Type	Type of station being defined, either as an individual **Station** or a station **Set**.
Station Name	Name of the individual station.
Set Name	Name of the station set.
Parent Activity Area	Name of the Activity Area's parent.
Associated Intersection	Name of the intersection associated with this station in a guided transporter network.
Report Statistics	Specifies whether or not statistics will automatically be collected and stored in the report database for this station and its corresponding activity area.
Save Attribute	Attribute name used to store the index number into the station set of the member that is selected.
Station Set Members	Names of the stations that are members of this station set.
Station Name	A given station can only exist once within a model. Therefore, an individual station can only be the member of one station set, and that individual station may not be the name of a station in another module.
Parent Activity Area	Name of the Activity Area's parent for the station set member.
Associated Intersection	Name of the intersection associated with this station set in a guided transporter network.

Prompt	Description
Report Statistics	Specifies whether or not statistics will automatically be collected and stored in the report database for this station set member and its corresponding activity area.

Conveyor flowchart modules

Access module

DESCRIPTION



The Access module allocates one or more cells of a conveyor to an entity for movement from one station to another. Once the entity has control of the cells on the conveyor, it may then be conveyed to the next station.

When an entity arrives at an Access module, it will wait until the appropriate number of contiguous cells on the conveyor are empty and aligned with the entity's station location.

TYPICAL USES

- Parts accessing a conveyor to be sent to a paint booth
- Glass accessing a conveyor to be transferred to a cutting station

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Conveyor Name	Name of the conveyor that the entity desires.
# of Cells	Number of contiguous conveyor cells the entity requires for movement on the conveyor.
Queue Type	Determines the type of queue used to hold the entities, either an individual **Queue**, a queue **Set**, and **Internal** queue or an **Attribute** or **Expression** that evaluate to the queue name.
Queue Name	Name of the queue that will hold the entity until it accesses the conveyor.
Set Name	Name of the set of queues.

Prompt	Description
Set Index	Defines the index into the queue set. Note that this is the index into the set and not the name of the queue in the set. For example, the only valid entries for a queue set containing three members is an expression that evaluates to 1, 2, or 3.
Attribute Name	Defines the name of the attribute that stores the queue name to which entities will reside.
Expression	Defines the name of the expression that stores the queue name to which entities will reside.

Convey module

Description

The Convey module moves an entity on a conveyor from its current station location to a specified destination station. The time delay to convey the entity from one station to the next is based on the velocity of the conveyor (specified in the Conveyor module) and the distance between the stations (specified in the Segment module).

When an entity enters a Convey module, its Station attribute (*Entity.Station*), is set to the destination station. The entity is then conveyed to the destination station. If the destination type is specified as By Sequence, the next station is determined by the entity’s Sequence and Jobstep within the sequence (special-purpose attributes *Entity.Sequence* and *Entity.Jobstep*, respectively).

Typical uses

- Convey bags from a location to the baggage claim area
- Convey parts from a load station to processing station

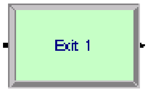
Prompts

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Conveyor Name	Name of the conveyor on which the entity will convey. If left blank, the previously accessed conveyor is assumed.

Prompt	Description
Destination Type	Method for specifying the entity destination (**By Sequence**, **Station**, **Attribute**, or **Expression**). Selection of By Sequence requires that the entity has been assigned a sequence name and that the sequence itself has been defined.
Station Name	Name of the individual station.
Attribute Name	Attribute that stores the station name to which entities will route.
Expression	Expression that is evaluated to the station name where entities will route.

Exit module

DESCRIPTION



The Exit module releases the entity's cells on the specified conveyor. If another entity is waiting in queue for the conveyor at the same station when the cells are released, it will then access the conveyor.

TYPICAL USES

- Cases exit a conveyor for packing
- Bad parts are removed from the conveyor and disposed
- Passengers remove luggage from the baggage claim conveyor

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Conveyor Name	Name of the conveyor on which the entity will exit. If left blank, the previously accessed conveyor is assumed.
# of Cells	Number of contiguous conveyor cells the entity will relinquish.

Start module

DESCRIPTION

The Start module changes the status of a conveyor from inactive to active. The conveyor may have been deactivated from either the Stop module or by initially being set to inactive at the start of the simulation. The velocity of the conveyor may be changed permanently when the conveyor is started.

Typical uses

- Start a bottling conveyor after scheduled maintenance
- Start a baggage claim conveyor when bags have arrived

Prompts

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Conveyor Name	Name of the conveyor to start.
Velocity	Speed of the conveyor once it begins to operate. This value will change the speed of the conveyor permanently, until it is changed in another module.
Units	Velocity time units.

Stop module

Description

The Stop module sets the operational status of a conveyor to inactive. The conveyor may have been activated from either the Start module or by initially being set to active at the start of the simulation. When the entity enters the Stop module, the conveyor will stop immediately, regardless of the type of conveyor or the number of entities currently on the conveyor.

Typical uses

- Stop a baggage conveyor after a pre-determined amount of time
- Stop a conveyor for scheduled maintenance

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Conveyor Name	Name of the conveyor to stop.

Transporter flowchart modules

Activate module

DESCRIPTION

The Activate module increases the capacity of a previously halted transporter. The transporter unit that is activated will reside at the station location at which it was halted until it is moved or requested by an entity. If there is an entity) awaiting a transporter at the time the unit is activated, the entity will gain control of the transporter immediately.

TYPICAL USES

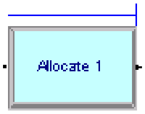
- Activate a forklift that was down for scheduled maintenance
- Activate gurneys in an emergency room after cleaning and restocking
- Start additional wait staff at a restaurant

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Transporter Name	Name of the transporter to activate.
Unit Number	Determines which of the transporter units in the transporter set to activate.

Allocate module

Description



The Allocate module assigns a transporter to an entity without moving it to the entity's station location. The entity then has control of the transporter to either move it to a particular location or to halt it for a breakdown or failure. A particular transporter unit may be allocated to the entity or a selection rule may be used to determine which of the transporters will be assigned to the entity.

Typical uses

- A mechanic allocates a forklift for scheduled maintenance
- Allocate a taxi to pick up waiting passengers

Prompts

Prompt	**Description**
Name	Unique name of the module that will be displayed in the flowchart.
Transporter Name	Name of the transporter to allocate.
Selection Rule	Rule for determining which of the transporters to allocate to the entity. The selection rule has the following options: **Cyclical**, **Random**, **Preferred Order**, **Specific Member**, **Largest Distance**, and **Smallest Distance**.
Save Attribute	The attribute name that will store the unit number of the selected transporter.
Unit Number	Determines the specific transporter in the transporter set to allocate to the entity. It is only visible if selection rule is Specific Member.
Priority	Priority value of the entity waiting at this module for the transporter specified if one or more entities are waiting for the same transporter anywhere in the model.
Entity Location	This field is used to determine the transporter closest to or farthest from the requesting entity when using the Largest Distance or Smallest Distance transporter selection rules. For free-path transporters, the entity location must be specified as a station. For guided transporters, the entity location may be specified as a station, intersection, or network link.

Prompt	Description
Queue Type	Type of queue used to hold the entities while waiting to allocate the transporter, either an individual **Queue**, a queue **Set**, an **Internal** queue, or an **Attribute** or **Expression** that evaluate to the queue name.
Queue Name	Name of the individual queue.
Queue Set Name	Name of the queue set that contains the queue being referenced.
Set Index	The index into the queue set. Note that this is the index into the set and not the name of the queue in the set. For example, the only valid entries for a queue set containing three members is an expression that evaluates to 1, 2, or 3.
Attribute Name	The attribute name that will be evaluated to the queue name.
Expression	The expression that will be evaluated to the queue name.

Free module

DESCRIPTION

The Free module releases the entity's most recently allocated transporter unit. If another entity is waiting in a queue to request or allocate the transporter, the transporter will be given to that entity. If there are no waiting entities at the time the transporter unit is freed, the transporter will wait idle at the freeing entity's station location, unless otherwise specified in the Transporter module.

TYPICAL USES

- A part awaiting a shipping truck frees its forklift
- An airport transfer cart completes its trip

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Transporter Name	Name of the transporter to free. A blank value assumes the most recently allocated or requested transporter.
Unit Number	Determines which of the transporter units in the transporter set to free.

Halt module

Description

The Halt module changes the status of a transporter unit to inactive. If the transporter is busy at the time when an entity enters the Halt module, the status is considered busy and inactive until the entity that controls the transporter frees the unit. If the transporter is idle at the time when an entity halts the transporter, it is set to inactive immediately. Once a transporter unit has been halted, no entities will get control of the transporter until it is activated.

Typical uses

- Stop a forklift for scheduled maintenance
- Disable a broken gurney in an emergency room

Prompts

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Transporter Name	Name of the transporter to halt.
Unit Number	Determines which of the transporter units in the transporter set to halt.

Move module

Description

The Move module advances a transporter from one location to another without moving the controlling entity to the destination station. The controlling entity remains at its current module location until the transporter arrives at its destination. At that time, the entity will be able to move to another module or task in the model.

If the transporter being moved is a free-path transporter, then the time delay to move the transporter from one station to the next is based on the velocity of the transporter and the distance between the stations (specified in the Distance module). If the transporter being moved is a guided transporter, then the time delay to move the transporter depends not only on the velocity and distance to the destination, but also the transporter’s acceleration, deceleration, and any traffic it encounters on the way.

TYPICAL USES

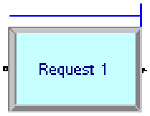
- Move a broken forklift to a service station
- Move a worker to a break room
- Move a waiter to the kitchen

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Transporter Name	Name of the transporter to move.
Unit Number	Determines which of the transporter units in the transporter set to move.
Destination Type	Determines the method for specifying the transporter destination. The **Intersection** and **Network Link** options apply only to guided transporters.
Station Name	Name of the individual destination station.
Attribute Name	Name of the attribute that stores the destination station name to which entities will route.
Expression	Expression that is evaluated to the destination station name where entities will route.
Intersection Name	Defines the name of the intersection to which the guided transporter will move.
Network Link Name	Defines the name of the network link to which the guided transporter will move.
Zone	The specific zone number of the Network Link Name.
Velocity	Specifies the temporary velocity at which the transporter is moved to the destination station.
Units	Velocity time units.

Request module

DESCRIPTION



The Request module assigns a transporter unit to an entity and moves the unit to the entity's station location. A specific transporter unit may be specified or the selection may occur based on a rule.

When the entity arrives at the Request module, it is allocated a transporter when one is available. The entity remains at the Request module until the transporter unit has reached the entity's station. The entity then moves out of the Request module.

TYPICAL USES

- A sanded part requests a cart to take it to the paint shop
- Customers in a restaurant are ready to order and thus request a waiter

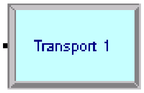
PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Transporter Name	Name of the transporter to allocate.
Selection Rule	Rule for determining which of the transporters to allocate to the entity, either **Cyclical, Random**, **Preferred Order**, **Specific Member**, **Largest Distance**, or **Smallest Distance**.
Save Attribute	Denotes the attribute name that will store the unit number of the selected transporter.
Unit Number	Determines the specific transporter unit in the transporter set to request.
Priority	Priority value of the entity waiting at this module for the transporter specified if one or more entities are waiting for the same transporter anywhere in the model.
Entity Location	The location to which the transporter will move after being allocated. For free-path transporters, the entity location must be specified as a station. For guided transporters, the entity location may be specified as a station, intersection, or network link.
Velocity	Speed at which the transporter is moved to the entity location in length units per unit time. The unit time is specified in the Units field.
Units	Velocity time units.

Prompt	Description
Queue Type	Type of queue used to hold the entities while waiting to access the transporter, either an individual **Queue**, a queue **Set**, an **Internal** queue, or an **Attribute** or **Expression** that evaluates to the queue's name.
Queue Name	Name of the individual queue.
Set Name	Name of the queue set.
Set Index	The index into the queue set. Note that this is the index into the set and not the name of the queue in the set. For example, the only valid entries for a queue set containing three members is an expression that evaluates to 1, 2, or 3.
Attribute Name	The attribute name that will be evaluated to the queue name.
Expression	Expression evaluated to the queue name.

Transport module

DESCRIPTION



The Transport module transfers the controlling entity to a destination station. After the time delay required for the transport, the entity reappears in the model at the Station module. Free-path transporters are always moved to the same station destination as the entity. Guided transporters may be moved to a different network location than the entity's station destination using the Guided Tran Destination Type.

If a free-path transporter is being used, then the time delay to move the entity and transporter from one station to the next is based on the velocity of the transporter and the distance between the stations (specified in the Distance module). If a guided transporter is being used, then the time delay to transport the entity depends not only on the velocity and distance to the guided transporter's destination, but also the transporter's acceleration, deceleration, and any traffic it encounters on the way.

When an entity enters the Transport module, its Station attribute (Entity.Station) is set to the entity's destination station. The entity is then transported to the destination station.

If the station destination is entered as By Sequence, the next station is determined by the entity's Sequence and Jobstep within the set (special-purpose attributes Entity.Sequence and Entity.Jobstep, respectively).

Typical uses

- A forklift transports a pallet of parts to the next processing station
- A mail cart transports packages from the mailroom to the order-processing station

Prompts

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Transporter Name	Name of the transporter to move. If blank, the most recently requested or allocated transporter is assumed.
Unit Number	Determines which of the transporter units in the transporter set to transport.
Entity Destination Type	Determines the method for specifying the entity destination station logic. Selection of **By Sequence** requires that the entity has been assigned a sequence name and that the sequence itself has been defined.
Station Name	Name of the individual station.
Attribute Name	Name of the attribute that stores the station name to which entities will route.
Expression	Expression that is evaluated to the station where entities will route.
Velocity	Specifies the temporary velocity at which the entity and transporter are moved to the destination station.
Units	Velocity time units.
Guided Tran Destination Type	Allows the specification of a guided transporter destination that is different from the entity's destination. This field is ignored if Transporter Name is a free-path transporter.
Station Name	Defines the name of the station with an associated intersection to which the guided transporter will move.
Attribute Name	Defines the name of the attribute that stores the station name with an associated intersection to which the guided transporter will move.

Prompt	Description
Expression	Defines an expression that evaluates to the network location (intersection, station, network link) to which the guided transporter will move.
Intersection Name	Defines the name of the intersection to which the guided transporter will move.
Network Link Name	Defines the name of the network link to which the guided transporter will move.
Zone	The specific zone number on Network Link Name.

Data modules

Data modules are the set of objects in the spreadsheet view of the model that define the characteristics of various process elements, such as resources and queues.

Sequence module

DESCRIPTION

The Sequence module is used to define a sequence for entity flow through the model. A sequence consists of an ordered list of stations that an entity will visit. For each station in the visitation sequence, attributes and variables may be assigned values.

Each station in the visitation sequence is referred to as a step (or jobstep) in the sequence.

Three special-purpose attributes are provided for all entities. The Sequence attribute (Entity.Sequence) defines the sequence that an entity is to follow; a value of 0 indicates that the entity is not following any sequence. In order for an entity to follow a sequence, its Sequence attribute must be assigned a value (for example, in the Assign module).

The Jobstep attribute (Entity.Jobstep) stores the entity’s current step number in the sequence. This value is updated automatically each time an entity is transferred. You typically do not need to assign explicitly a value to Jobstep in the model.

The PlannedStation attribute (Entity.PlannedStation) stores the number of the station associated with the next jobstep in the sequence. This attribute is not user-assignable.

It is automatically updated whenever Entity.Sequence or Entity.JobStep changes, or whenever the entity enters a station.

Jobstep names must be globally unique.

Typical uses

- Define a routing path for part processing
- Define a sequence of steps patients must take upon arrival at an emergency room

Prompts

Prompt	Description
Name	Name of the sequence.
Steps	Repeat group that defines the ordered list of stations an entity visits for the sequence named, as well as attribute and variable assignments to be made at each of the stations in the sequence.
Station Name	Name of the next station in the visitation sequence.
Step Name	A step name may optionally be associated with each station in the visitation sequence. The same step name may appear in different sequences, but not more than once in a given sequence.
Next Step	Name of the next step to be visited in the visitation sequence. If left blank, the default is for steps to be performed in the order in which they are defined in the sequence.
Assignments	Repeat group is used to define attribute, variable, picture, picture set, or other assignments.
Assignment Type	Type of assignment to be made before transferring the entity to the station specified in this step of the sequence. Assignment type may be **Attribute**, **Variable**, **Picture**, **Picture Set**, or **Other**.
Attribute Name	Name of the attribute to which the specified value will be assigned.
Variable Name	Name of the variable to which the specified value will be assigned.
Picture	Name of the picture that will be assigned to the entity’s Entity.Picture value.
Picture Set	Name of the picture set for assigning an Entity.Picture value to an entity.

Prompt	Description
Set Index	Index into the picture set that will be used for assigning Entity.Picture.
Other	Defines any model status variable or special-purpose attribute to which the specified value will be assigned.
Value	Value to be assigned to the variable specified.

Conveyor module

DESCRIPTION

The Conveyor module allows the definition of either an accumulating or non-accumulating conveyor for entity movement between stations. When using a conveyor to transfer between modules, the conveyor must be defined using this module.

TYPICAL USES

- Baggage-handling belt
- Bottling conveyor

PROMPTS

Prompt	Description
Name	Name of the conveyor.
Segment Name	Name of the segment set defined in the segment module that is associated with the specified conveyor. The segment set defines the stations between which the conveyor transfers entities.
Type	Determines the conveyor type, either **Accumulating** or **Non-Accumulating**.
Velocity	The initial speed at which entities move along the conveyor in length units per base unit time.
Units	Velocity time units.
Cell Size	The smallest portion of a conveyor that an entity can occupy.
Max Cells Occupied	The maximum number of conveyor cells that any given entity will access at a time.

