

Chapter 5

Building a Model with Compound Components

The ability to assemble elementary components into larger aggregates is an important feature of the QSIM Application. It encourages hierarchical model building, information hiding, and component reuse. This chapter discusses the details of model building using compound components.

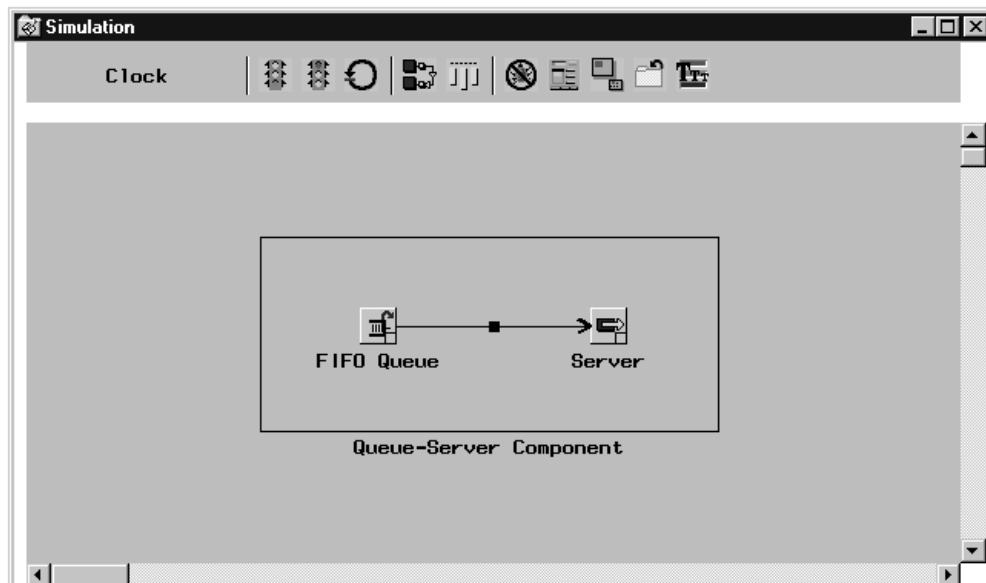


Figure 5.1. A Compound Component with Queue and a Server

Assembling Components into Compound Components

The pop-up menu on the Simulation window (see Figure 5.2) has as its first entry **Assemble Components**. When you choose this selection, you get a rubberband rectangle with which you can sweep out an area on the Simulation Window. Any components that are completely within this region will be encapsulated in a compound component.

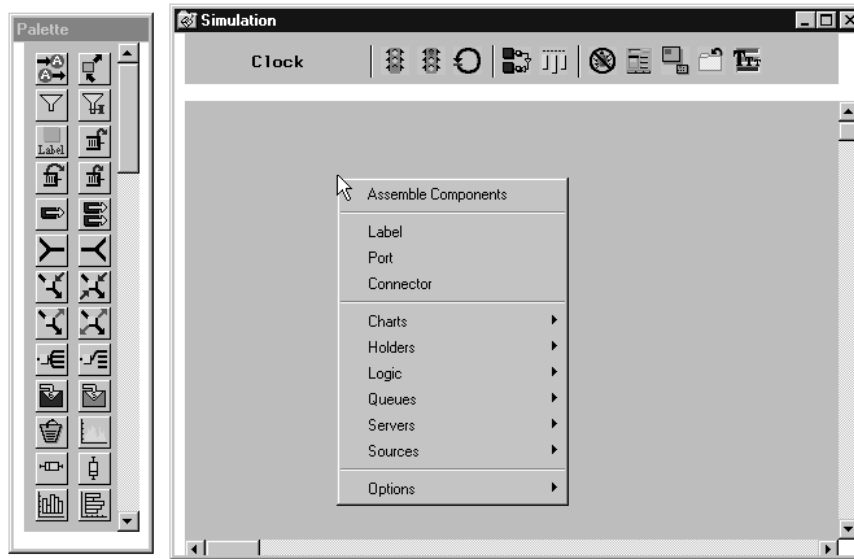


Figure 5.2. The Simulation Window Pop-up Menu

Figure 5.1 shows an example compound component. This encapsulated queue and server can now be treated as a single unit, called a compound component. It has a pop-up menu, shown in Figure 5.3, which includes **Assemble Components** and **Disassemble Components** for assembling additional compound components within it and for removing the encapsulation.

