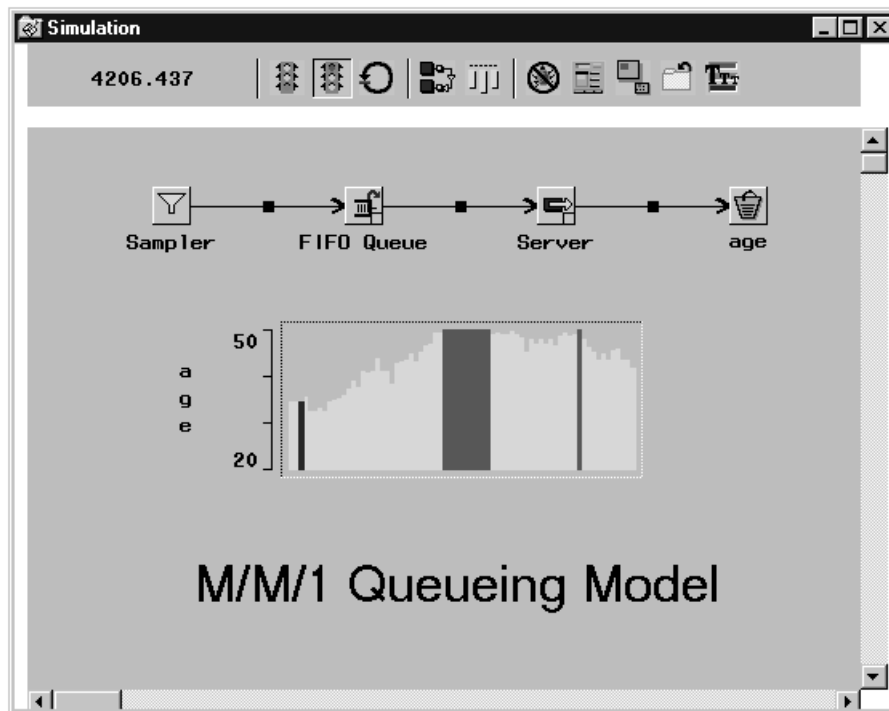


# Chapter 1

## Getting Started

The QSIM Application is a SAS application for modeling and analyzing queueing systems using discrete event simulation. These models are used in a wide variety of scenarios that might be encountered in network and telecommunications systems, manufacturing systems, and transportation networks. For example, before instituting a re-engineered business process, you could use the QSIM Application to model the new process and study the model behavior to gain insight into how the new process might behave.



**Figure 1.1.** An M/M/1 Queueing Model

The application has a graphical user interface that requires no programming and provides all the tools for building, executing, and analyzing discrete event simulation models.

Figure 1.1 shows a simple M/M/1 queueing model built with the QSIM Application. An M/M/1 queueing model has a Poisson arrival process, exponential service times for a single server, and a FIFO queueing discipline. You can build a model of such a queueing system, control the simulation of the model, and produce summary statistics from the simulation sample path from within the application. You can save the model and the sample path in SAS data sets for reuse and further analysis. The application is designed to simplify model building by encouraging the construction of

hierarchical models based on user-built model components that can be stored, shared, and replicated easily.

In this context a model is a directed network. Transactions flow through the network changing the state of the model upon arrival at vertices or nodes. The type of state change that occurs depends on the current state of the simulation and the particulars of the model.

---

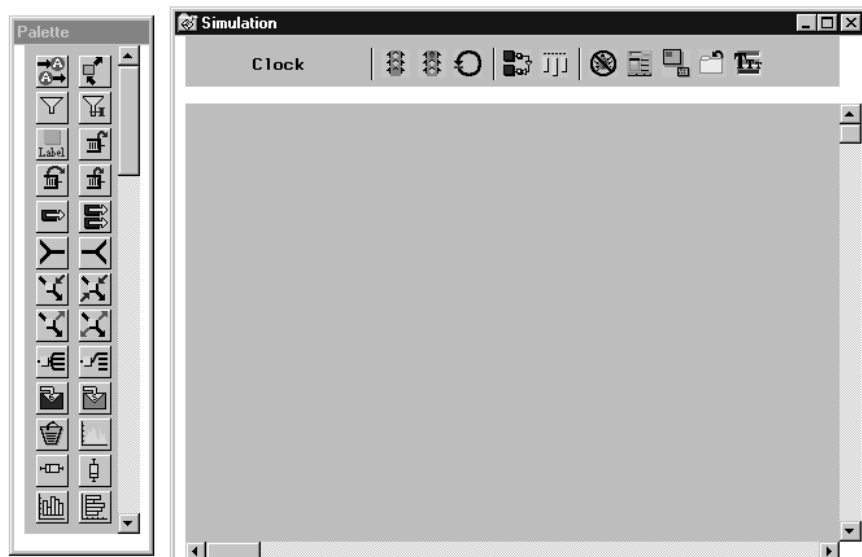
## Invoking the QSIM Application

You invoke the application by typing QSIM on a command line or selecting **Solutions** ► from the menu bar then **Analysis** ► **Queueing Simulation** from the pull-down menu.