Prompt	Description
Accumulation Length	The accumulation length of the entity. This size is utilized in the simulation only when the conveyor type is Accumulating.
Initial Status	Initial status of the conveyor, either **Active** or **Inactive**.
Report Statistics	Specifies whether or not statistics will be collected automatically.

Segment module

Description

The Segment module defines the distance between two stations in the segment set of a conveyor. The beginning station, ending station, and segment are used to create the appropriate segment set, which defines the path of the conveyor. Typically, more than one Segment module is used to define the segment set.

Typical uses

- Distillery operation that has a path from bottling to labeling to the boxing area
- Baggage-handling area that has a path from check-in to a holding area to a staging area

Prompts

Prompt	Description
Name	Name of the segment.
Beginning Station	The name of the first station that is located on the conveyor.
Next Station	The name of the next station that is located on the conveyor.
Length	Distance between this station and the previous station.

Transporter module

DESCRIPTION

The Transporter module allows the definition of a free-path or guided transporter device for entity movement from one location to another. Free-path transporters move freely between stations and are not influenced by other transporter traffic. The Distance module specifies the distances between stations. Guided transporters, on the other hand, are restricted to run on fixed paths such as tracks or rails. Movement may be affected by traffic congestion from other vehicles. Guided transporters travel on networks defined using the Network and Network Link modules.

Each spreadsheet entry creates a single transporter set.

When using a transporter to transfer between modules or as the server in a Leave or Enter module, the transporter must be defined using the Transporter module.

TYPICAL USES

- Forklifts that transfer parts between processing stations
- Wheelchairs that move patients between diagnostic areas

PROMPTS

Prompt	Description
Name	Name of the transporter. Can be used to represent a set of multiple transporter units; each unit within the set is capable of independent movement.
Number of Units	Number of independent transporter units in the transporter set.
Type	Specifies whether the transporter set is free path or guided.
Distance Set	Specifies the name of the distance set that contains the station names and distances between which the free path transporter will travel. Distances are defined in the Distance module.
Network Name	Specifies the name of the network that defines the distances and zone configuration of all paths that the guided transporter set may follow. Networks are defined in the Network module and the links/segments that make up a network are defined using the Network Link module.
Velocity	The initial velocity of the set of transporter units in length units per base time unit.
Units	Velocity time units.

Prompt	Description
Acceleration	Specifies the acceleration in length units per unit time squared that is applied to return a guided transporter to its normal operating velocity after slowing down. The unit time is specified in the Units field.
Deceleration	Specifies the deceleration in length units per unit time squared that is applied to brake a guided transporter when it needs to slow down or stop. The unit time is specified in the Units field.
Turning Velocity Factor	Defines the factor to be applied to a guided transporter's velocity when it negotiates a turn. In order for slow-downs through turns to be performed automatically, directions must be defined for all network links in the network Network Name.
Zone Control Rule	Determines when a guided transporter controlling a zone releases it, allowing another transporter access to that zone.
Units	This field specifies the time units for the Acceleration and Deceleration entries (such as per minute squared or per hour squared).
Initial Position Status	Repeat group that allows the specification of initial position. One entry may be used for each unit in the transporter set (as defined by Capacity).
Initial Position	Determines the type of initial position for the transporter unit(s). If a free-path transporter, Default will place the transporter at the first station location specified in the Distance module Distance Set. If a guided transporter, Default will place the transporter at the first network link specified in the Network module Network Name.
Station Name	Name of the station for the transporter's initial position.
Intersection Name	Defines the name of the intersection at which the guided transporter unit will be initialized.
Network Link Name	Defines the name of the network link at which the guided transporter unit will be initialized.
Zone	Defines the zone number on the network link at which the guided transporter unit will be initialized.
Initial Status	Establishes the initial status, **Active** or **Inactive**, of the transporter unit.

Prompt	Description
Size Based On	Determines whether the guided transporter's size will be specified in length units or number of zones.
Size Value	Specifies the initial transporter size in number of length units or zones according to the Size Based On field.
Report Statistics	Specifies whether or not statistics will be collected automatically.

Distance module

DESCRIPTION

The Distance module is used to define the travel distances between all stations that free-path transporters may visit. It consists of a distance name and a list of station pairs and individual distances.

TYPICAL USES

- Defining the distance a forklift travels between processing stations
- Defining the distance hospital gurneys travel between various diagnostic areas

PROMPTS

Prompt	Description
Name	Distance set name.
Beginning Station	Starting station name.
Ending Station	Ending station name.
Distance	Distance between the beginning and ending station.

Network module

Description

The Network module defines a system map that a set of guided transporters will follow. A network encompasses the set of links specified in its Network Links repeat group. The parameters of a network link (for example, length, intersections, directions) are defined in the Network Link module.

Typical uses

- Required when using guided vehicles

Prompts

Prompt	Description
Name	Name of the network.
Network Link Name	Name of network link included in the network.

Network Link module

Description

The Network Link module defines the characteristics of a guided transporter path between an intersection pair *Beginning Intersection Name* and *Ending Intersection Name*. The Network module then references a set of network links to define a network that guides transporters follow for movement.

Typical uses

- Required when using guided vehicles

Prompts

Prompt	Description
Name	Name of the network link.
Type	The type of movement through the link.
Beginning Intersection Name	Name of beginning intersection of the intersection pair that the link connects.
Ending Intersection Name	Name of ending intersection of the intersection pair that the link connects.

Prompt	Description
Beginning Direction	Direction of link as it leaves the beginning intersection.
Ending Direction	Direction of link as it enters the ending intersection.
Number Of Zones	Number of zones in the link.
Zone Length	The length of each zone in the link. The total length of the link is the number of zones multiplied by the zone length.
Velocity Change Factor	Multiplier that is to be applied to the current velocity of any vehicle traveling through the link.

Activity Area module

DESCRIPTION

An Activity Area is an area that can be associated with one or more stations. These areas are used to collect statistics associated with a station. Each activity area may have an associated parent activity area that is used to define the hierarchy of the system modeled. Statistics are automatically rolled up through the activity area hierarchy.

TYPICAL USES

- A workstation
- A manufacturing line
- A manufacturing site
- The entire enterprise

PROMPTS

Prompt	Description
Name	Name of the activity area's parent.
Description	Optional field to describe the activity area.
Organization Level	Number used to define the hierarchy of the area.
Parent Activity Area	Name of the activity area's parent.
Report Statistics	Specifies whether or not statistics will automatically be collected and stored in the report database for this activity area.

6 The Flow Process Panel

To meet high-volume production requirements, manufacturing operations in the consumer packaged goods industry usually involve semi-continuous batch processing and/or high-speed packaging operations. While some of these operations can be modeled using discrete approximations, often a better approach is required.

The Arena Flow Process template is one of a family of application solution templates (ASTs) built on the Arena simulation system. The flowchart and spreadsheet data modules in the Flow Process panel were designed specifically to model combined discrete-continuous systems (for example, batch processing and material flow). If a system contains semi-continuous batch processes as well as high-speed packaging or filling operations, then it may be desirable to use modules from the Flow Process template with those from the Packaging template.

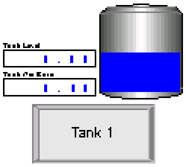
The Flow Process panel is accessible to licensed editions of Professional Edition.

Flowchart modules

Flowchart modules are the set of objects that are placed in the model window to describe the simulation process.

Tank module

Description



The Tank module defines a location where material is held or stored.

The *Capacity* is the maximum quantity that may be stored in the tank. The *Initial Level* is the quantity in the tank at the beginning of the simulation or when the system is cleared.

The *Regulators* repeat group specifies a list of devices that may be used to add or remove material from the tank (for example, valves or pumps) at a specified rate. Semi-continuous flow operations may then be executed by discrete entities using the regulators from the Flow module.

If a system contains both semi-continuous batch processes as well as high-speed packaging or filling operations, then it may be desirable to use the Flow Process panel with the Arena Packaging template. Enable the *Packaging Input Connection* and/or *Packaging Output Connection* options to connect a Flow Process Tank graphically to a Valve or Machine module in the Packaging panel.

When an Arena model (that is, *.doe* file) is run, the simulation results are stored in a Microsoft Access database (*.mdb*) file by the same name (for example, the simulation results for Banking.doe are stored in Banking.mdb). If statistics collection for all tanks is enabled in the Project Parameters dialog box AND Report Statistics is checked for an individual tank module, then Arena will automatically create, collect, and report statistics for that tank.

TYPICAL USES

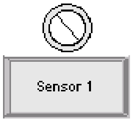
- Defines a location or holding area where bulk material is stored
- Defines a tank or vat holding liquids where the regulator represents a valve
- Defines a fluid-like material storage vessel where the regulator represents a gate valve
- Defines a bulk storage area or pile where the regulator represents a piler/reclaimer

PROMPTS

Prompt	Description
Name	The tank’s identifier.
Capacity	The tank’s capacity.
Initial Level	The quantity in the tank at the start of the simulation.
Regulator Name	Specifies a device that is used to add to or remove material from the tank (for example, a valve name).
Maximum Rate	The initial maximum rate for the regulator.
Units	Time units of maximum rate.
Packaging Input Connection	Specifies whether the tank is connected to and allows input flow from a Valve module in the Packaging panel.
Packaging Output Connection	Specifies whether the tank is connected to and allows output flow to a Valve or Machine module in the Packaging panel. The Machine Type for the Machine module must be Filling Machine.
Animation	Specifies whether to display default animation of the tank’s level and net rate.
Report Statistics	Specifies whether to report statistics on the tank.

Sensor module

DESCRIPTION



The Sensor module defines a detection device that monitors the level of material in a tank (Tank module). A sensor's location is specified using the *Tank Name*, *Location Type*, *Level*, or *Percentage* prompts.

A sensor is activated when its location is crossed in the *Crossing Direction* AND the sensor is enabled. When the sensor is activated, one or more *Actions* may be executed. Additionally, the *Create Discrete Entity* option may be used to create a discrete entity and send it to custom logic.

The Initial State field specifies whether the sensor is enabled or disabled at the beginning of the simulation run. If a sensor is disabled, it will be ignored and never activate. A sensor may be enabled/disabled dynamically at any point during the simulation run by assigning the SensorState (*Sensor ID*) variable.

TYPICAL USES

- Defines a device to monitor tank level so decisions can be made based on sensor state
- Defines a device to initiate action when a tank becomes full, empty, or some other significant level is crossed
- Defines a device to trigger re-ordering of material when stock runs low
- Defines a device to start and stop reclamations from a pile based on inventory level

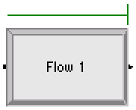
PROMPTS

Prompt	**Description**
Name	The sensor's identifier.
Tank Name	The name of the tank where the sensor is located.
Level	The level location of the sensor if Location Type is *Specific Level*.
Percentage (0-100)	The percentage location of the sensor if Location Type is *Percentage Capacity.*
Crossing Direction	The sensor is activated (triggered) when its location is crossed in this direction.
Initial State	Specifies whether the sensor is enabled or disabled at the beginning of the simulation run.

Prompt	Description
If Sensor is Activated:	
Action Type	Specifies the type of action to perform.
Regulator Name	The name of the regulator to regulate.
New Maximum Rate	The new maximum rate for the regulator.
Units	Time units of new maximum rate.
Signal Value	The signal value to send.
Variable Name	The name of the variable to assign.
New Value	The new variable value.
Create Discrete Entity	Specifies whether to create a discrete entity when the sensor is activated (triggered).
Animation	Specifies whether to display a "covered/uncovered" global picture for the sensor.

Flow module

DESCRIPTION



The Flow module is used in conjunction with the Tank module to model semi-continuous flow operations such as adding material to a tank, removing material from a tank, or transferring material between two tanks.

When an entity enters the Flow module, a flow operation of the specified Type is initiated (that is, Add, Transfer, or Remove). The tanks affected by the flow are determined by the regulators specified as the source and/or destination. A regulator is a device (for example, a valve) that is used to add or remove material from a tank. A tank's regulators are defined in the Tank module.

The entity is held in the Flow module until its flow operation is completed. The flow operation is completed when the *first* of three possible conditions is satisfied:

- When the specified *Quantity* has been transferred.
- When the specified *Time* has elapsed.
- When the specified *Signal* Value is received from a Signal module.

The flow rate of the entity is regulated according to the maximum rate(s) of the source regulator and/or destination regulator. Additionally, the flow rate may be further constrained due to starvation or blocking from empty or full tanks. The initial maximum rate of a regulator is specified in the Tank module. A regulator's maximum rate may be changed during a run using the Regulate module. The RegulatorMaxRate(*Regulator ID*) function returns the current maximum rate of a regulator. The RegulatorRate(*Regulator ID*) function returns the current rate of a regulator.

The actual quantity transferred by an entity in the Flow module may be stored in the *Quantity Save Attribute* when the entity exits the module.

Typical uses

- Adding material to a tank
- Removing material from a tank
- Transferring material between two tanks

Prompts

Prompt	Description
Name	The module's identifier.
Type	The flow type, which may be **Transfer**, **Add**, or **Remove**.
Flow Destination/ Source:	
Source Regulator Type	Indicates how the source regulator will be specified; as **Regulator**, **Regulator Set**, **Attribute**, **Expression**.
Regulator Name	Name of the source regulator.
Regulator Set Name	Name of the regulator set that contains the source regulator name.
Set Index	Index into the regulator set.
Attribute Name	Name of the attribute that stores the source regulator name.
Expression	Expression that evaluates to the source regulator name.
Destination Regulator Type	Indicates how the destination regulator will be specified; as **Regulator**, **Regulator Set**, **Attribute**, **Expression**.
Regulator Name	Name of the destination regulator.

Prompt	Description
Regulator Set Name	Name of the regulator set that contains the destination regulator name.
Set Index	Index into the regulator set.
Attribute Name	Name of the attribute that stores the destination regulator name.
Expression	Expression that evaluates to the destination regulator name.
Stop Flow After:	
Quantity	The flow will be stopped after this quantity has been transferred.
Time	The flow will be stopped after this time duration has elapsed.
Units	The time units of the *Time*.
Signal Value	The flow will be stopped if this signal value is received.
Priority	The priority of the flow operation when allocating limited input or output to multiple flow operations occurring at a tank.
Allocation	Determines how the time and costs accrued during the flow operation will be allocated to the entity.
Quantity Save Attribute	Attribute to save the quantity transferred by the entity.

Regulate module

DESCRIPTION

The Regulate module provides discrete control for assigning the maximum rates of regulators defined in the Tank module.

TYPICAL USES

- Used to simulate such actions as the closing or opening of valves

PROMPTS

Prompt	Description
Name	The module description.
Maximum Rate Assignments:	
Regulator Type	Indicates how the regulator will be specified; defined as **Regulator**, **Regulator Set**, **Attribute**, or **Expression**.
Regulator Name	Name of the regulator.
Regulator Set Name	Name of the regulator set that contains the regulator.
Set Index	Index into the regulator set.
Attribute Name	Name of the attribute that stores the regulator name.
Expression	Expression that evaluates to the regulator name.
New Maximum Rate	The new maximum rate.
Units	Time units of new maximum rate.

Seize Regulator module

DESCRIPTION

A tank regulator may be used for only one flow operation at any given time.

The Seize Regulator module allocates one or more regulators to an entity. When an entity enters this module, it waits in a queue until all specified regulators are available simultaneously. An allocated regulator is released by an entity using the Release Regulator module.

TYPICAL USES

- Controls ownership of regulators and avoids multiple entities trying to use the same regulator in a Flow module simultaneously
- Chooses from a set of alternative regulators using a selection rule

PROMPTS

Prompt	Description
Name	The module identifier.
Priority	Priority value of the entity waiting at this module for the regulator(s) specified if one or more entities from other Seize Regulator modules are waiting for the same regulator(s).
Regulators:	
Regulator Type	Indicates how the regulator to seize will be specified; as **Regulator**, **Regulator Set**, **Attribute**, or **Expression**.
Regulator Name	Name of the regulator to seize.
Regulator Set Name	Name of the regulator set that contains the regulator to seize.
Selection Rule	Method of selecting among available resources in a set. **Cyclical** will cycle through available members (1-2-3-1-2-3). **Random** will randomly select a member. **Preferred Order** will always select the first available member as listed in the set (1 if available, then 2 if available, then 3). **Specific Member** requires an index value to specify which member of the set. **User Rule** may be used to code custom selection rules. The number argument is passed to the UserRule routine (which may be coded in either VBA or a user-coded dll).
Save Attribute	Attribute name used to store the index number into the set of the member that was selected. This attribute can then later be referenced in Flow, Regulate, and Release Regulator modules to identify the regulator that was chosen and seized.
Set Index	Index into the Regulator Set.
Rule Number	Specifies number argument that is passed to the UserRule routine.
Attribute Name	Name of the attribute that stores the name of the regulator to seize.
Expression	Expression that evaluates to the name of the regulator to seize.

Release Regulator module

DESCRIPTION

The Release Regulator module is used to release tank regulators that have been allocated to an entity using the Seize Regulator module. This makes those regulators available to other entities waiting to seize the regulator(s).

When the entity enters the Release Regulator module, it gives up control of the specified regulator(s). Any entities waiting in queues for those regulators will gain control of the regulators immediately.

TYPICAL USES

- Releases control of a regulator. A regulator is a device that adds material or removes it from a tank (for example, a valve).

PROMPTS

Prompt	**Description**
Name	The module identifier.
Regulators:	
Regulator Type	Indicates how the regulator to release will be specified; defined as **Regulator**, **Regulator Set**, **Attribute**, or **Expression**.
Regulator Name	Name of the regulator to release.
Regulator Set Name	Name of the regulator set that contains the regulator to release.
Release Rule	Method of determining which regulator within a set to release. **Last Member Seized** and **First Member Seized** will release the last/first member from within the set that was seized. **Specific Member** indicates that a member number or attribute (with a member number value) will be used to specify the member to release.
Set Index	Index into the regulator set.
Attribute Name	Name of the attribute that stores the name of the regulator to release.
Expression	Expression that evaluates to the name of the regulator to release.