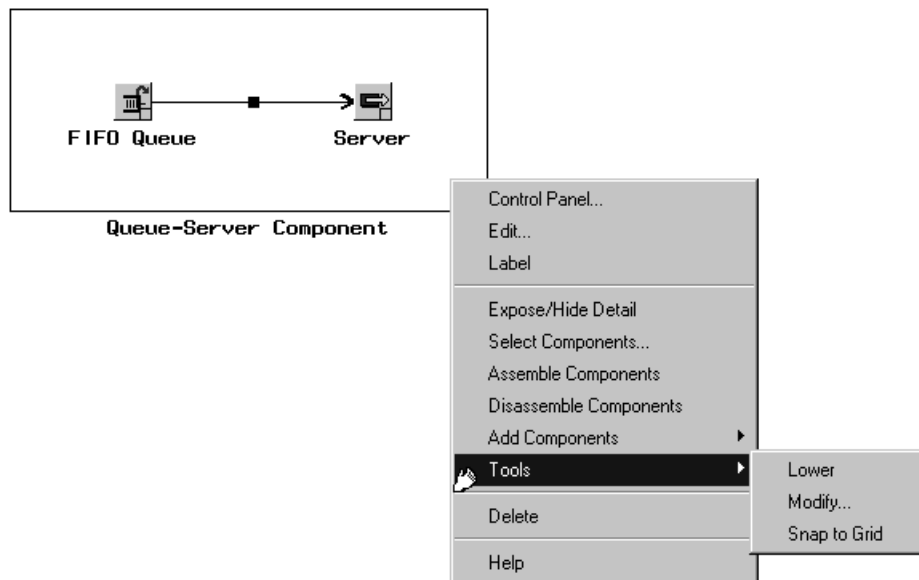



Figure 5.3. The Compound Component Pop-up Menu

This compound component also includes **Edit**, a selection that opens a separate window for editing the contents of the compound component, and **Expose/Hide Details**, a selection that toggles the compound component with an icon. The default icon, , can be replaced with a user icon by using the Component Attributes Panel selection on the Command Panel (see Figure 8.2).

You can also pick up components and move them into and out of compound components. Compound components themselves can be moved into and out of compound components as well. And, it is not necessary to use drag and drop to accomplish this. You simply move the component by pressing the mouse button down as you scroll over the component when you see the hand icon. When you release the mouse button, the component will be in the new position within the compound component.

Nested Compound Components

It is often useful to nest compound components within one another when making reusable components. One issue that arises when doing this is how to expose externally connections to internal component logic. The Port component is particularly helpful to accomplish this when nested compound components are built for subsequent reuse. Consider an example of a queue-server combination. By adding two Ports, one for transactions to enter the queue and the one for transactions to leave after service, and then assembling the ports and the compound component into a larger unit, you can further encapsulate the queue-server combination as shown in Figure 5.4.

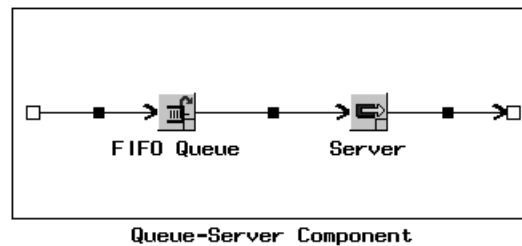


Figure 5.4. A Multi-Level Compound Component with Queue, Server, and Ports

Controlling Subcomponent Exposure

Another feature that helps in building hierarchical models is the ability to limit the exposure of compound components. Even though the model in Figure 5.4 is nested and allows you to connect to ports, it still exposes submodel detail. However, the pop-up menu on the compound component also has an entry labeled **Expose/Hide Details**. Selection of this entry hides the detail in the compound component behind

the icon  .

With the inner compound component hidden, the model shown in Figure 5.4 looks like Figure 5.5.

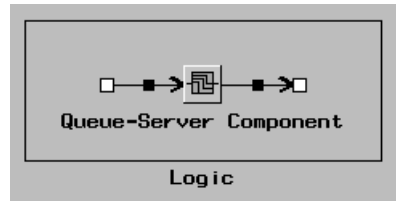


Figure 5.5. A Multi-Level Compound Component with Ports

You can expose the detail in the inner Queue/Server component by selecting **Expose/Hide Details** again on the component labeled Logic. With these features, you assemble components with complex behavior, attach ports to the substructures, then hide the detail, exposing only the structure necessary for using the compound component.

Not only can you hide the visual details of compound components, but you can also hide the details of the logic in a compound component. Just hide the details of the Logic component in Figure 5.5 and you obtain the example shown in Figure 5.6.



Figure 5.6. A Multi-Level Compound Component with Details Hidden

Notice that the Ports remain exposed. That is because the **Toggle Exposure** was selected from the pop-up menu on the Port as shown in Figure 5.7.