---


## User Interface

When you invoke the QSIM Application, a Simulation window and a palette opens, as shown in [Figure 1.2](#). The Simulation window has two panels; the command panel on the top and the model panel on the bottom. The palette contains the components that can be used to build a model. You build the model in the model panel, using the mouse to drag and drop model components from the palette into the model panel. These model components are connected by arcs to produce a directed network representation of the model. You can change parameters and other properties of model components using pop-up menus from the components.



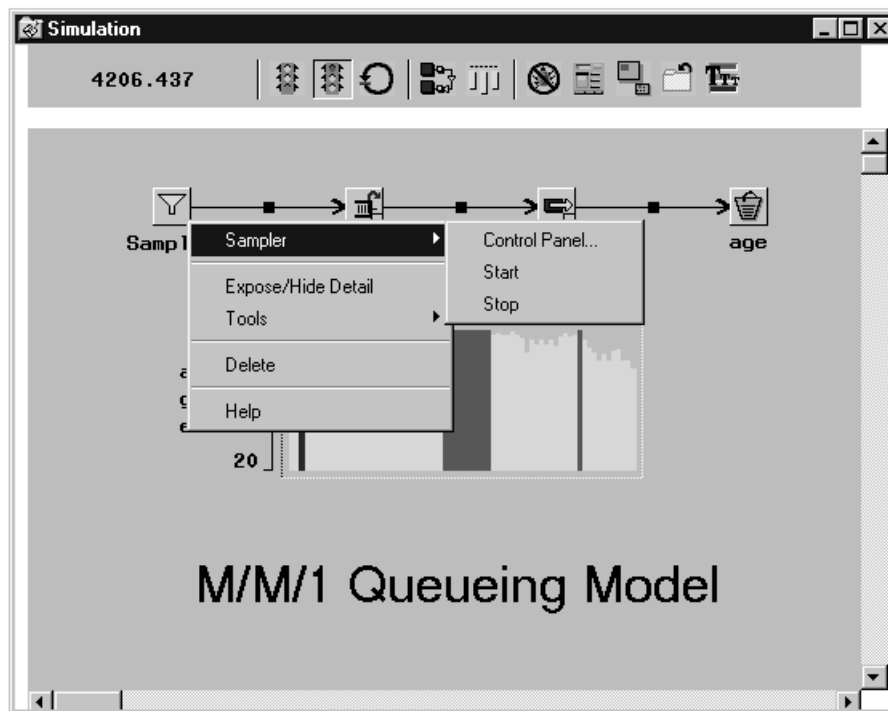
**Figure 1.2.** The Palette and the Simulation Window

If you don't see the component that you want in the palette, use the scrollbar on its right side to scroll the palette. You pick up the component from the palette and drag it to the model panel and then drop it. For example, to place a [Sampler](#) in the Simulation window, you pick up the  component in the palette and then drop it on the model panel.

The command panel contains command buttons. These are push buttons that execute commands. For example, the animate button, , is a command button that toggles animation on and off when pressed.

The QSIM Application also supports cursor tracking. As you move the cursor over a component, the shape of the cursor will change depending on where the cursor is in the component. When the cursor changes shape, you can click the left mouse button and elicit location-dependent behavior such as resizing the component, drawing an arc, or replicating (cloning) a component. For example, if you click when the cursor is a +, you can draw an arc from that component to the next component that you click. Also, picking and dragging a component with the cursor will move the component. Finally, clicking the right mouse button will always display a pop-up menu.


This paradigm for the user interface is used throughout the QSIM Application.