Data module

Data modules are the set of objects in the spreadsheet view of the model that define the characteristics of various process elements, such as resources and queues.

Regulator Set module

DESCRIPTION

The Regulator Set module defines a set of regulators to select from and use in other Flow Process modules (Seize Regulator, Release Regulator, Regulate, or Flow).

TYPICAL USES

- Defines a grouping of regulators for logical purposes (for example, to select an available member from the set using a selection rule)

PROMPTS

Prompt	Description
Name	The set's identifier.
Regulator Name	Regulator member in set.

7 Arena Visual Designer

Introduction

Arena Visual Designer is the visualization application inside Arena for quickly and easily designing dashboards for advanced business graphic reporting and 3D animation in customizable display windows. Dashboards create and display business graphics using controls such as Picture; Text Box; Level; Column, Bar and Pie charts; Histogram; Plot; and Scoreboard that will update dynamically as the simulation runs. The Scene Editor allows you to create realistic 3D animations of your Arena model. Dashboards and the Scene are visible simultaneously for professional presentations.

Arena Visual Designer offers:

- Intuitive drag-and-drop environment to build dashboards and animations quickly and easily
- Greater flexibility and visuual control of your models
- Use of common charting controls to communicate your simulation results
- Unlimited dashboard functionality

Visual Designer is a separate application that can be launched in Arena by selecting either the Visual Designer button from the Standard toolbar or by selecting Visual Designer from the Tools menu.

Your software license will determine which features of Arena Visual Designer will be available to enhance your model’s visualization. New project options are outlined in the table below.

Dashboard Project	Creates a Visual Designer project that contains an initial Dashboard window.
3D Project	Creates a Visual Designer project that contains a 3D Scene. Dashboard windows may be added, if desired.
3D and Dashboard Project	Creates a Visual Designer project that contains both a 3D Scene and an initial Dashboard window.

Visual Designer windows

Before we create a simple dashboard or a 3D animation, let's examine the components that comprise the Visual Designer.

WHAT IS A DASHBOARD?

Visual Designer's dashboards are customizable graphic display windows for presenting business information and statistics. Graphs, data, charts, text or images are placed in the workspace to present important business reporting from disparate sources in a concise visual display. This content may deliver both static and dynamic information that can enhance business presentations and aid in decision making.

WHAT IS A CONTAINER?

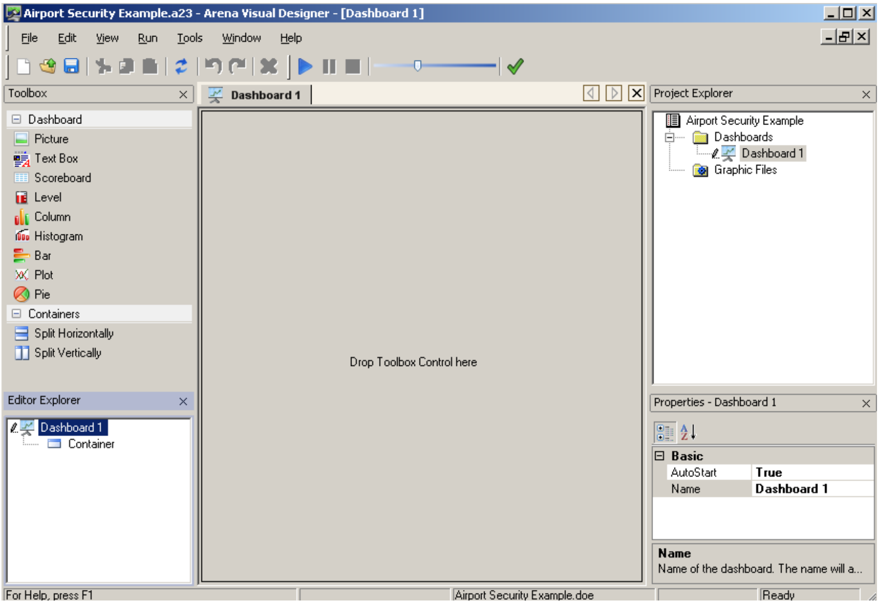
Containers are the partitions of the Visual Designer workspace that are used to build a dashboard or a Views Config display. By default the initial instance starts with one container. Because each container can only display a single component, the Containers tools in the Toolbox are used to divide the dashboard or Views Config window into more than one container. As you drag and drop the Containers tools (Split Horizontally or Split Vertically) onto the workspace, you further divide it into multiple containers.

Dashboard controls (graphs, data, charts, text or images) or Saved Views are dragged into the containers to create the desired display. Controls and context-sensitive menu options are described in the online help.

Dashboard window

The Dashboard window is an editor that allows you to view and edit the dashboards that have been added using the Project Explorer.

The Dashboard window is used to design each dashboard. By default, each new dashboard has one container in the Dashboard window, which is the partition space used for building the display. A gray text message in the empty dashboard, "Drop Toolbox Control here," guides you to begin creating a dashboard by dragging and dropping a control into the empty container.



AutoStart	True
Name	Dashboard 1

Name
Name of the dashboard. The name will a...

For Help, press F1 | Airport Security Example.doe | Ready

When a Dashboard editor is active, two categories of tools are available in the Toolbox window: *Dashboard* and *Containers*.

The *Containers* tools are used to divide the window into more than one container.

As you drag and drop the Containers tools into the Dashboard window, it will be divided into multiple containers. The tools available are:

Split Horizontally

Split Vertically

The *Dashboard* controls in the Toolbox window are selected to be placed into one or more of the containers on a dashboard.

When a specific control is dragged from the Toolbox into the dashboard, a default value for the property name is provided for the control instance. For example, if you place two plots in the dashboard, the first instance will automatically be named “Plot 1,” and the second “Plot 2.” The name can later be changed to something more meaningful.

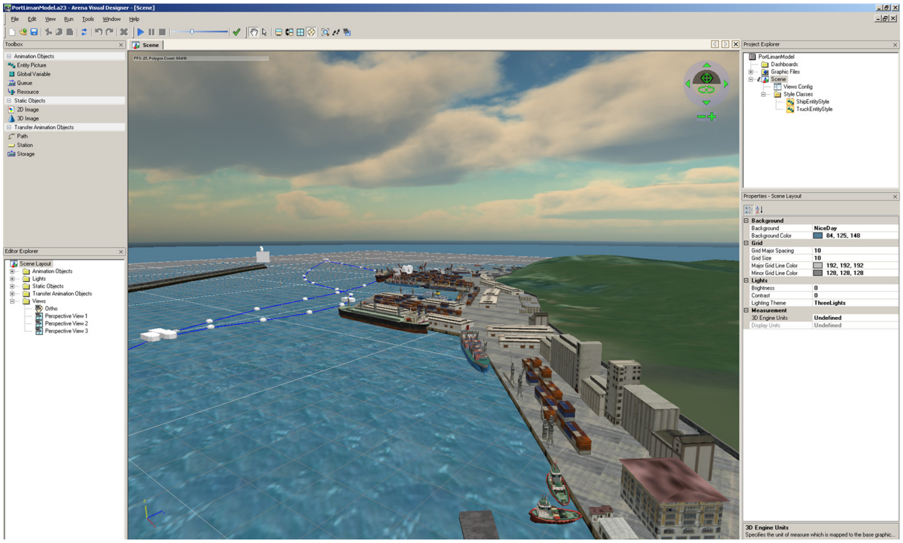
Refer to “Dashboard Controls” in the online help for examples of controls and properties.

Scene window

The Scene window is an editor that allows you to view and edit the 3D scene for a project. If you specify a new project to include a 3D Scene, a Scene node will be added to the Project Explorer by default. Each node that is created in a project can be selected by clicking on the corresponding tab on the window bar or by double-clicking on the desired node in the Project Explorer list.

The initial scene will be displayed in perspective view and will be empty. You can begin creating a scene by dragging and dropping element items from the Editor Explorer into the scene. The Editor Explorer contains a list of all element items that may be animated in the scene. Items placed using the Editor Explorer will have the element information automatically defined (e.g., the name of the resource, or the name of the queue). The object can be further configured using the Properties tool window for that object.

You can also drag and drop generic instances of animation objects from the Toolbox into the scene to be further defined using the Properties window.



FPS: 25, Polygon Count: 66416

Background	
Background	NiceDay
Background Color	84, 125, 148
Grid	
Grid Major Spacing	10
Grid Size	10
Major Grid Line Color	192, 192, 192
Minor Grid Line Color	128, 128, 128
Lights	
Brightness	0
Contrast	0
Lighting Theme	ThreeLights
Measurement	
3D Engine Units	Undefined
Display Units	Undefined

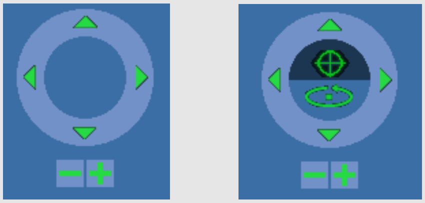
3D Engine Units
Specifies the unit of measure which is mapped to the base graphic...

What are Helper Objects?

Helper objects in Visual Designer are used to navigate in the scene and to edit objects in the scene. There are two main categories of helper objects: *Scene* and *Edit Object*. (See the online help for information on tool control functions.)

Scene Helper Object

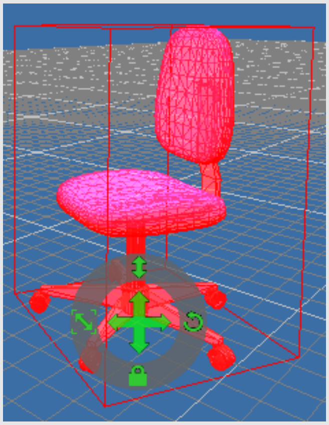
This widget is used to navigate and move throughout the 3D scene. It is located in the top right of the Scene when in Edit mode. Depending on the current view, the Scene Helper Object will vary slightly.



Scene Helper Objects: Ortho View on left, Perspective View on right

Edit Helper Object

This widget is used to edit objects (animated objects, static objects, transfer animation objects) in the scene. The Edit Helper Object will vary slightly depending on the type of object being edited. When you are in Edit Mode and you select an object in the scene, the Edit Helper Object will appear, overlaying the object, as below:



Edit Helper Object overlaying a chair object in the scene

Style Class window

Four animation object types require a style class to be associated with them. For each, there is an associated style class type. A style class is a generic representation of a resource, a transporter, an entity picture or a global variable.

The Style Class window is an editor that allows you to view and edit a style class for a resource, transporter, global variable or entity picture. The style class specifies the graphic file(s) associated with either the different trigger values of a global variable, the states of a resource and tranporter, or the states of the entity assigned to a given entity picture.

Style Classes allow you to manage changes to your animation objects quickly and easily. For example, in the case of a resource style class, if a manufacturing floor has 20 injection mold machines that are exactly alike, then only one style class needs to be defined — such as "InjectionMold_MachineStyle." This same resource style class can then be used for any injection mold machine. If any animation changes are needed, simply changing the style class will effect all resources using that style class.

Views Config(uration) window

The Views Config(uration) window is an editor that allows you to specify which camera views will be displayed for the 3D animation when the visualization is in Run mode.

When a new project is initiated, if you specify it to include a 3D scene, a Scene node will be added to the Project Explorer by default. Under the Scene object is the Views Config node. Double-click Views Config to open the Views Config editor window or right-click on Views Config and select Open Editor.

When Views Config is the active editor, two categories of tools are shown: *Views* and *Containers*.

Just like the Dashboard control, the *Containers* tools are used to divide a Views Config window into more than one container to display the number of scene compartments you want to reveal during the animation run. By default, the Views Config has one initial container. When the Views Config editor is empty, the (gray) text in the container states "Drop Toolbox Control here" to help guide you to create a view configuration.

The *Views* tools are used to select a saved view to place into one or more of the containers on the Views Config window.

Views can be created when the Scene is the active editor using the Save View icon or selecting Save View from the View menu. The layout can be comprised of multiple containers, each containing a previously saved view of the scene. Selecting Save View will save the current camera view in either orthogonal or perspective mode. See the Views section on page 150 for additional details.

When you have completed your configuration and are ready to start your 3D animation, select the Go button from the toolbar, and each container will show the animation from its unique view.

Visual Designer projects that contain a Scene window, but do not have a Views Config defined will display the 3D animation during runtime using the current view of the Scene window.

Editor Explorer

The Editor Explorer tool window provides a hierarchical listing (using the tree view control) of the active window's content. This window is context-sensitive based on the current active window. There are four main types of Editor Explorer views based on the active editor: Scene, Dashboard, Views Config, and Style Classes.

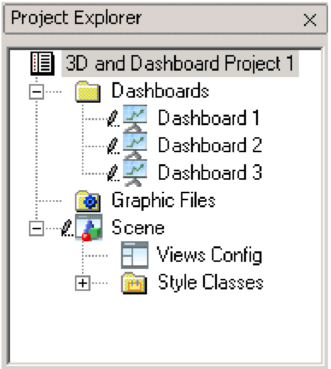
If there is no editor open and active for a project, the Editor Explorer displays a gray text message, "There is currently no data to display."

Project Explorer

The Project Explorer tool window provides a hierarchical listing (using a tree view control) of the project file's content. It can be used for opening or adding new Dashboard or Scene windows and for controlling the current active editor window. When the project is opened as either a 3D Project type or a 3D and Dashboard Project type, the tree structure for this explorer is divided into three main folder areas: *Dashboards*, *Graphic Files* and the *Scene*. Nested beneath the Scene object are one Style Classes folder and one Views Config object. All Style Classes, regardless of type (entity, resource, and so on) will be listed alphabetically in this folder.

When only a Dashboard Project type is opened, the Scene entries will not be displayed.

Dashboard instances, graphic files and scene objects for each project are managed from the Project Explorer tool window and are listed in alphabetical order within each section, as in the example below.

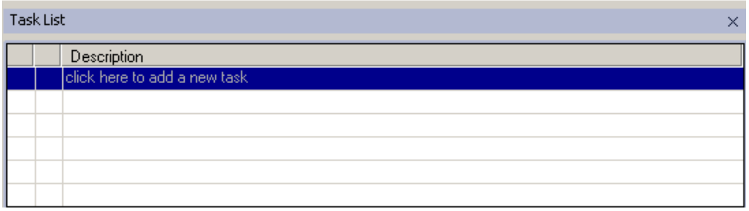


Project Explorer

- 3D and Dashboard Project 1
  - Dashboards
    - Dashboard 1
    - Dashboard 2
    - Dashboard 3
  - Graphic Files
  - Scene
    - Views Config
    - Style Classes

Task List

The Task List tool window provides a list of application errors and warnings to be addressed by the user. In addition, user-specified activities can also be added to aid in project organization and task sharing. This tool is added to the window view by selecting **View > Tool Windows > Task List**. See the topic "Task List" in online help for additional details on this tool.



Task List

		Description
		click here to add a new task

Properties

The Properties tool window provides a listing, using a tree view control, of properties for selected objects in the project. The Properties tool window has two buttons at the top, *Categorized* and *Alphabetical*, that control the listing order of the properties. By default, the properties are displayed alphabetically by category.

When a specific property is selected, a description of that property appears at the bottom of the tool window.

DASHBOARD PROPERTIES

Click the name of the dashboard in the Editor Explorer or Project Explorer for its properties to become visible in the Properties tool.

When an object is selected that does not have associated properties, the properties window displays the message: “No properties associated with this object.”

Individual dashboard windows, controls (e.g., plots, levels), and the project itself are examples of objects that have properties associated with them that can be customized.

An undefined container is an example of an object that does not have any properties associated with it.

Scene properties

Select *Scene Layout* in the Editor Explorer or *Scene* in the Project Explorer for their properties to become visible in the Properties window. Use the Scene Properties to adjust the background color or add a user-defined texture such as clouds in the sky or a sunset view. These properties also allow such changes as grid sizing, display measurement and lighting.

Characteristics of the objects in the scene (e.g. resources, queues, etc.) can be customized by altering the properties that are associated with the selected objects.

Toolbox

The Toolbox is a context-sensitive tool window that provides a listing of objects and controls used for inserting objects into the current active editor window (either a Dashboard, the Scene, the Views Config, or a Style Class).

If there is no editor window open and active for a project, this window displays the message “There are no suitable items available.”

Dashboard controls

When a dashboard is the active editor, two categories of tools are shown: *Dashboard* and *Containers*, each with a sub-list of available controls.

An existing control may be redefined as another control type. After a control has been placed in a container and its properties defined, the control type can be changed by dragging and dropping a new control over the existing one. Visual Designer will then attempt to inherit all properties from the old control type into the new control type. To place a new control without inheriting any of the defined properties, press the Shift key while dragging and dropping the new control on top of the existing one. This action is equivalent to deleting the old control and placing a new one. Refer to “Dashboard Controls” in the online help for examples.

Bar control

The Bar control is used to display one or more series of model data (variables and expressions) in a bar chart.

Column control

The Column control is used to display one or more series of model data (variables and expressions) in a column chart.

Histogram control

The Histogram control is used to assess the probability distribution of a given variable by depicting the frequencies of observations occurring in certain ranges of values.

The histogram provided with Visual Designer is based on observational data values; the histogram available in Arena is based on observations, but those observations are time weighted.

If looking at the number in queue, the histogram in Visual Designer will graph just the individual observation values. In Arena, the histogram will time weight the observed values, which means that the longer an expression holds a certain value, the greater the weight it is given.

For example, if the observed values for the number in queue range from 0 through 8, the Visual Designer histogram will update the bins with the number of instances at the observed value. In Arena, the amount of time spent at each observed value is factored in as part of the observation. If the majority of the time the number in queue is equal to 3, then that observation will have a greater precedence and be weighted accordingly.

Level control

The Level control is used to display a single model variable or expression as a column or bar.