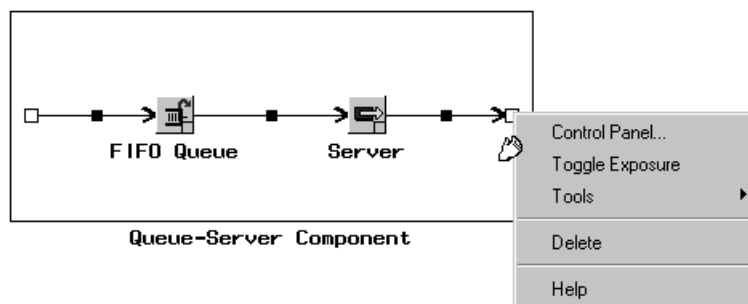


Figure 5.7. A Multi-Level Compound Component with Queue, Server, and Ports

The pop-up menu on compound components in Figure 5.3 shows the selection **Select Components...**. When this is selected for the component labeled Logic, the window in Figure 5.8 is displayed. In this window, you can select those components that you want exposed when queries are made on the Logic compound component.

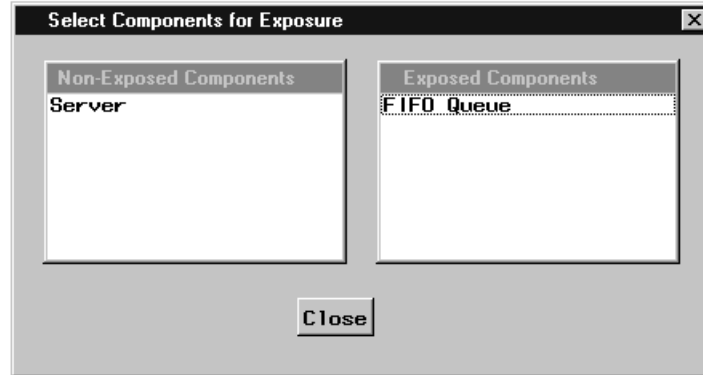


Figure 5.8. A Compound Component Control Panel

Initially the Exposed Components list box contains all the components in the Logic compound component. These are the only components that will be exposed. You move components between the two Exposed Components and Non-Exposed Components list boxes by simply selecting them. Notice that only FIFO Queue will be exposed. For example, consider the model shown in Figure 5.9.

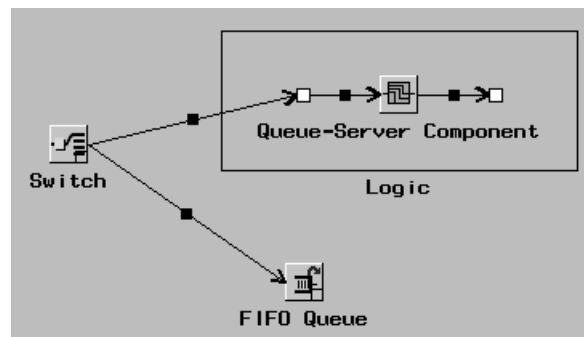


Figure 5.9. A Model with Switch and Compound Component

If you select the **Controls...** from the pop-up menu on the Queue-Server Component then only the FIFO Queue Control Panel will be raised, as shown in Figure 5.10.

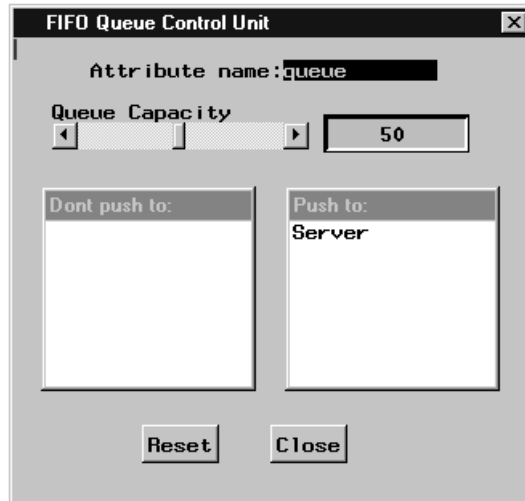


Figure 5.10. Control Panel for the Formula Model Element

This is because you had previously (in Figure 5.8) selected only that element to be exposed.

Editing Components

Suppose that you want to edit the Logic component in the model in Figure 5.9. One way is to visually expose the detail in an edit window by selecting **Edit...** from the pop-up menu on compound components (see Figure 5.3). Figure 5.11 shows such an edit window for the Logic component.