**Figure 1.3.** The Sampler Pop-Up Menu

Figure 1.3 on page 9 shows the pop-up menu on the Sampler obtained by clicking the right mouse button. Notice that the first item in the menu names the type of component. The submenu from the Sampler entry lists the actions that are specific to that component. In this case, it includes: **Control Panel...**, which opens a window that enables you to control the component, an action often used; **Start**, which starts the transaction sampling, an action seldom needed; and **Stop**, which stops the transaction sampling, also an action seldom needed. Other general actions include **Expose/Hide Detail** for changing the Sampler representation from an image to a line drawing, and **Tools** for other less frequently used tools.

Model components are connected by arcs. Transactions flow down arcs between components. You can connect two components with an arc by placing the cursor over the right side of a component. Notice that the cursor shape changes to +. If you click the left mouse button you will have a rubberband line to the cursor. Now when you select another component, an arc will connect the two components. If you move the cursor out of the Simulation window while the rubberband line is connected to it, the window may start power scrolling. Bringing the cursor back into the window will stop the scrolling. Clicking outside the window will take you out of the rubberband line mode.












Model components have control panels associated with them. With these panels, you specify parameters and control the behavior of the component. The simulation also has a control panel, as shown in Figure 1.5 on page 12. It is displayed by selecting the  command button.

## The Command Buttons



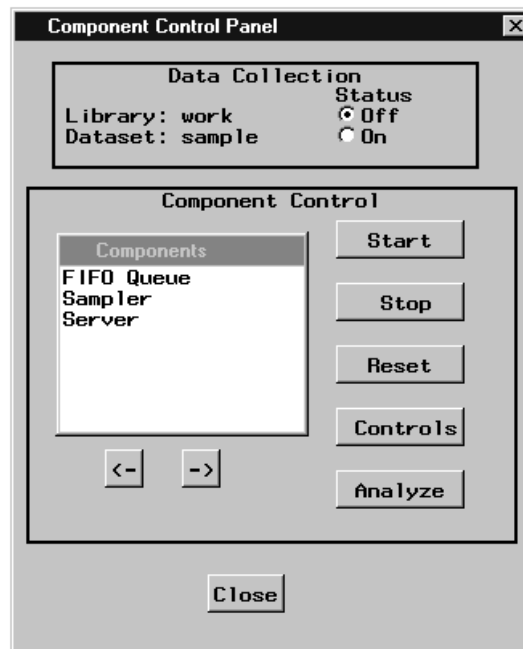
**Figure 1.4.** The Command Panel and Buttons

The command buttons shown in [Figure 1.4](#) initiate global simulation commands. Notice that the stop button is pressed. This means that the simulation is temporarily halted. If the button were pressed again, the simulation would proceed from the point at which it was interrupted. From left to right the command buttons are

Button	Description
	shows the simulation time. By default time is measured in dimensionless units, but it can be changed to any unit from milliseconds to years using the pull down menu <b>Options &gt; Clock...</b> Pressing on this button toggles between the simulation time and the clock icon.
	starts the simulation.
	stops the simulation.
	resets the simulation.
	toggles transaction animation on and off. When animation is on, you can see transactions that originate at <b>Samplers</b> traversing the network.
	aligns components vertically and horizontally.
	raises the Debug Control Panel.
	raises the Components Control Panel.
	raises the Viewport Window.
	saves the simulation model.
	raises the Component Attributes Panel for changing colors, fonts, and images.

## The Component Control Panel



The Component Control Panel gives you control over individual components in your models. You select components in the Components list box and then press buttons in the window to perform actions. The Components list box shows the highest level of components in the model. Since components can be nested into a hierarchy, there can be multiple levels.



**Figure 1.5.** The Component Control Panel

If you select a component and press one of the five buttons to the right of the Components list box, the component executes that action. For example, selecting **Sampler** in the list box and pressing the **Start** button starts transaction sampling in that sampler.

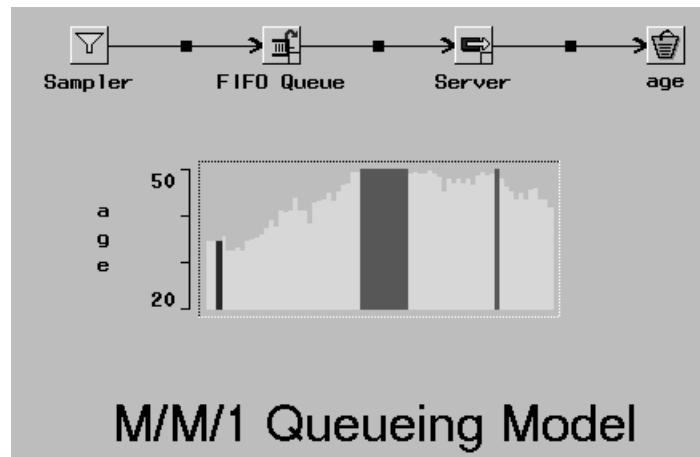
- The **Start** button starts all the selected components.
- The **Stop** button stops all the selected components.
- The **Reset** button resets all selected components.
- The **Controls** button raises the control panels for all selected components.
- The **Analyze** button executes statistical analyses on all selected components.