Picture control

The Picture control is used to display a graphic image inside a dashboard. The following types of image files are supported:

- Bit mapped picture (.bmp)
- Graphics interchange format (.gif)
- Joint photographic Experts group (.jpg, .jpeg)

- Portable network graphics (.png)
- Icon graphics file format (.ico)
- Extended (Enhanced) Windows Metafile Format (.emf)
- Windows metafile (.wmf)

Pie control

The Pie control is used to display a series of model variables and expressions in a pie chart.

Plot control

The Plot control is used to monitor and plot one or more model variable(s) or expression(s). The independent (X-axis) variable plotted is simulated time.

Scoreboard control

The Scoreboard control is used to display model variables and expressions similar to a spreadsheet. The number of horizontal and vertical cells is limited to 500.

Cell contents can display descriptive text or the current value of a model variable or expression. To display the value of a model variable or expression, the contents of the cell must be preceded by an equal (=) sign; (for example, =TNOW).

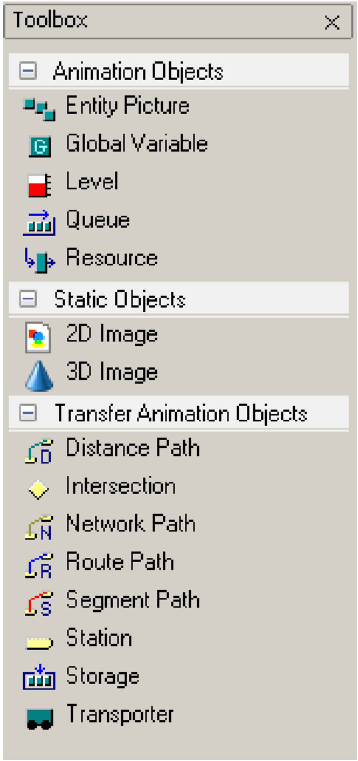
Text Box control

The Text Box control is used to present narrative text on the dashboard. This feature does not allow the display of a model variable or expression. To do so, use the Scoreboard control.

To add text to the control, open the Text Editor window by double-clicking inside the container, clicking on the ellipsis in the Text property, or right-clicking in the container and selecting “Text Editor.” The window has the label “Text Editor” and contains buttons to help you edit the text.

Scene controls

When the Scene is the active editor, three categories of Toolbox tools are displayed: *Animation Objects*, *Static Objects*, and *Transfer Animation Objects*.

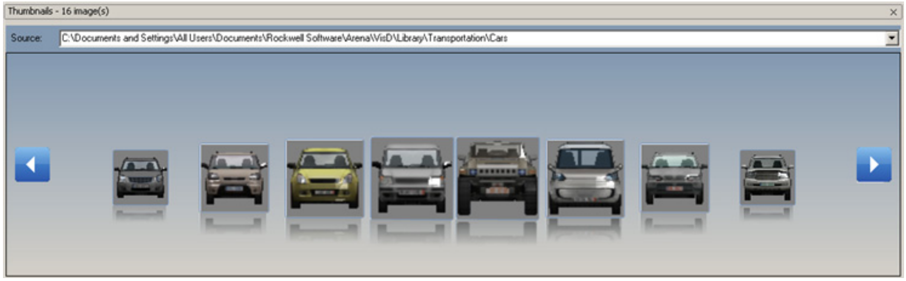


The *Animation Objects* and *Transfer Animation Objects* tools can be used to place objects in the 3D scene. This method places generic instances of an animated object, meaning that the associated Arena element or expression must be defined by the user. This is the only method to place a global variable in the 3D scene or to place any path. See “Animation Objects” on page 147 and “Transfer Animation Objects” on page 149 for more details.

The *Static Objects* tools are used to place static 2D and 3D objects into the scene. To place a static object, simply drag the generic icon from the Toolbox into the scene. After placing the object, the “File Name” fields need to be defined. See “Static Objects” on page 149 for more details.

Thumbnails

The Thumbnails tool window is useful for viewing 2D and 3D images that can then be used when creating a style class or dragged into place in the scene layout. The image below shows the thumbnail carousel of pictures in the “Cars” library being previewed.



Thumbnails - 16 image(s)

Source: C:\Documents and Settings\All Users\Documents\Rockwell Software\Arena\VisD\Library\Transportation\Cars

Getting started with Visual Designer

Working with Dashboards

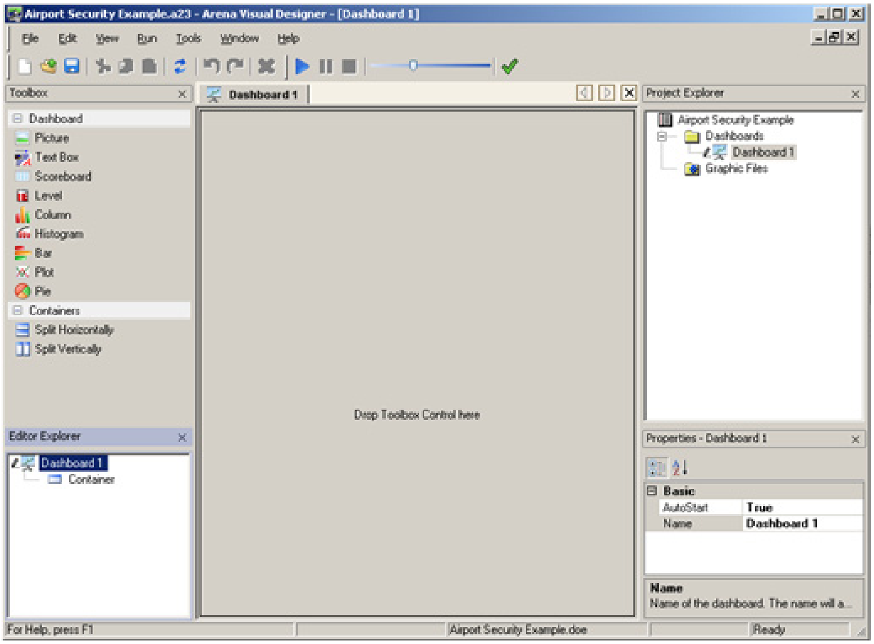
Creating a basic dashboard

1. For this example, we will return to the *Airport Security Example.doe* model. This model was extracted to the Examples folder when you installed Arena. Note that the corresponding Visual Designer project file “Airport Security Example.a23” has already been created. When a corresponding Visual Designer project exists, the file will be automatically opened when starting Visual Designer. To create this visualization project from scratch, you can either rename the existing Arena model or use the model you may have created from Chapter 2. With the airport model open, go to **Tools > Arena Visual Designer** to start the application or click the Arena Visual Designer icon ( ).

   When Arena Visual Designer opens, the application prompts you to name the project. Visualization projects are saved with an *.a23* file extension (*Airport Security Example.a23*). The default is the name of the current Arena model with the .a23 extension. The default directory is the same as the current directory of the saved Arena model. It is recommended that visualization projects be saved to the same directory as the associated Arena *.doe* model file. Only one Visual Designer project may be open at a time for editing.

Since we will be starting with a dashboard for our Airport Security Example, Select “Dashboard Project” from the New Project dialog box. Click **OK** to proceed to Arena Visual Designer.

2. When opening a new Visual Designer Dashboard Project, the initial dashboard window (shown below) is displayed with “Drop Toolbox Control here” shown in the center of the dashboard. On the left are the Toolbox and Editor Explorer and on the right are the Project Explorer and Properties window. Use the Toolbox to design the layout and to add controls to the dashboard.



AutoStart	True
Name	Dashboard 1

Name
Name of the dashboard. The name will a...

3. The dashboard may be segmented to hold multiple containers by dragging the Split Horizontally or Split Vertically controls into the dashboard container window.

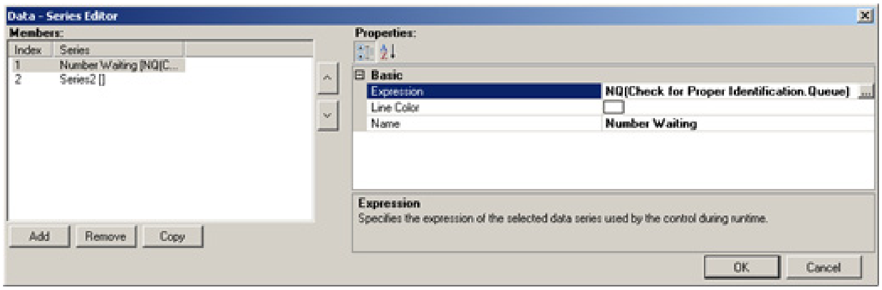
Alternatively, segment the dashboard by placing the mouse pointer in the dashboard window, right-clicking and selecting either the Split Horizontally or Split Vertically options.

For this simple example, you will create 1 plot of the number of passengers waiting in security.

4. From the Toolbox, drag the Plot control into the container that states "Drop Toolbox Control here." Once a control is placed, the properties associated with it can be edited in the Properties window.

5. In the Properties window, click on the *Data* properties right column, then select the ellipsis (...). This will be the first field listed in the Properties window. The *Data-Series Editor* dialog box will open.

6. Highlight **Series1** from the Members list. Select the right column of the *Expression* property, then initiate the Expression Builder by selecting the ellipsis. Use the tree view to build the expression for the current number in queue for the *Check for Proper Identification.Queue*, or enter the expression **`NQ(Check for Proper Identification.Queue)`** directly. Update the Name field to read **`Number Waiting`**. Your Data - Series Editor should look like the example below.



Index	Series
1	Number Waiting [NQ(C...
2	Series2 []

Basic

Expression	NQ(Check for Proper Identification.Queue)
Line Color	□
Name	Number Waiting

Expression
Specifies the expression of the selected data series used by the control during runtime.

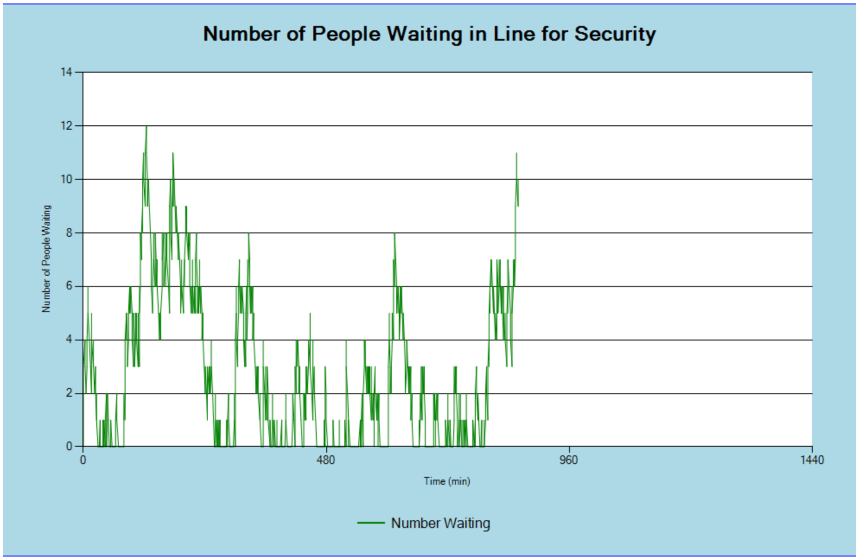
7. Highlight **Series2** from the Members list. Select the **Remove** button.

8. Select OK to confirm updates and to close out of the Data - Series Editor.

9. Add a title to your chart by updating the Title field in the Properties window to **`Number of People Waiting in Line for Security`**.

10. Change the Background Color field to something other than white for a more vibrant graph.

11. Adjust other properties, such as your *x* and *y* axis titles until you are satisfied with the appearance of your graph. Below is an example of the finished plot. You can also view and run this completed example from the Arena Examples folder (*Airport Security Example with Visual Designer.doe/.a23*). Select the Dashboard tab or double-click the Dashboard in the Project Explorer to navigate to this plot example.



To run a dashboard, save the project and select **Go** from the Visual Designer Run menu or select the Go icon (▶) from the toolbar. To activate the animation of the dashboards, you must start the model run from within Visual Designer.

Once the model run is complete, or you select Stop from the Run menu, you may resume editing the dashboard.

Remember, as with all applications, it is wise to save your work frequently.

Now that you have a basic understanding of the dashboard environment and how to navigate, let's now look at some of the components of the Dashboard window.

Editing dashboards

Start Arena, open the desired model and then start Visual Designer. A Visual Designer project will open that will be associated with the model.

When editing an existing dashboard, it is recommended that the model file associated with the dashboard be opened first. Doing so makes it easier to reference the model when updating expressions and information that is directly tied to the model data.

If multiple dashboards exist, use the Project Explorer to navigate among the dashboards or select the tabs associated with the different dashboard nodes.

The Editor Explorer can be used to navigate within a dashboard from one control to another or the controls can be selected directly to edit them.

Only the containers that are highlighted, with or without controls, can be edited.

Adding dashboards

Visual Designer allows you to create additional dashboards through the Project Explorer. To add a new dashboard, simply right-click the Dashboard folder within the Project Explorer and select Insert Dashboard. If the Project Explorer is not visible, click **View > Tool Windows > Project Explorer**.

Deleting dashboards

When no longer needed, dashboards can be removed from Visual Designer through the Project Explorer. First select the dashboard to be removed and then right-click to select the Delete option. Visual Designer will prompt with the question, "Are you sure you want to remove the node 'Name of Dashboard'?" To permanently delete the dashboard, choose Yes. Once dashboards are removed, they cannot be recovered unless earlier versions of the dashboard project were saved (and not overwritten).

Dividing dashboards

Dashboards can be divided into separate containers by dragging the Split Horizontally or Split Vertically controls from the Toolbox to the dashboard or by right-clicking in the dashboard and selecting the option to split the container.

Running your model and dashboard

When running a model in Arena, only the animation within the main Arena application will animate. The dashboards within Visual Designer will not be animated. To view the dashboard animation within Visual Designer, the model run must be started from within Visual Designer.

To run the model from Visual Designer, select **Run>Go** or press F5.

When a model is running, it can be paused or the model run can be ended within Visual Designer.

When a model run is initiated from Visual Designer, by default the Arena animation is disabled. Arena is running in *Fast-Forward* mode where the animation is not updated on the screen. This behavior is controlled by the *Enable Arena Animation* option located on the Run menu. When this option is disabled (the default), anytime the *Go* button is initiated from Visual Designer, Arena is put into Fast-Forward mode and any animation define in the .doe file is not updated. Only the Visual Designer animation is enabled.

When *Enable Arena Animation* is enabled, pressing the Go button from Visual Designer places Arena in Go/standard run mode and any animation defined in the .doe file is displayed.

From Arena, the Go, Step, Fast-Forward, Pause, Start Over and End options can be activated.

Note that if the visualization is paused and Arena is made the active window, pressing *Go* from Arena always places Arena in standard run mode, displaying the .doe file's animation regardless of the *Enable Arena Animation* setting in Visual Designer. This setting controls the mode Arena will be placed in (Fast-Forward or Go/standard run mode) when *Go* is selected from Visual Designer only.

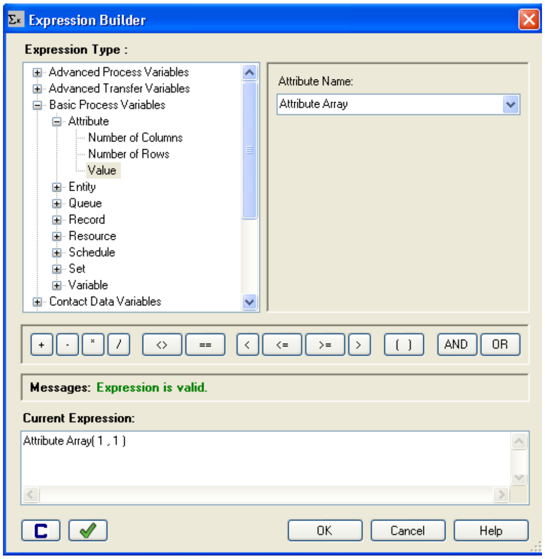
Models that are designated to run in Batch mode will not animate in either Arena or Visual Designer.

Expression entry

Expression Builder

The Visual Designer Expression Builder is a tool that can be used to build expressions. The performance of the tool resembles the Expression Builder in Arena.

Invoke the tool by selecting the Expression Builder menu option when right-clicking an edit field that requires an expression, or when an ellipses is clicked in an expression field for a dashboard control, or when a dashboard control is double-clicked. The Expression Builder dialog box is also resizable.



Attribute Name: Attribute Array

Messages: Expression is valid.

Current Expression:

Attribute Array(1 , 1)

The Expression Builder also allows you to check the validity of an expression. If there is a problem with the checked expression, the Expression Builder will identify a specific error and a specific location in the expression and will display the error statement in the Messages box. If the expression is valid and proper, the message “Expression is valid” will be shown.

Intellisense

Intellisense is a common tool used in programming languages to auto-complete expressions or lines of code quickly and efficiently.

Intellisense is being introduced in Arena Visual Designer for use with the @ROW and @COL functions. The @ROW and @COL functions allow the user to define the starting values for an array. Future releases will include other Arena Variable functions; for example, NR, NC, TNOW.

Intellisense is triggered in the Scoreboard control when a user enters the @ symbol or selects the Ctrl+Spacebar. Below is the information on the use of the @ROW and @COL functions:

@ROW – function to define the beginning row value for an array
@COL – function to define the beginning column value for an array

Refer to the Intellisense online help for several example scenarios using this tool.

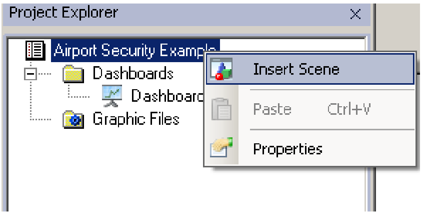
Working with 3D scenes

The following section walks through the process of creating a simple 3D scene. You must have activation for Arena to continue this portion of the tutorial.