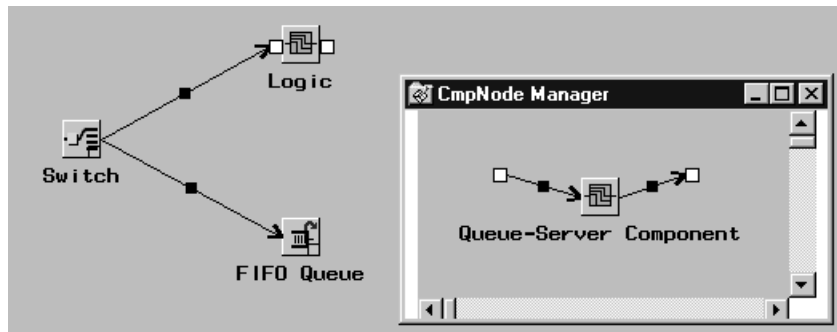


Figure 5.11. Editing the Logic Compound Component

In this new window, you can modify the detail structure of the Logic compound component.

Palettes for Reusing Components

Another powerful feature of compound components is the ability to use them as templates for replication. You do this by simply dragging and dropping them into a palette. Notice the compound component at the bottom of the palette in Figure 5.12. Since this is in the palette and is a template, it can be replicated.

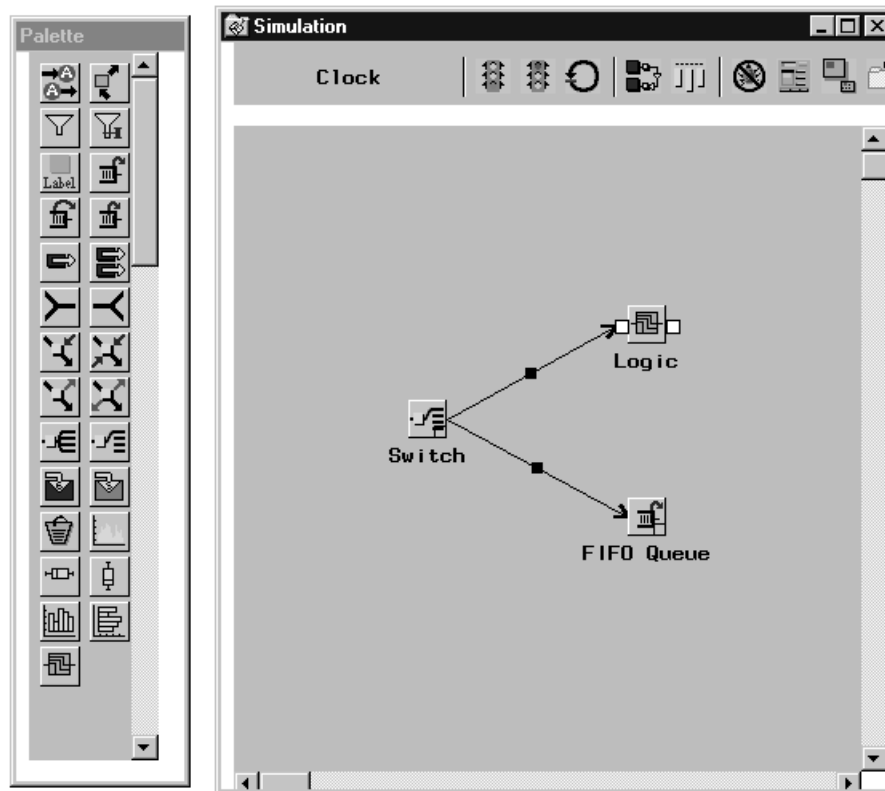


Figure 5.12. The Toolbar with the New “Logic” Button

Now, this icon can be replicated by simply dragging it onto the Simulation window as you would any other component in the palette. You can change the image on the icon by using the pop-up menu on the icon and selecting **Graphic Attributes**. Furthermore, this icon or template can be saved in a SAS dataset by saving the palette. See Chapter 7, “Saving and Restoring,” for details.

Note that it is possible to create compound components and as a result templates that reference components not in the compound component and template itself. For example, you can have a Trigger as a template that references a queue not in that template. When the template is replicated, the Trigger will no longer reference the queue, and its reference to a component in the model will have to be re-established. However, any references to components within the template will be preserved in replicates.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/OR® User's Guide: QSIM Application, Version 8*, Cary, NC: SAS Institute Inc., 1999. 110 pp.

SAS/OR® User's Guide: QSIM Application, Version 8

Copyright © 1999 SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-488-8

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, by any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute, Inc.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of the software by the government is subject to restrictions as set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.® indicates USA registration.

IBM®, ACF/VTAM®, AIX®, APPN®, MVS/ESA®, OS/2®, OS/390®, VM/ESA®, and VTAM® are registered trademarks or trademarks of International Business Machines Corporation.® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

The Institute is a private company devoted to the support and further development of its software and related services.