The two buttons  and  are used for navigating into and out of compound components in the Components list box.

Before performing any statistical analysis, data must be collected. The Data Collection check box controls saving the simulation sample path into a SAS data set which is then used for the data analysis.

## A Simple M/M/1 Queueing Model

A simple example illustrates some of the concepts involved in model building. The network shown in [Figure 1.6](#) models an M/M/1 queue. Transactions originate at the source node **Sampler**. The user chooses the interval between transactions, called the *inter-arrival time*, to be a sample of a random variable (from one of several distributions), a fixed amount, or the value of a variable read from a SAS data set. By default, the inter-arrival time is an observation of an exponential random variable with parameter 1. This models a Poisson process for transaction arrivals.



**Figure 1.6.** An M/M/1 Queueing Model

### Queueing

When the transaction leaves the **Sampler**, it flows down the arc to the **FIFO Queue**. It is important to note that the movement of the transaction down the arc does not advance the simulation clock. On transaction arrival at the **FIFO Queue**, the queue broadcasts messages down arcs asking nodes downstream if they are busy. The responses depend on the types of components that are connected to the queue and the state of the simulation when the message is received. (Details are given in the next chapter.) It is important to note that broadcasting and evaluation of these messages does not advance the simulation clock. If the queue gets a response that there is a non-busy node, then it sends the transaction down the arc leading to that node. Otherwise, the transaction remains in the queue. When the simulation is first started, the **Server** is empty; when it gets the message “are you busy” from the queue, it responds “no.” As a result, the queue routes the transaction down the arc to the **Server**.

### Service

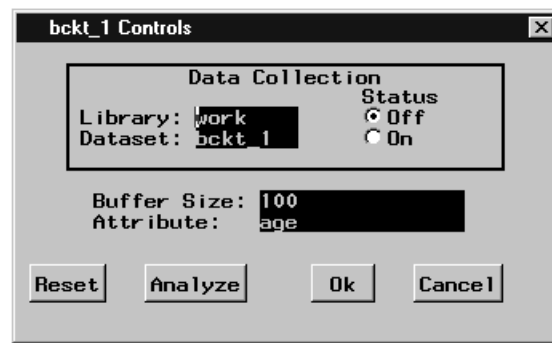
When the transaction arrives at the **Server**, service is scheduled and the transaction ties up the server. By default, the service time is an observation of an exponential random variable with parameter 1. Both the service distribution and its parameters can be changed using the server’s control panel. While the server is serving this transaction, any “are you busy” messages sent to it result in a “yes” response. When service is complete, the server sends the transaction on any arcs directed away from it and also

sends a message up the arcs directed into it requesting an additional transaction. In this example, if the **FIFO Queue** is not empty, it will remove the transaction that has been there the longest and send it to the Server. By default, all queues in the system have a capacity of 50 transactions. Of course, this capacity can be changed through the user interface or programmatically, as discussed in the next chapter. Since by default the inter-arrival times and the service times are  $\mathcal{E}(1)$ , exponentially distributed with mean 1, the transaction time in the system would not have a stationary distribution if the queue had infinite capacity.