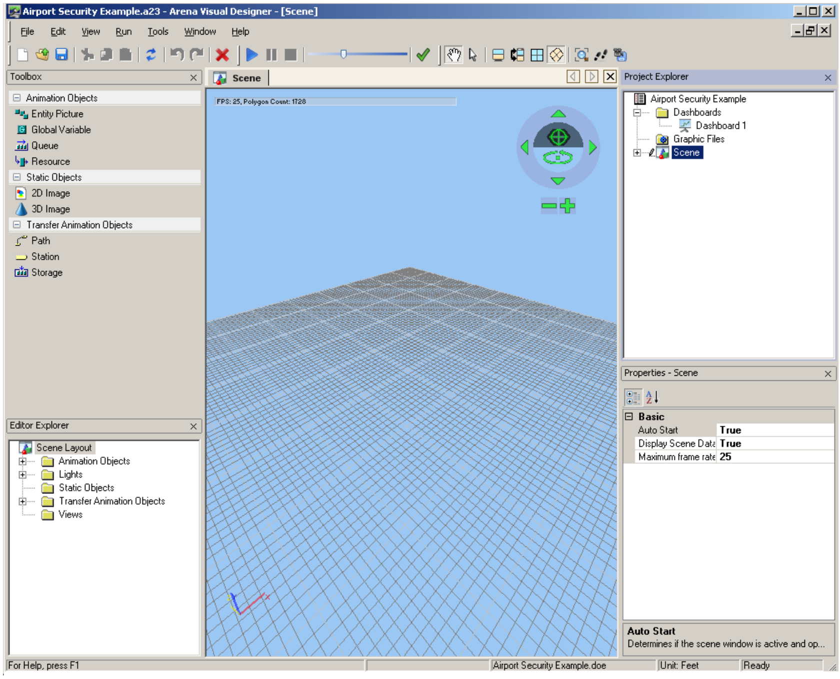
1. For this example, we will return to the *Airport Security Example.doe* model. This model is located in the example folder where you installed Arena. Note that the corresponding Visual Designer project file “Arena Security Example.a23” has already been created. When a corresponding Visual Designer project exists, the file will be automatically opened when starting Visual Designer. To create this visualization project from scratch, you can either rename the existing Arena model or use the model you may have created from Chapter 2. With the airport model open, go to **Tools>Arena Visual Designer** to start the application or click the Arena Visual Designer icon ( ). Select either 3D Project or 3D and Dashboard Project. The application prompts you to name the project. Visualization projects are saved with an *.a23* file extension. The default is the name of the current Arena model with the *.a23* extension. The default directory is the same as the current directory of the saved Arena model. It is recommended that visualization projects be saved to the same directory as the associated Arena *.doe* model file. Only one Visual Designer project may be open at a time for editing.

   Click **OK** to proceed to Visual Designer.

   Note: You may already have created a Visual Designer project for this model if you completed the Dashboard tutorial earlier in this chapter. To add a Scene to your existing Dashboard project, right-click on Airport Security Example in the Project Explorer and choose **Insert Scene**.

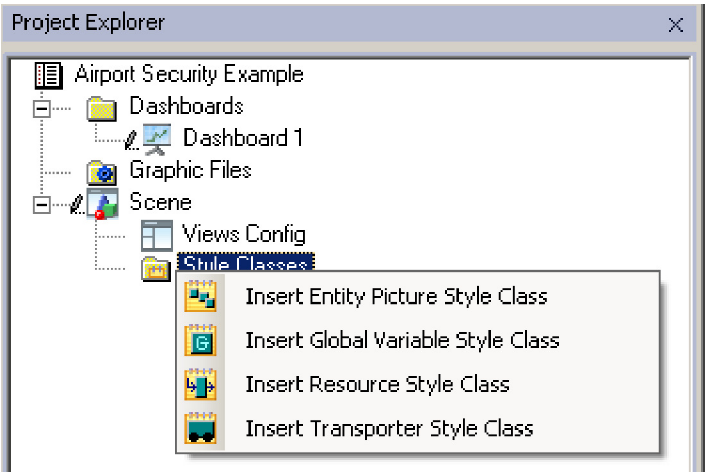
2. A new 3D Visual Designer project displays an initial Scene window in perspective view. On the left are the Toolbox and Editor Explorer and on the right are the Project Explorer and Properties window.



Auto Start	True
Display Scene Data	True
Maximum frame rate	25

Auto Start
Determines if the scene window is active and op...

3. Let’s first define the animation for the Transportation Security Officer resource. In the Project Explorer window, identify the Style Classes folder under the Scene node. Right-click on Style Classes and select **Insert Resource Style Class**.

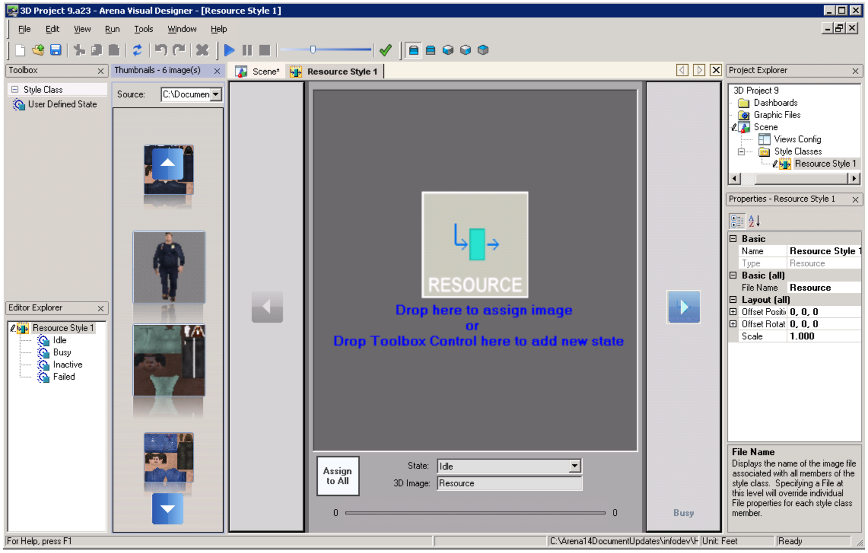

Project Explorer
Airport Security Example
Dashboards
Dashboard 1
Graphic Files
Scene
Views Config
Style Classes
Insert Entity Picture Style Class
Insert Global Variable Style Class
Insert Resource Style Class
Insert Transporter Style Class

4. In the Resource Style Class editor, you will define the animation for your resource. The easiest way to assign a 3D image to resource states is with the Thumbnails window which will be automatically opened when you create or edit a style class.

   a. From the Thumbnails window, navigate to the installed Visual Designer "People" image library.
   b. Drag an image from the People library to the Resource Style Class editor area marked as "Drop here to assign image or Drop Toolbox Control here to add new state." You can also assign your resource to your own custom image. Simply open the Thumbnails to where your image is stored and drag the image onto the editor.

c. Drag and drop the image in the box that states “Drop here to assign all images” to assign the same image to every resource state (Idle, Busy, Inactive, Failed).

d. Once your image is assigned, close the Style Class editor.



Drop here to assign image
or
Drop Toolbox Control here to add new state

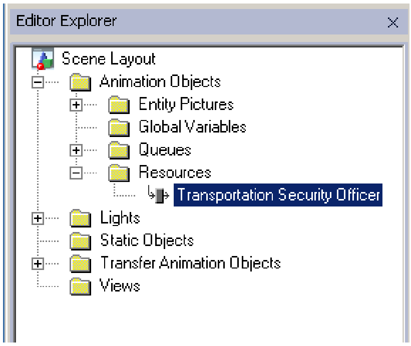
State: Idle
3D Image: Resource

Basic	
Name	Resource Style 1
Type	Resource
Basic (all)	
File Name	Resource
Layout (all)	
Offset Positi	0, 0, 0
Offset Rotat	0, 0, 0
Scale	1.000

File Name
Displays the name of the image file associated with all members of the style class. Specifying a File at this level will override individual File properties for each style class member.

For Help, press F1 | C:\Arena14DocumentUpdates\infodev\F | Unit: Feet | Ready

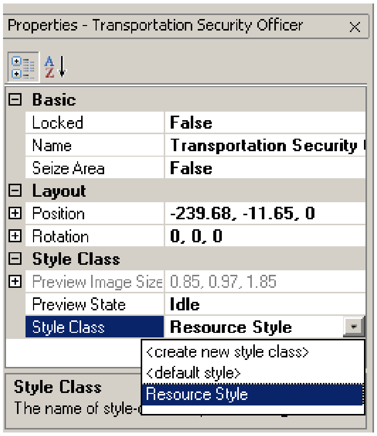
5. From the Editor Explorer, expand the Animation Objects folder and then the Resources folder. Select the **Transportation Security Officer** resource and drag it into the scene.



Use the Toolbox to drag and drop generic instances of animation objects (entity pictures, global variables, queues and resources), static objects (2D and 3D images), or transfer animation objects (paths, stations and storages) into the scene. Use the Properties window to further define any objects placed in the scene from the Toolbox.

The Editor Explorer will provide a hierarchical listing of the current objects in your project. For example, if you have defined resources in your project, select Animation Objects in the tree and then expand the Resources. You will see a list of all the resources in your Arena project. If a Resource has not yet been placed in your scene, the icon will appear gray.

6. With the newly added resource highlighted in the scene, update the Properties window and assign the Style Class field to be **Resource Style**.



Properties - Transportation Security Officer

⊟ **Basic**	
Locked	**False**
Name	**Transportation Security**
Seize Area	**False**
⊟ **Layout**	
⊞ Position	**-239.68, -11.65, 0**
⊞ Rotation	**0, 0, 0**
⊟ **Style Class**	
⊞ Preview Image Size	0.85, 0.97, 1.85
Preview State	**Idle**
Style Class	**Resource Style**

<create new style class>
<default style>
Resource Style

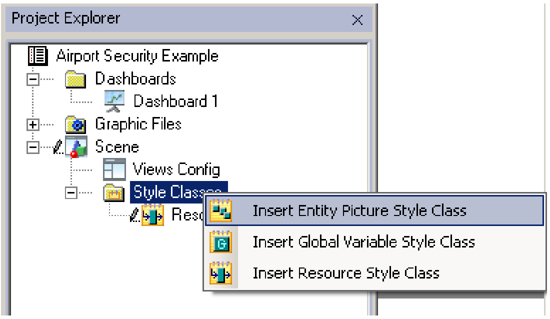
Style Class
The name of style-

The 3D environment is drag and drop. You can drag controls from the Toolbox into the scene and then define them using the Properties. You can also drag objects from the Editor Explorer into the Scene and further define them using the Properties.

7. Use the Helper Object to resize the resource so that it fits appropriately in the scene and is moved to the desired location.

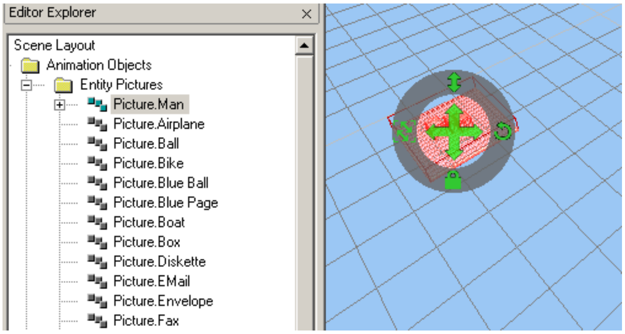
The Scene Layout includes properties for defining the scale of the 3D model. Adjusting the measurement settings allows you to design your model to an exact scale or simply to have a visual reference to approximate object size relationships. Measurement properties include 3D Engine Units, Display Units, Grid Major Spacing and Grid Size. See “Defining Your Scene Scale” in online Help for details.

8. Next you will define the entity image. In the Project Explorer window, identify the Style Classes folder under the Scene node. Right-click on Style Classes and select **Insert Entity Style Class**.



9. In the Entity Style Class editor, you will define the animation for your entity. Assign a 3D image to an entity is using the Thumbnails window in the same manner as you did in Step 4 when you assigned the resource image. From the Thumbnails window, navigate to the installed Visual Designer People image library and drag an image onto the editor. In this example, you will only assign one image to a single entity state. For more advanced animation, you can define entity sub-states and select different animation sequences when the entity is being seized, transferred or in a queue. Once your image is assigned, close the Style Class editor.

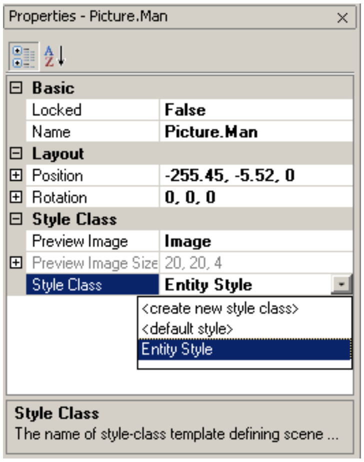
10. With the Scene as the active editor, from the Editor Explorer, expand the Animation Objects folder and then the Entity Pictures folder. Select the **Picture.Man** entity and drag it into the scene.



Editor Explorer

Scene Layout
- Animation Objects
  - Entity Pictures
    - Picture.Man
    - Picture.Airplane
    - Picture.Ball
    - Picture.Bike
    - Picture.Blue Ball
    - Picture.Blue Page
    - Picture.Boat
    - Picture.Box
    - Picture.Diskette
    - Picture.EMail
    - Picture.Envelope
    - Picture.Fax

11. With the newly added entity highlighted in the scene, update the Properties window and assign the Style Class field to be `Entity Style`. Use the Helper Object or Properties window to size your entity.



Properties - Picture.Man

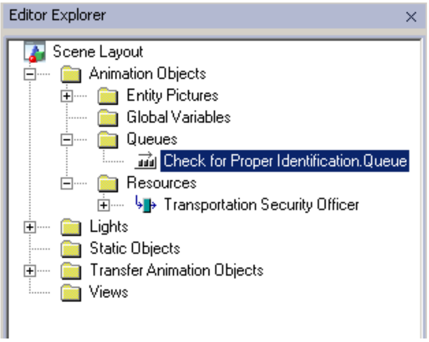
Basic	
Locked	**False**
Name	**Picture.Man**
Layout	
Position	**-255.45, -5.52, 0**
Rotation	**0, 0, 0**
Style Class	
Preview Image	**Image**
Preview Image Size	20, 20, 4
Style Class	**Entity Style**

<create new style class>
<default style>
Entity Style

Style Class
The name of style-class template defining scene ...

12. Now you will animate the queue in the scene. From the Editor Explorer, expand the Animation Objects folder and then the Queues folder. Select the **Check for Proper Identification.Queue** and drag it into the scene. Use the Helper Object and the Properties to position and size the queue as desired.

    You can select and move individual points on the queue by holding the Ctrl key and selecting the point in edit mode. With the point selected, you can move the point to a new position in the scene.



Editor Explorer

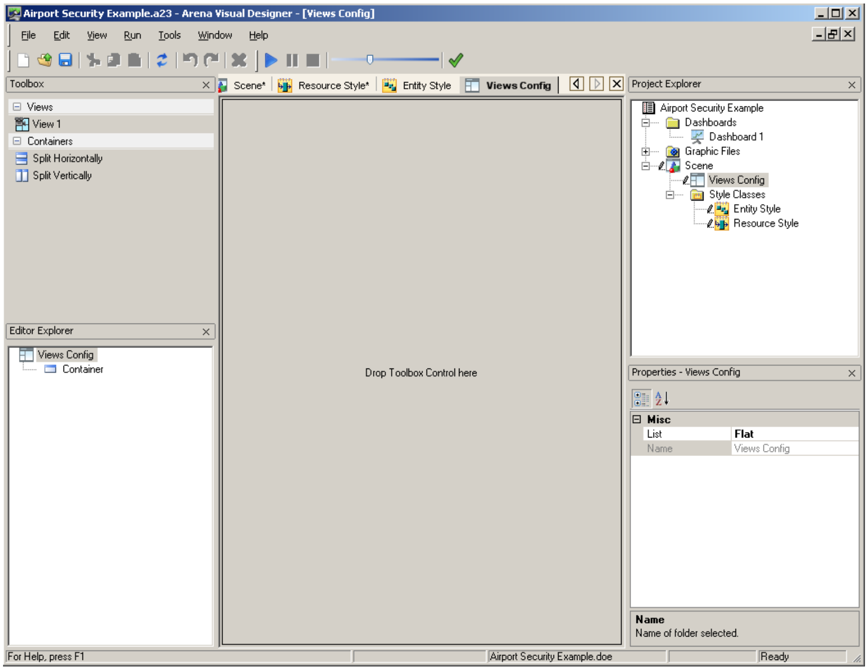
- Scene Layout
  - Animation Objects
    - Entity Pictures
    - Global Variables
    - Queues
      - Check for Proper Identification.Queue
    - Resources
      - Transportation Security Officer
  - Lights
  - Static Objects
  - Transfer Animation Objects
  - Views

13. Your model is almost ready to run. You only need to save a view and edit the Views Config.

    a. Navigate around the scene until you are satisfied with a particular view.
    b. Select the **Save View** icon from the Toolbar. Select **OK** from the dialog box.
    c. You will then be prompted to name your new view. Select **OK**.
    d. You will notice that your new view is now visible under the Views folder of the Editor Explorer.

The toolbar at the top of the application allows you to change the view of the application as you design your scene. You can design in the orthogonal view (top down), the perspective view or a split view. Refer to online help for additional information on Split View, View Synchronization, Perspective View and Ortho View.

14. In the Project Explorer, double-click **Views Config**. The Views Config allows you to configure the number of containers that will display the scene animation and the specific camera views in those containers. For this example, drag the view you created from the Toolbox into the active editor and then close the Views Config. Note that if the Views Config is left undefined, the scene animation will be displayed using the current scene view.



15. You are now ready to view your 3D animation or you can further enhance it by adding static background objects and fine-tuning your resource, entity and queue. When complete, select **Go** from the Run menu or press the Go button on the toolbar.

16. You can also view and run this completed example from the Arena Examples folder (Airport Security Example with *Visual Designer.doe/.a23*). Select the Scene tab or double-click the Scene in the Project Explorer to navigate to the 3D animation. Note that some additional static objects and concepts have been introduced in this model. Once you complete the basics from the above tutorial

and review the concepts in this manual, you should be able to navigate the entire example.