---

## Statistics

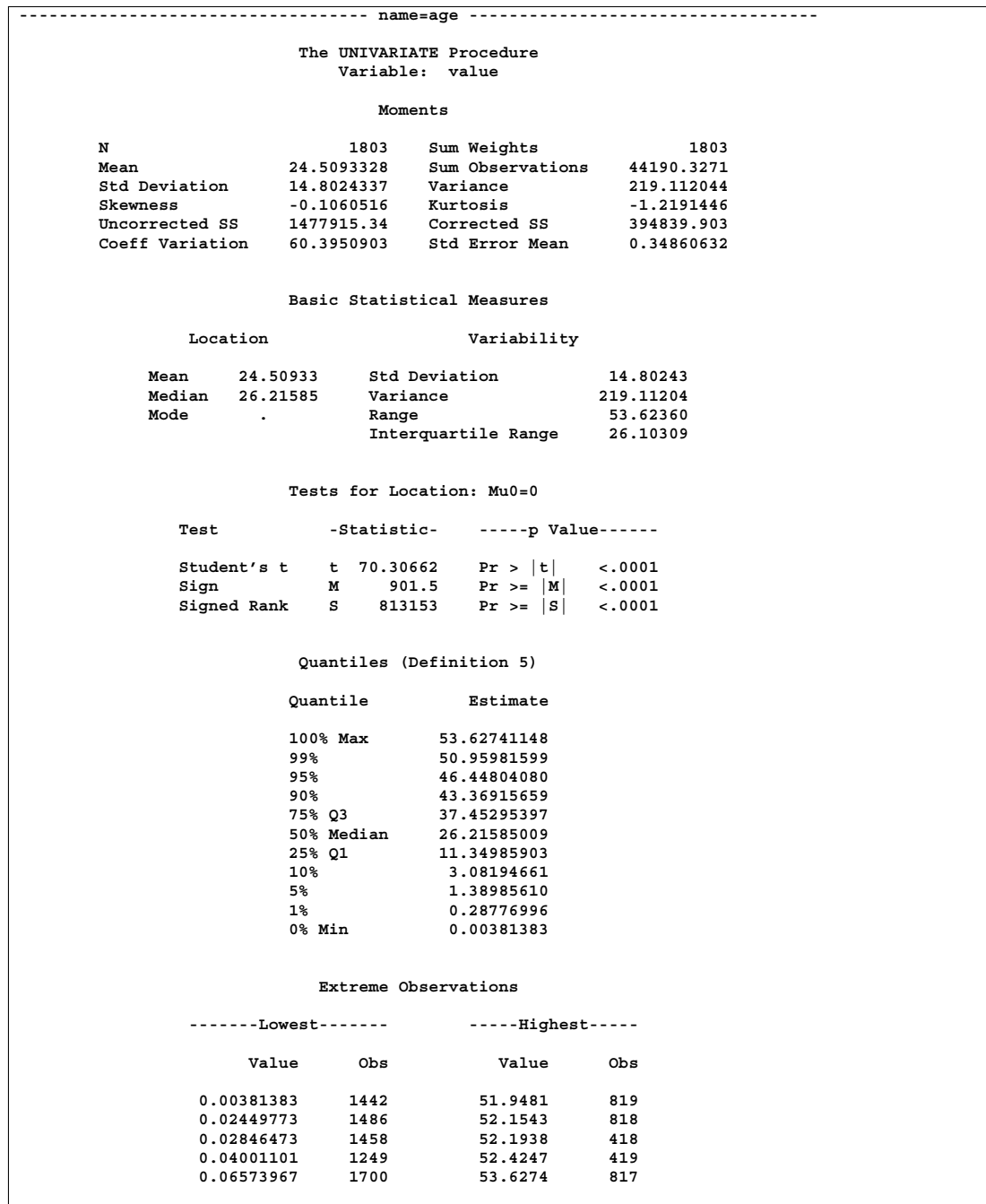
Finally, the **transaction** flows to a **Bucket**, which collects transactions and can also save the value of transaction attributes in a SAS data set.



**Figure 1.7.** A Bucket Control Panel

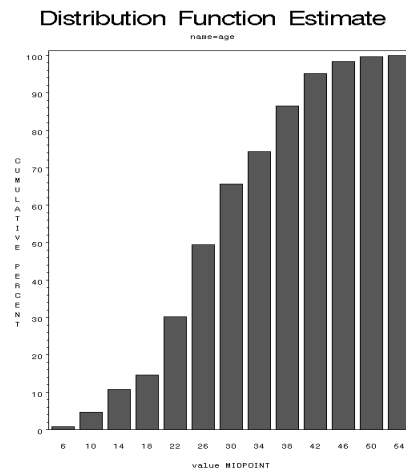
From the bucket control panel (**Figure 1.7**) you can set the size of the transaction collection buffer and name the transaction attribute to accumulate. You can also name a SAS data set into which to collect the transaction attribute values.





**Figure 1.8.** Statistics on a Transaction Attribute

If you collect data into a SAS data set and then press the **Analyze** button, univariate statistics will be calculated by the UNIVARIATE procedure (Figure 1.8) and a sample distribution function as shown in Figure 1.9 on page 16 will be plotted.



**Figure 1.9.** Sample Distribution Function

See [Chapter 8, “Analyzing the Sample Path,”](#) for more information on on collecting statistics and analyzing simulation data.