The Arena Visual Designer application does not include 2D or 3D design tools, but it supports the import of graphic files in both animated 3D .x, .3ds and .skp file formats as well as 2D .bmp and .jpg file formats.

Default 2D and 3D static images can be added to your scene by dragging the controls from the Toolbox onto the perspective or orthogonal view of the scene. Once the default image is placed in the scene, you can replace it by clicking on the default object to open the Properties toolbar. In the File Name drop-down box, you can then select any previously defined object in the model or you can browse your picture libraries or computer folders for a desired new image.

Now that you have a basic understanding of the 3D scene environment and how to navigate, let's now look at some of the components of the Scene window that are used when creating your animation.

Animation Objects

Animating using the Toolbox

The Animation Objects from the Toolbox (Entity Picture, Global Variable, Level, Queue and Resource) can be used to place objects in the 3D scene. This method places generic instances of an animated object, meaning that the associated Arena element or expression must be defined by the user. This is the only method to place a global variable in the 3D scene. Once the objects are placed in the scene and defined, they will appear in the corresponding folder (Entity Picture, Global Variable, Level, Queue and Resource) in the tree view of the Editor Explorer.

To place an object, simply drag the generic icon from the Toolbox into the scene. The scene view can be set as either perspective or orthogonal. Since these are generic instances of animated objects, the properties associated with the object will need to be defined.

When placing objects using the Toolbox, it is important to define those objects. For example, after placing a resource, the Name property needs to be defined with the symbol name of the resource identified in the model. After placing a global variable, the Expression property needs to be defined with the model expression that will animate the various defined triggers.

ANIMATING USING THE EDITOR EXPLORER

In the Editor Explorer, the Animation Objects folder contains four system subfolders: Entity Pictures, Global Variables, Levels, Resources, and Queues.

For entity pictures, resources, and queues, all elements defined in the associated model will appear under the corresponding folder in the Editor Explorer. Each object can be placed into the scene in either perspective or orthogonal view using drag and drop from the Editor Explorer. Once an object has been placed in the scene, the object name changes color in the tree view to indicate that it has been defined in the animation. Unplaced objects remain gray until they are added to the scene.

ANIMATING ENTITIES, RESOURCES, TRANSPORTERS, AND GLOBAL VARIABLES

To animate entities, resources, transporters, or global variables, you must first create a style class (a generic representation of either a resource, an entity picture or a global variable) to store your object images. From the Project Explorer, right-click on Style Classes and select Insert (Object) Style Class.

You can assign an image using the Thumbnails tool window (**View > Tool Windows > Thumbnails**) or by using Properties and pointing to a specific file. If you open the Thumbnails and browse to a folder with 3D objects, you will be able to view the image before you drag it to assign your object picture.

For more advanced object animation, you can assign different images depending on the object state. When defining an Entity Style Class, by right-clicking on the image in the Editor Explorer and selecting Split to sub-states, you can choose a specific sub-state (in Seize Area, in Queue, in Storage, in Transfer) and assign a unique image. Arena Visual Designer supports animated meshes so that each state can be assigned a specific animation sequence for realistic entity animation.

For example, you may want a person to appear as if they are walking along a route (in Transfer state). Another image may be required when they are in the queue waiting for the resource (in Queue state). You may want to change the appearance when they are then seized by the resource (in Seize state) or in a storage (in Storage state).

ANIMATING A QUEUE OR LEVEL

The queues you previously defined in your Arena model are automatically displayed in Visual Designer in the Editor Explorer when you browse **Animation Objects > Queues**. Simply drag the queue you want to animate from the Editor Explorer into the scene. You can also drag a queue from the Toolbox into the scene and then use the Properties box to define it. Use the Properties to further manipulate the queue. The default is a line queue, however, you can change it to a point queue. A point queue will allow you to specify where and in what direction the entity resides while in a particular position in the queue.

You drag a level from the Toolbox into the scene and then use the Properties box to define its behavior.

Transfer Animation Objects

The Transfer Animation Objects in the Toolbox can be used to place distance, intersection, station, storage, transporter, and path objects in the 3D scene.

This method places generic instances of an animated object, meaning that the associated Arena element or expression must be defined by the user. This is the only method to place any path. To place an object, simply drag and drop the generic icon from the Toolbox into the scene. The scene can be in either perspective or orthogonal view.

The existing distances, intersections, networks, segments, stations, storages and transporters defined in the model will be listed in the Editor Explorer in the Transfer Animation Objects folder. You can also drag and drop these specific instances of your objects into the scene. Once you drag your specific stations and connect them using the Path from the Toolbox, the Path will appear in the Paths folder in the Editor Explorer.

Static Objects

Static objects (such as trees, building structures, flooring and furniture) are not linked to the model logic but are useful to help make the scene animation appear more realistic.

From the Toolbox, the Static Objects tools are used to place static 2D and 3D objects into the scene. To place a 3D static object, simply drag and drop the generic default icon (cone) from the Toolbox into the scene in either the orthogonal or perspective view. After placing the object, use the File Name drop-down box to replace the default image with an appropriate graphic. You can select any object already defined or you can browse through your picture libraries or computer folders for a desired image. Then continue to define the remaining object characteristics in the Property fields for size, position, and more.

The Static Objects folder in the Editor Explorer contains all of the static images you place in the scene. Once an object is listed in the folder, you can simply drag and drop the graphic object directly into the scene to create duplicate instances. If you save the animation with the default names of multiple placements, you will notice that the system automatically appends the name with a number to create a unique name for each instance.

The **View > Find Object in Scene** menu item can help locate animated objects in any view (orthogonal, perspective, or split) in the scene. To use this feature, select the desired object in the scene Editor Explorer. In the View menu, select “Find Object in Scene.” If necessary, the current camera view will adjust to include the object specified, and the object will be highlighted in red and displayed in vector graphics.

This menu item is enabled when an appropriate object is placed in the scene and the scene is the active window.

This method allows you to locate all objects in the scene Editor Explorer with the exception of Lights or Views.

Lights

The Lights folder contains all the lighting sources used in the scene. The amount and type of lighting is defined by the scene’s Light Theme property. There are three light theme choices: three lights, two lights, and sun. If you select three lights or two lights, the lights folder of the Scene Editor Explorer will display two or three lights. The characteristics of each light can be modified using the Properties tool window.

Views

Arena Visual Designer permits you to display your animation scene in one or more windows using orthogonal or perspective views. Each view is displayed as if it were observed through a camera’s lens. As you create your animation and position a camera with a view you would like to save, you can select **View > Save View** from the menu, or click the Save View icon from the toolbar, to save and name each desired camera snapshot. This allows you to recall and display any saved view quickly and easily. Details on views, view settings and camera views can be found in the online help.

A Statistical Distributions

Arena contains a set of built-in functions for generating random numbers from the commonly used probability distributions. These distributions appear on pull-down menus in many Arena modules where they're likely to be used. They also match the distributions in the Arena Input Analyzer. This appendix describes all of the Arena distributions.

Each of the distributions in Arena has one or more parameter values associated with it. You must specify these parameter values to define the distribution fully. The number, meaning, and order of the parameter values depend on the distribution. A summary of the distributions (in alphabetical order) and parameter values is given in the table below.

Summary of Arena's Probability Distributions

Distribution		Parameter Values
Beta	BETA	Beta, Alpha
Continuous	CONT	$\mathrm{CumP}_1,\mathrm{Val}_1, \ldots \mathrm{CumP}_n,\mathrm{Val}_n$
Discrete	DISC	$\mathrm{CumP}_1,\mathrm{Val}_1, \ldots \mathrm{CumP}_n,\mathrm{Val}_n$
Erlang	ERLA	ExpoMean, k
Exponential	EXPO	Mean
Gamma	GAMM	Beta, Alpha
Johnson	JOHN	Gamma, Delta, Lambda, Xi
Lognormal	LOGN	LogMean, LogStd
Normal	NORM	Mean, StdDev
Poisson	POIS	Mean
Triangular	TRIA	Min, Mode, Max
Uniform	UNIF	Min, Max
Weibull	WEIB	Beta, Alpha

To enter a distribution in an Arena field, you type the name of the distribution (or its four-letter abbreviation) followed by its parameters enclosed in parentheses. You may use spaces around punctuation to help read the distribution. A few examples appear below.

UNIF(3.5, 6)

Uniform distribution with a minimum value of 3.5, a maximum value of 6

NORMAL(83, 12.8)

Normal distribution with a mean of 83, a standard deviation of 12.8

DISCRETE(0.3,50, 0.75,80, 1.0,100)

Discrete probability distribution that will return a value of 50 with probability 0.3, a value of 80 with cumulative probability 0.75, and a value of 100 with cumulative probability of 1.0. (See “Discrete Probability” for a description of these parameters.)

TRIA(10, 15, 22)

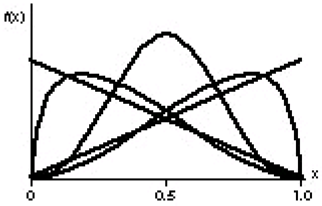
Triangular distribution with a minimum value of 10, mode (most likely value) of 15, and maximum value of 22

In the following pages, we provide a summary of each of the distributions supported by Arena, listed in alphabetical order for easy reference. Each includes the density or mass function, parameters, range, mean, variance, and typical applications for the distribution.

If you have existing data and want to select the appropriate distribution for use in your model, use Arena’s Input Analyzer. Click **Tools > Input Analyzer** to launch the program, or launch it from the Windows Start menu.

Beta(β, α): BETA(Beta, Alpha)

Probability Density Function



$$f(x) = \begin{cases} \dfrac{x^{\beta-1}(1-x)^{\alpha-1}}{B(\beta,\alpha)} & \text{for } 0 < x < 1 \\ 0 & \text{otherwise} \end{cases}$$

where β is the complete beta function given by

$$B(\beta, \alpha) = \int_0^1 t^{\beta-1}(1-t)^{\alpha-1}dt$$

Parameters Shape parameters Beta (β) and Alpha (α) specified as positive real numbers.

Range [0, 1] (Can also be transformed to [a,b] as described below)

Mean

$$\frac{\beta}{\beta + \alpha}$$

Variance

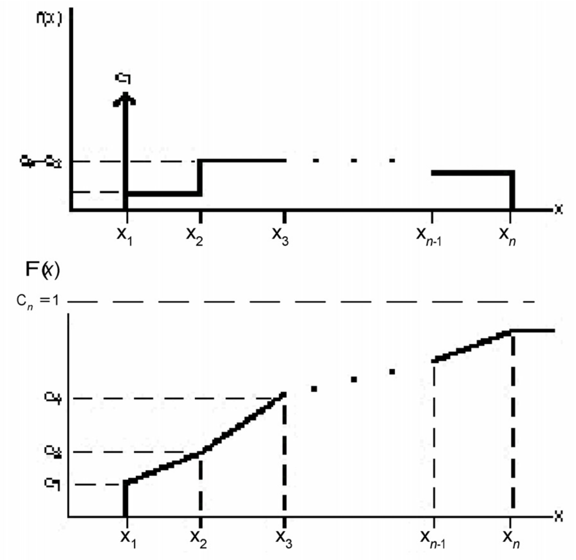
$$\frac{\beta a}{(\beta + a)^2(\beta + a + 1)}$$

Applications Because of its ability to take on a wide variety of shapes, this distribution is often used as a rough model in the absence of data. Also, because the range of the beta distribution is from 0 to 1, the sample X can be transformed to the scaled beta sample Y with the range from a to b by using the equation $Y = a + (b - a)X$. The beta is often used to represent random proportions, such as the proportion of defective items in a lot.

Continuous: CONTINUOUS(CumP$_1$, Val$_1$, . . ., CumP$_n$, Val$_n$) (c_1, x_1,. . ., c_n, x_n)

Probability Density Function

Cumulative Distribution Function



$$f(x) = \begin{cases} c_1 & \text{if } x = x_1 \text{ (a mass of probability } c_1 \text{ at } x_1\text{)} \\ c_j - c_{j\text{-}1} & \text{if } x_{j\text{-}1} \leq x < x_j, \text{ for } j = 2, 3, \ldots, n \\ 0 & \text{if } x < x_1 \text{ or } x \geq x_n \end{cases}$$

Parameters The CONTINUOUS function in Arena returns a sample from a user-defined distribution. Pairs of cumulative probabilities c_j (= CumP$_j$) and associated values x_j (= Val$_j$) are specified. The sample returned will be a real number between x_1 and x_n, and will be less than or equal to each x_j with corresponding cumulative probability c_j. The x_j's must increase with j. The c_j's must all be between 0 and 1, must increase with j, and c_n must be 1.

The cumulative distribution function $F(x)$ is piecewise linear with "corners" defined by $F(x_j) = c_j$ for $j = 1, 2, \ldots, n$. Thus, for $j \geq 2$, the returned value will be in the interval $(x_{j-1}, x_j]$ with probability $c_j - c_{j-1}$; given that it is in this interval, it will be distributed uniformly over it.

You must take care to specify c_1 and x_1 to get the effect you want at the left edge of the distribution. The CONTINUOUS function will return (exactly) the value x_1 with probability c_1. Thus, if you specify $c_1 > 0$, this actually results in a mixed discrete-continuous distribution returning (exactly) x_1 with probability c_1, and with probability $1 - c_1$ a continuous random variate on $(x_1, x_n]$ as described above. The graph of $F(x)$ above depicts a situation where $c_1 > 0$. On the other hand, if you specify $c_1 = 0$, you will get a (truly) continuous distribution on $[x_1, x_n]$ as described above, with no "mass" of probability at x_1; in this case, the graph of $F(x)$ would be continuous, with no jump at x_1.

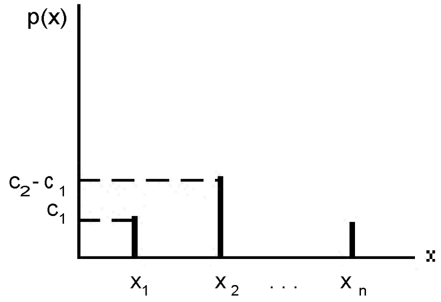
As an example use of the CONTINUOUS function, suppose you have collected a set of data $x_1, x_2, \ldots, x_n$ (assumed to be sorted into increasing order) on service times, for example. Rather than using a fitted theoretical distribution from the Input Analyzer, you want to generate service times in the simulation "directly" from the data, consistent with how they're spread out and bunched up, and between the minimum x_1 and the maximum x_n you observed. Assuming that you don't want a "mass" of probability sitting directly on x_1, you'd specify $c_1 = 0$ and then $c_j = (j - 1)/(n - 1)$ for $j = 2, 3, \ldots, n$.

Range $[x_1, x_n]$

Applications The continuous empirical distribution is often used to incorporate actual data for continuous random variables directly into the model. This distribution can be used as an alternative to a theoretical distribution that has been fitted to the data, such as in data that have a multimodal profile or where there are significant outliers.

Discrete: DISCRETE(CumP$_1$, Val$_1$, . . ., CumP$_n$, Val$_n$) (c_1, x_1, . . ., c_n, x_n)

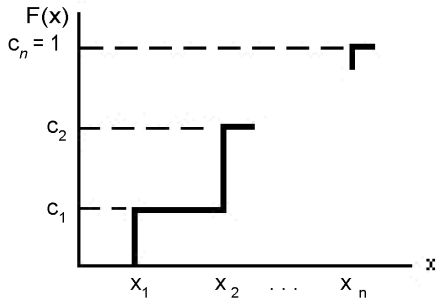
Probability Mass Function



$$p(x_j) = c_j - c_{j\text{-}1}$$

where $c_0 = 0$

Cumulative Distribution Function



Parameters The DISCRETE function in Arena returns a sample from a user-defined discrete probability distribution. The distribution is defined by the set of n possible discrete values (denoted by $x_1, x_2, \ldots, x_n$) that can be returned by the function and the cumulative probabilities (denoted by $c_1, c_2, \ldots, c_n$) associated with these discrete values. The cumulative probability (c_j) for x_j is defined as the probability of obtaining a value that is less than or equal to x_j. Hence, c_j is equal to the sum of $p(x_k)$ for k going from 1 to j. By definition,
$c_n = 1$.

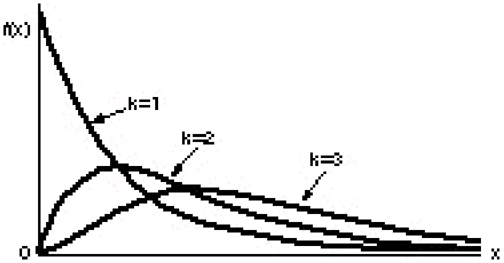
Range $\{x_1, x_2, \ldots, x_n\}$

Applications

The discrete empirical distribution is often used to incorporate discrete empirical data directly into the model. This distribution is frequently used for discrete assignments such as the job type, the visitation sequence, or the batch size for an arriving entity.

Erlang(β, *k*): ERLANG(ExpMean, *k*) or ERLA(ExpMean, *k*)

Probability Density Function



$$f(x) = \begin{cases} \dfrac{\beta^{-k} x^{k-1} e^{-x/\beta}}{(k-1)!} & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Parameters If X1, X2, . . . , Xk are independent, identically distributed exponential random variables, then the sum of these *k* samples has an Erlang-*k* distribution. The mean (β) of each of the component exponential distributions and the number of exponential random variables (*k*) are the parameters of the distribution. The exponential mean is specified as a positive real number, and *k* is specified as a positive integer.

Range $[0, +\infty)$

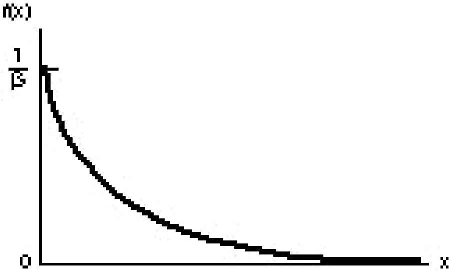
Mean $k\beta$

Variance $k\beta^2$

Applications The Erlang distribution is used in situations in which an activity occurs in successive phases and each phase has an exponential distribution. For large *k*, the Erlang approaches the normal distribution. The Erlang distribution is often used to represent the time required to complete a task. The Erlang distribution is a special case of the gamma distribution in which the shape parameter, α, is an integer (*k*).

Exponential(β): EXPONENTIAL(Mean) or EXPO(Mean)

Probability Density Function



$$f(x) = \begin{cases} \frac{1}{\beta} e^{-x/\beta} & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Parameters The mean (β) specified as a positive real number.

Range $[0, +\infty)$

Mean β

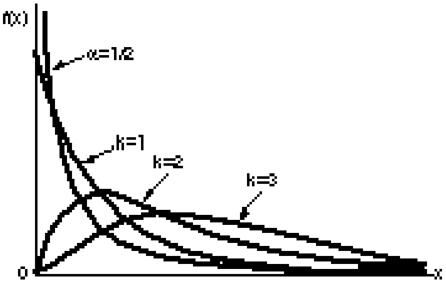
Variance β^2

Applications This distribution is often used to model inter-event times in random arrival and breakdown processes, but it is generally inappropriate for modeling process delay times.

In Arena's Create module, the Schedule option automatically samples from an exponential distribution with a mean that changes according to the defined schedule. This is particularly useful in service applications, such as retail business or call centers, where the volume of customers changes throughout the day.

Gamma(β, α): GAMMA(Beta, Alpha) or GAMM(Beta, Alpha)

Probability Density Function



Unbounded FamilyBounded Family

$$f(x) = \begin{cases} \dfrac{\beta^{-\alpha} x^{\alpha - 1} e^{-x/\beta}}{\Gamma(\alpha)} & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

where Γ is the complete gamma function given by

$$\Gamma(\alpha) = \int_0^{\infty} t^{\alpha - 1} e^{-1} dt$$

Parameters Shape parameter (α) and scale parameter (β) specified as positive real values.

Range $[0, +\infty)$

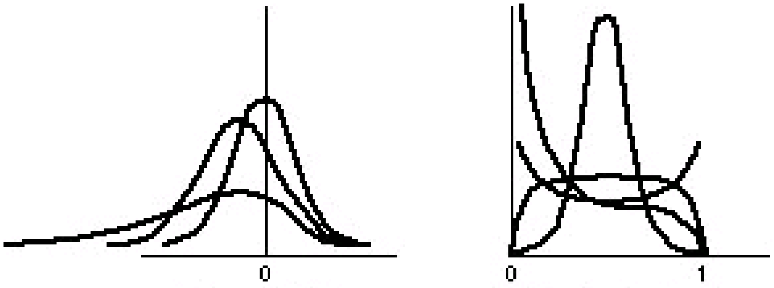
Mean $\alpha\beta$

Variance $\alpha\beta^{2}$

Applications For integer shape parameters, the gamma is the same as the Erlang distribution. The gamma is often used to represent the time required to complete some task (for example, a machining time or machine repair time).

Johnson: JOHNSON(Gamma, Delta, Lambda, Xi) or JOHN(Gamma, Delta, Lambda, Xi)

Probability Density Function



Parameters Gamma shape parameter (γ), Delta shape parameter (δ > 0), Lambda scale parameter (λ), and Xi location parameter (ξ).

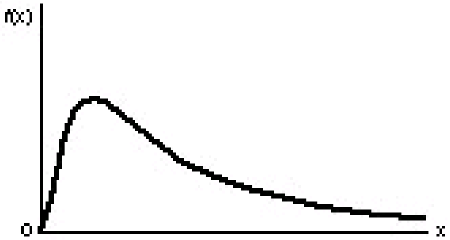
Range

(− ∞, + ∞)	Unbounded Family
[ξ, ξ + λ]	Bounded Family

Applications The flexibility of the Johnson distribution allows it to fit many data sets. Arena can sample from both the unbounded and bounded form of the distribution. If Delta (δ) is passed as a positive number, the bounded form is used. If Delta is passed as a negative value, the unbounded form is used with |δ | as the parameter.

Lognormal(μ, σ): LOGNORMAL(LogMean, LogStd) or LOGN(LogMean, LogStd)

Probability Density Function



Denote the user-specified input parameters as LogMean = μ_l and LogStd = σ_l .

Then let $\mu = \ln(\mu_l^2 / \sqrt{\sigma_l^2 + \mu_l^2})$ and

$$\sigma = \sqrt{\ln\left[(\sigma_l^2 + \mu_l^2)/\mu_l^2\right]}$$

the probability density function can then be written as

$$f(x) = \begin{cases} \dfrac{1}{\sigma x\sqrt{2\pi}} e^{-(\ln(x)-\mu)^2/(2\sigma^2)} & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Parameters Mean LogMean ($\mu_1 > 0$) and standard deviation LogStd ($\sigma_1 > 0$) of the lognormal random variable. Both LogMean and LogStd must be specified as strictly positive real numbers.

Range $[0, +\infty)$

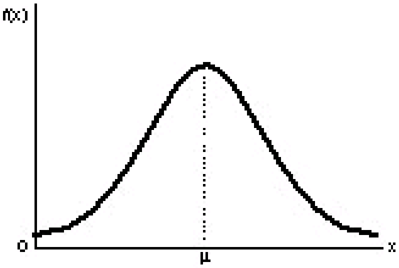
Mean LogMean = $\mu_1 = e^{\mu + \sigma^2/2}$

Variance (LogStd)$^2 = \sigma_1^2 = e^{2\mu + \sigma^2}(e^{\sigma^2} - 1)$

Applications The lognormal distribution is used in situations in which the quantity is the product of a large number of random quantities. It is also frequently used to represent task times that have a distribution skewed to the right. This distribution is related to the normal distribution as follows. If X has a lognormal (μ_l, σ_l) distribution, then ln(X) has a normal (μ, σ) distribution. Note that μ and σ are *not* the mean and standard deviation of the lognormal random variable X, but rather the mean and standard deviation of the normal random variable lnX.

Normal(μ, σ): NORMAL(Mean, StdDev) or NORM(Mean, StdDev)

Probability Density Function



$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)} \text{ for all real } x$$

Parameters The mean (μ) specified as a real number and standard deviation (σ) specified as a positive real number.

Range $(-\infty, +\infty)$

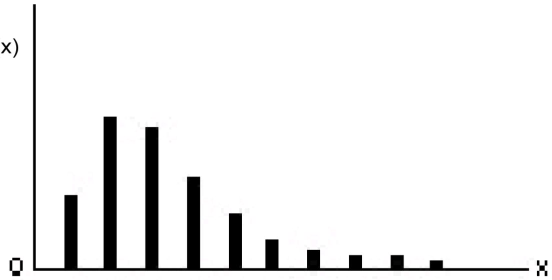
Mean μ

Variance σ^2

Applications The normal distribution is used in situations in which the central limit theorem applies; that is, quantities that are sums of other quantities. It is also used empirically for many processes that appear to have a symmetric distribution. Because the theoretical range is from - ∞ to + ∞, the distribution should only be used for positive quantities like processing times.

Poisson(λ): POISSON(Mean) or POIS(Mean)

Probability Mass Function



$$p(x) = \begin{cases} \dfrac{e^{-\lambda}\lambda^{x}}{x!} & \text{for } x \varepsilon \{0, 1, ...\} \\ 0 & \text{otherwise} \end{cases}$$

Parameters The mean (λ) specified as a positive real number.

Range {0, 1, . . .}

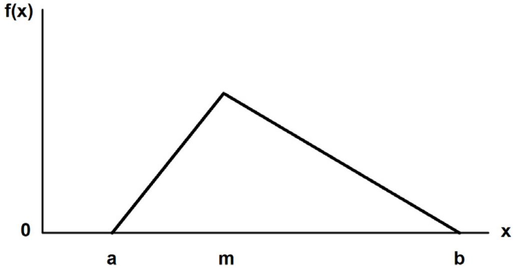
Mean λ

Variance λ

Applications The Poisson distribution is a discrete distribution that is often used to model the number of random events occurring in a fixed interval of time. If the time between successive events is exponentially distributed, then the number of events that occur in a fixed-time interval has a Poisson distribution. The Poisson distribution is also used to model random batch sizes.

Triangular(*a, m, b*): TRIANGULAR(Min, Mode, Max) or TRIA(Min, Mode, Max)

Probability Density Function



$$f(x) = \begin{cases} \dfrac{2(x-a)}{(m-a)(b-a)} & \text{for } a \leq x \leq m \\ \dfrac{2(b-x)}{(b-m)(b-a)} & \text{for } m \leq x \leq b \\ 0 & \text{otherwise} \end{cases}$$

Parameters The minimum (a), mode (m), and maximum (b) values for the distribution specified as real numbers with $a < m < b$.

Range $[a, b]$

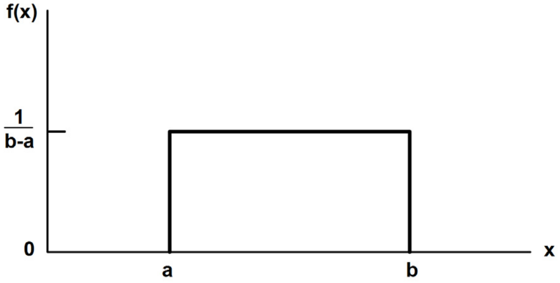
Mean $(a + m + b)/3$

Variance $(a^2 + m^2 + b^2 - ma - ab - mb)/18$

Applications The triangular distribution is commonly used in situations in which the exact form of the distribution is not known, but estimates (or guesses) for the minimum, maximum, and most likely values are available. The triangular distribution is easier to use and explain than other distributions that may be used in this situation (for example, the beta distribution).

Uniform(*a, b*): UNIFORM(Min, Max) or UNIF(Min, Max)

Probability Density Function



$$f(x) = \begin{cases} \dfrac{1}{b-a} & \text{for } a \leq \ x \leq b \\ 0 & \text{otherwise} \end{cases}$$

Parameters The minimum (a) and maximum (b) values for the distribution specified as real numbers with $a < b$.

Range $[a, b]$

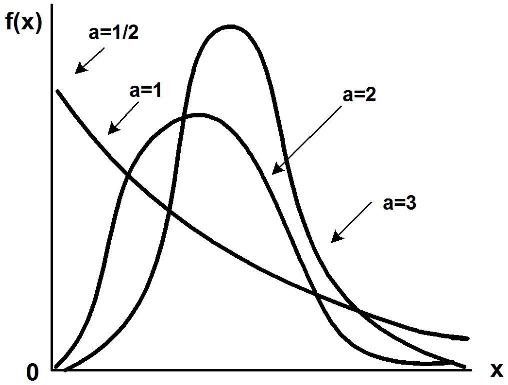
Mean $(a + b)/2$

Variance $(b - a)^2/12$

Applications The uniform distribution is used when all values over a finite range are considered to be equally likely. It is sometimes used when no information other than the range is available. The uniform distribution has a larger variance than other distributions that are used when information is lacking (for example, the triangular distribution).

Weibull(β, α): WEIBULL(Beta, Alpha) or WEIB(Beta, Alpha)

Probability Density Function



$$f(x) = \begin{cases} a\beta^{-a}x^{a-1}e^{-(x/\beta)^{a}} & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Parameters Shape parameter (α) and scale parameter (β) specified as positive real values.

Range $[0, +\infty)$

Mean $\frac{\beta}{a}\Gamma\left(\frac{1}{a}\right)$, where Γ is the complete gamma function (see gamma distribution).

Variance $\frac{\beta^2}{a}\left\{2\Gamma\left(\frac{2}{a}\right) - \frac{1}{a}\left[\Gamma\left(\frac{1}{a}\right)\right]^2\right\}$

Applications

The Weibull distribution is widely used in reliability models to represent the lifetime of a device. If a system consists of a large number of parts that fail independently, and if the system fails when any single part fails, then the time between successive failures can be approximated by the Weibull distribution. This distribution is also used to represent non-negative task times that are skewed to the left.

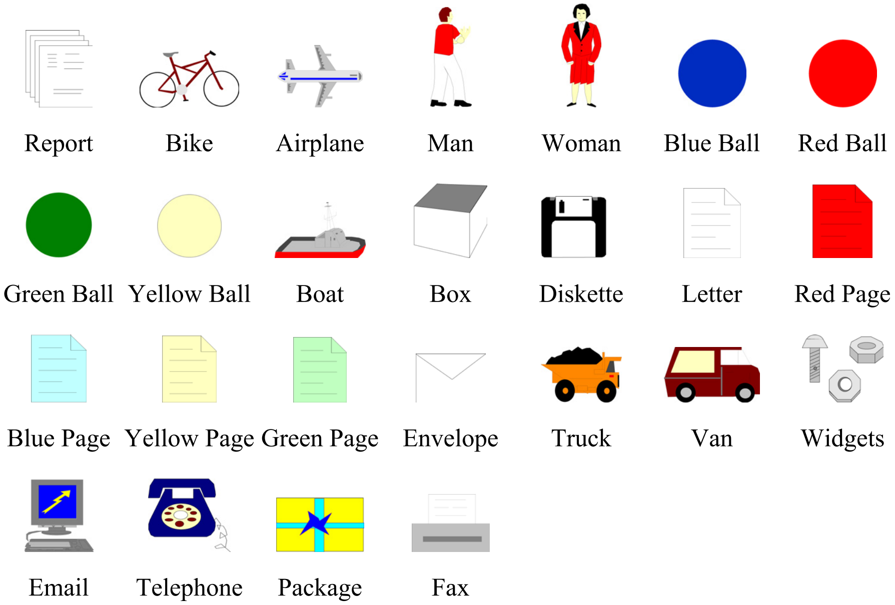
B Library Shapes and Symbols

Arena *.plb* picture files

Arena's picture libraries (**.plb* files) contain images that you may use for entity, resource, and global pictures. You can use the images in these libraries for any of these three types of picture animation. You also can copy these pictures into your own libraries, draw new pictures in Arena's picture editor, or add pictures from the Arena Symbol Factory to any *.plb* file. View the online help topic "Animating Objects" to learn more about adding symbols to your picture libraries.

The Basic Process library supplies a set of default entity pictures that are preloaded into your Arena models. When you start a model, Arena adds the pictures from *BasicProcess.plb* to the entity picture list (see the **Edit > Entity Pictures** menu). Also, the predefined names associated with these pictures are added to the entity picture list (for example, *Picture.Report*) in the Entity module and Entity picture placement dialog box.

Example pictures in Basic Process.plb



Arena .plb libraries

The list of standard Arena picture libraries includes:

- Buldings.plb
- Equipment.plb
- Faces.plb
- Factory.plb
- General.plb
- Machines.plb
- Medical.plb
- Objects.plb
- Office.plb
- Office Equipment.plb
- People.plb
- Symbols.plb
- Vehicles.plb
- Workers.plb

Common shapes may be placed from any of these categories, or they may be added from the Arena Symbol Factory or other image source by pasting from your Clipboard into a specific library. If you wish to create your own library of shapes, simply select **New** from the Entity Picture Placement window. The "Current Library" displays "(None Opened)" because the new library has not yet been named or saved. Simply select **Add** for each library shape you wish to include in your new library file (or an existing one), and draw or paste the appropriate images onto the picture editor board. Remember to drag-and-drop the Ride Point to the appropriate position on your image.

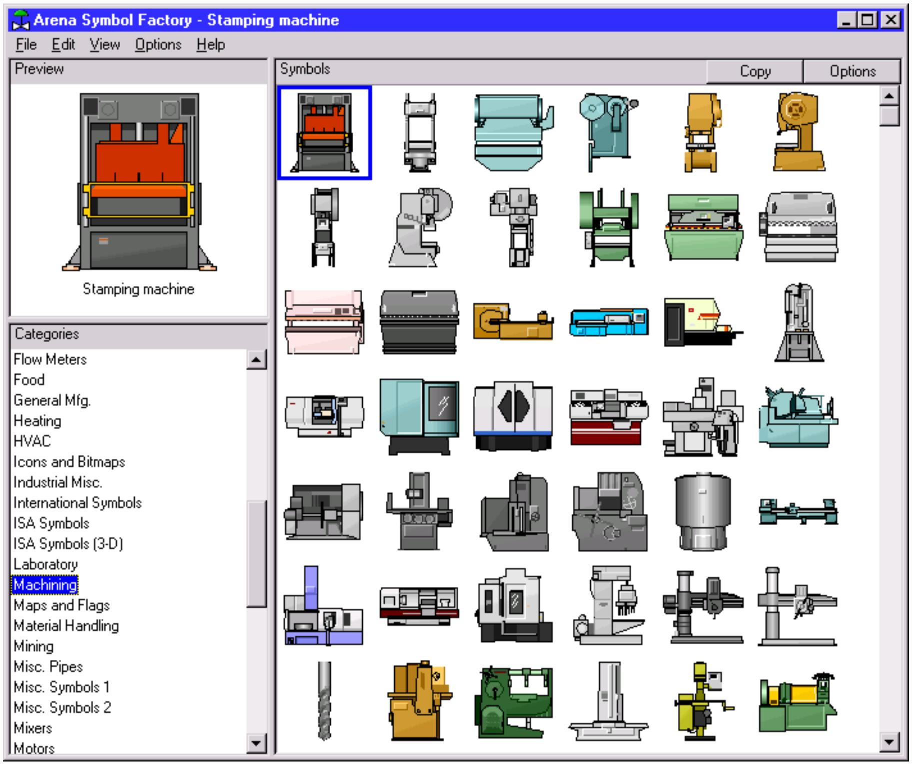
Arena Symbol Factory

Arena animations can also be enhanced using Arena Symbol Factory's extensive library of over 4,000 symbols. These symbols can be used for entity, resource, transporter, or global pictures, or as graphic symbols within a model window. You can copy these images directly to the Arena model window, add them to your own libraries (.plb files), or add them to any of the standard Arena picture library files.

The Symbol Factory application

The Symbol Factory is launched from the main Arena toolbar by selecting **Tools > Arena Symbol Factory**. The application screen contains three distinct areas: the *Preview* pane, showing an enlarged image of the currently selected symbol; the

Categories pane, containing the list of library symbols by groupings; and the *Symbol* pane, displaying images of all shapes contained within a selected category. An example of the application is shown below with the Machining category selected.



The basic operation of Symbol Factory simply requires you to select an object from a category list, click **Edit > Copy**, and then paste the symbol into your picture editor window or the Arena application window (for static shapes). Details on the Arena Symbol Factory can be found in the extensive online help embedded in the application.

You must have suitable license activation to use the full Symbol Factory capabilities. In Arena training/evaluation and run-time versions, Arena Symbol Factory is only available in demo mode. Demo mode limits use to only the first symbol in each category. New categories cannot be created, and symbols cannot be moved, added, deleted, imported, or exported in demo mode.

C Using Visio and Arena

Introduction

Since the mid-1980s, Business Process Improvement (BPI) and simulation have become leading methodologies to allow corporations to deliver high-quality products and services through business process modeling. Businesses are seeking not simply to automate existing operations, but to improve and redesign business processes to meet and exceed customers' expectations for the delivery of products and services.

Many simulation systems are standalone, running only on a desktop and having limited capabilities to integrate with other technologies such as company databases, workflow engines, and other desktop solutions. At the same time, leading corporations have discovered the value of simulation technology. Many companies require simulation to be used as standard procedure in support of process improvements.

A much greater return on investment is possible if process descriptions and simulation results can be accessed throughout the enterprise. Microsoft® Visio®, the leading desktop drawing product, supplies a common graphics platform for representing business process models. Visio's open architecture using ActiveX® and Visual Basic® for Applications allows the overall modeling tool to be customized to fit changing needs. The models also can be shared throughout the enterprise by publishing on an intranet directly from Visio.

Arena software includes a built-in interface to Visio, mapping shapes from a Visio drawing to the appropriate simulation constructs in Arena. Process hierarchy is captured in the Arena simulation model, including activity-based costing and process logic, enabling an understanding of the distribution of cost and effort among process components. As with Visio, Arena supplies robust ActiveX support, a Microsoft® Office-compatible interface, and VBA to fulfill enterprise needs for best-of-breed software products.

The built-in interface to Visio is accessible to all licensed editions of Arena.

The Visio Process Simulation add-in

If you use Visio, you can immediately put Arena's process mapping tool to work with the Visio Process Simulator. Created as an add-in to Visio, the Process Simulator includes a Visio template and stencil for mapping processes with shapes designed to mirror the flowchart and data modules in the Basic Process panel of Arena. When you've completed your Visio process map using these tools, you're ready to simulate it in Arena!

Start a drawing

Before opening the Visio stencil, it is necessary to have an administrator on the machine set the Visio macro security options to Low so that macros will be enabled.

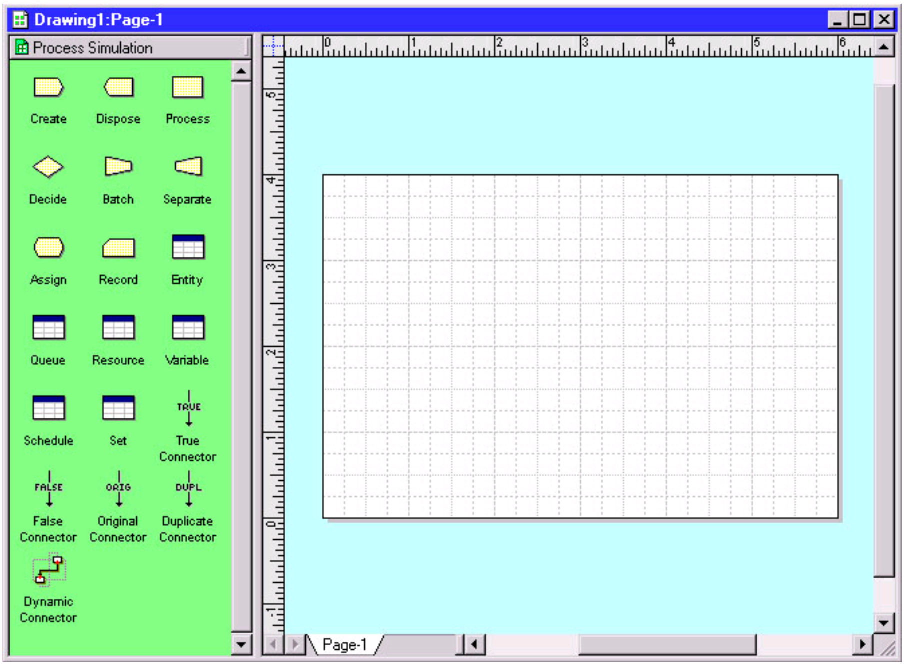
To begin a drawing, browse to the Process Simulation template, which the Process Simulator installs in the Program Files\Rockwell Software\Arena\Visio folder. This dialog box typically appears when you first start Visio, or you can open it using the **File > New > Choose Drawing Type > Browse Templates** menu.

It's important to start your drawing by browsing to the Process Simulation template, rather than opening or starting a new drawing and directly attaching the Process Simulation stencil. If you don't start your drawing with the template, custom dialog boxes and other features may not be available.

The Process Simulation template sets up the Visio environment to work with Arena and opens a new drawing. Drawings that are started with this template will automatically display the Process Simulation stencil, provide custom dialog boxes for editing shape properties, and add a Simulate menu to Visio for checking your drawing and transferring it to Arena.

The Process Simulation stencil, which is opened with your new drawing, contains shapes that will capture your process flow and data. The shapes mirror those in Arena's Basic Process panel, so you can use Visio to define fully a model of your process, including all of the parameters needed to exploit the power of simulation analysis.

The first 14 shapes in the Process Simulation stencil correspond to the Arena flowchart and data modules with the same names. In addition, the Visio stencil contains five connector shapes for properly defining the process flowchart.



Place and connect shapes

You can change the Visio page layout or size using the File > Page Setup menu. For many process maps, a Standard page size and Landscape orientation work well.

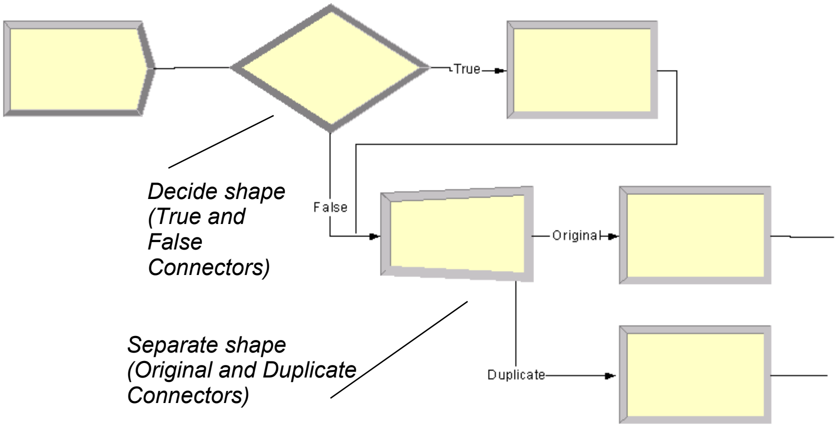
To define the process flow portion of your model, drag and drop flowchart shapes (for example, Create, Process, Dispose) from the Process Simulation stencil into your drawing, just as you would with other standard Visio stencils. Then, use the connectors from the Process Simulation stencil to establish flow.

When connecting shapes, the Dynamic Connector is used for all except Decide and Separate, which use special connectors to determine the type of each of the two possible outgoing connections. The table below lists the shapes from which the connectors can start (for example, the True Connector can only start at a Decide shape).

Connector	Valid Starting Shape
True Connector	Decide
False Connector	Decide

Connector	Valid Starting Shape
Original Connector	Separate
Duplicate Connector	Separate
Dynamic Connector	Create, Process, Batch, Assign, Record

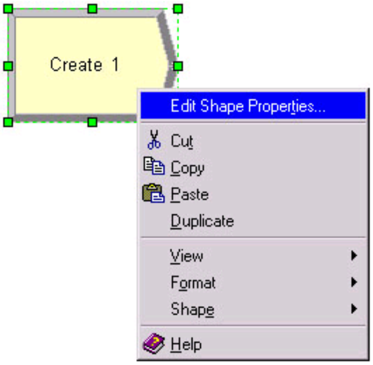
The sample flowchart below illustrates the use of these connectors, including the Dynamic Connector, which is unlabeled.


True
False
Original
Duplicate
Decide shape (True and False Connectors)
Separate shape (Original and Duplicate Connectors)

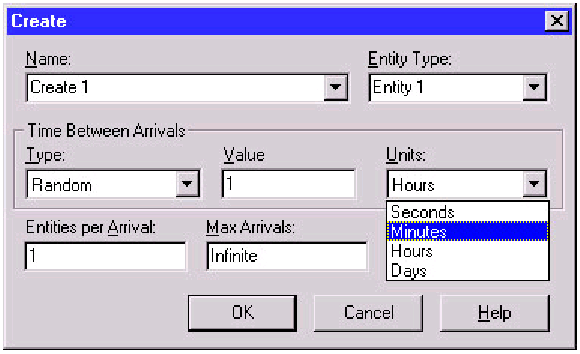
Enter custom properties

When you place shapes from the Process Simulation stencil, default names appear (for example, Create 1, Process 1) to identify the steps in the process flow. Typically, you'll replace these names with descriptions of the process steps. To do so, simply select the shape and type the new name, as with other standard Visio shapes.

The Process Simulation stencil's shapes also provide custom dialog boxes for defining property values, including lists of valid options and other features to aid you in quickly characterizing process data. To open the dialog box, either double-click the shape or right-click and select **Edit Shape Properties** from the pop-up menu, as shown next.



Either of these actions will display the appropriate dialog box for the shape. The dialog boxes that Arena presents for its corresponding modules are designed to match those in Visio, so that you’re working in a familiar interface, whether mapping your process in Visio or Arena.


Create

Name: Create 1

Entity Type: Entity 1

Time Between Arrivals

Type: Random

Value: 1

Units: Hours

Entities per Arrival: 1

Max Arrivals: Infinite

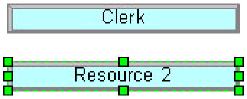
The property values that you define in the custom dialog boxes are stored with the individual shapes. When you transfer the process map to Arena, these parameters are used for simulation.

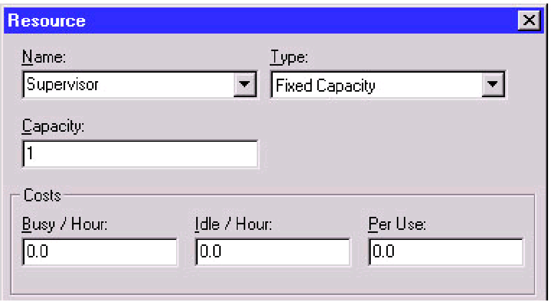
Add data shapes

To define the data modules (for example, Entity, Resource) for the model of your process, you place shapes directly in the drawing for each of the elements you want to define. When the individual data shapes are transferred to Arena, they will not appear in the model window, but will be added to the appropriate spreadsheets instead.

For example, if processes in your flowchart are performed by resources named Clerk and Supervisor, you would place two Resource shapes and specify their characteristics (cost, capacity) by opening the property dialog boxes.

You only need to add the data shapes if you want to change any of the default values. Otherwise, they are not needed in the Visio drawing; they'll automatically be added to the spreadsheets when you transfer the drawing to Arena.





Name: Supervisor

Type: Fixed Capacity

Capacity: 1

Busy / Hour: 0.0

Idle / Hour: 0.0

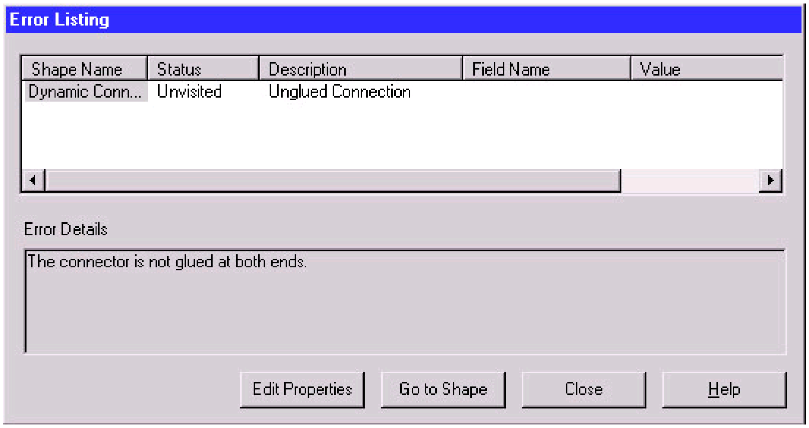
Per Use: 0.0

Simulate the process map in Arena

By using the Process Simulation stencil in Visio, you can define, document, and communicate your business processes. When you want to predict how your process will operate, examine it for bottlenecks, and compare possible changes to make informed decisions, you can transfer your process map from Visio to Arena, where the process will come to life with simulation and animation.

Check the drawing

While you're working with your process map, you can check it for any missing or incorrect data or connections. To check your drawing, use the **Check Drawing** item from the **Simulate** menu. After the drawing has been examined, a list of any warnings and error messages is displayed, as shown below for a drawing that has a connector that is not glued at both ends.

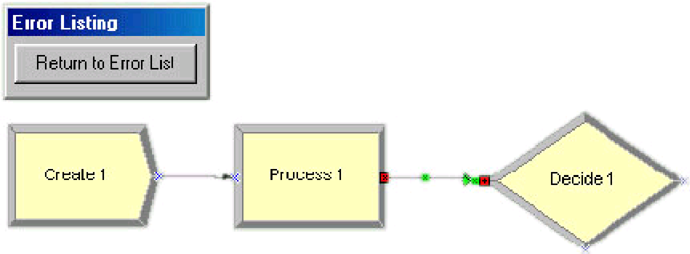


Shape Name	Status	Description	Field Name	Value
Dynamic Conn...	Unvisited	Unglued Connection		

Error Details

The connector is not glued at both ends.

If multiple problems are detected, each will be listed with a description and, in the case of invalid or missing properties, the field and value that are improper are listed. To correct the errors in your drawing, you can use the buttons at the bottom of the dialog box. For errors in shape properties, click the **Edit Properties** button, which displays the shape's dialog box, to correct the data. In the case of connector problems, click **Go to Shape**, which displays and selects the shape that caused the problem, as shown below. (Note that the Error Listing dialog box shrinks to allow more room for viewing the drawing.)



After you've corrected the problem, you can return to the list of warnings and error messages by clicking the **Return to Error List** button. Once you finish correcting any additional problems, click the **Close** button. You can check your drawing at any time, correcting any or all errors that are detected.

Transfer the drawing to Arena

When you're ready to simulate your process, it's a simple matter of selecting the **Simulate Drawing** item from the **Simulate** menu. This tool opens Arena and translates your Visio process map into an Arena simulation model. When the process is complete, you have an Arena model containing modules from the Basic Process panel that mirror the shapes you used from the Visio Process Simulation stencil, including the properties and connections you specified in Visio.

In Arena, you can set the simulation run length on the **Run > Setup** page. If you do not define the run length, you can stop the simulation at any time by clicking the **Pause** button and then the **End** button.

To run the simulation, click Arena's **Go** button. Your entities will animate the flowchart as the simulation proceeds, just as if you had created the model directly in Arena. Because the process is being simulated in Arena with the standard Basic Process modules, you have all of Arena's capabilities at your disposal, including reports on all aspects of your process from the Reports panel selections.

After you have simulated your process, you can save the Arena model (*.doe*) file to continue work in Arena, or you can close it to return to Visio for continued modifications. Because the transfer of the Visio drawing is from Visio to Arena (but not vice versa from Arena to Visio), it's usually best to return to Visio. However, you might want to save the Arena model in case you want to review the simulation or results as you're refining the Visio process map.

Simulate other Visio drawings

In addition to simulating process maps using the Process Simulation add-in, you can also simulate process maps created using other Visio stencils. Arena provides a shape map editor to define the relationship between the shapes in the Visio stencils and the modules in Arena's Basic Process panel. Arena's online help describes how to use this feature, which is accessed by opening the **Tools > Options** menu and clicking on the **Visio Options** tab.

Use Visio and Arena in your organization

The combination of Visio and Arena can widen and deepen the reach of process improvement in your organization. You can exploit the accessibility of Visio to ease the transition from drawing pictures to mapping processes. All types of process constituents—from those who govern the business to the staff who execute the process—can directly influence the definition of how the process works now and what future perations will provide the most dramatic improvements.

Distribute the Process Simulation add-in

The Process Simulation add-in is automatically installed with Arena and is available to all Visio users, with no additional fees or licenses required. You can distribute the add-in throughout your organization so that everyone who has an interest in the process definition can access it directly through Visio.

With this approach, process maps can be created and shared throughout the organization on any desktop with a Visio license. When simulation and animation are needed to visualize and analyze the behavior of the process, the drawings can be transferred to systems with the Arena software. By using the Process Simulation template and checking the process maps as they're created, you can be confident that taking the step to move from static drawing to dynamic simulation will be seamless.

Publish process maps on a Web page

You can share your process maps—the drawing of your process flowchart—with others by using Visio's HTML export feature to publish on an intranet. To save the drawing in HTML format, select the **File > Save As** menu option in Visio and change the type to `HTML Files`. (See Visio's online help for more information about the HTML export feature.)

The value of modeling and analyzing your processes can be further exploited by saving process performance information with the process maps. The results of simulation runs can be saved in HTML format, providing a benchmark against which actual operations or other improvement concepts can be compared. To save the reports, click the **Export** button when you have an Arena report open and select the **HTML** option as the file type.

Index