

Transaction Whiteboard: Creating an Application Model

Transaction Whiteboard allows you to create and modify application definitions graphically, ranging from small changes in existing applications captured with AppTransaction Xpert to creating new application models from scratch.

Sample use cases for Transaction Whiteboard include the following:

- 1) You created a Transaction Analyzer model of an application that currently exists in your network. This application always downloads an image of a fixed size. You are interested in studying how the application will perform if a different image size is used. You can use Transaction Whiteboard to:
 - a) Open the existing Transaction Analyzer model of the transaction
 - b) Change the image size
 - c) Use QuickPredict or Discrete Event Simulation to predict performance

An alternative approach is to parameterize the image size. This technique is shown in the tutorial Transaction Whiteboard: Modifying an Application Model.

- 2) You created a Transaction Analyzer model of a database application that currently exists on your network. This application has a variable number of SELECT statements, and the amount of data returned per statement is also variable. You can use Transaction Whiteboard to:
 - a) Open the existing Transaction Analyzer model of the transaction
 - b) Use distributions for the number of requests and sizes of the responses
 - c) Use Discrete Event Simulation to study the variability of the application's response time
- 3) You created a Transaction Analyzer model of an application that includes a significant amount of user think time. You want to study the response time of this application without user think time. You can use Transaction Whiteboard to:
 - a) Open the existing Transaction Analyzer model of the transaction
 - b) Change the dependencies on the client to only reflect tier processing time
 - c) Use QuickPredict or Discrete Event Simulation to predict performance
- 4) You are planning to add more robust authentication to an existing login transaction. You have a Transaction Analyzer model of the existing login and a separate Transaction Analyzer model of the authentication transaction. You can use Transaction Whiteboard to:
 - a) Open the two existing transactions into two separate Transaction Whiteboard models

- b) Copy the first transaction and paste it into the second Transaction Whiteboard model

A more advanced and flexible alternative is to use the Transaction Whiteboard Scripting API to dynamically invoke or call one transaction from the other. This technique is shown in the tutorial Transaction Whiteboard: Modeling an Application with Logic Scripts (Advanced).

- 5) You are interested in predicting the response times for an application that either does not currently exist or cannot be captured in AppTransaction Xpert. Based on your knowledge of how the application behaves, you can graphically draw the application layer data exchange in Transaction Whiteboard. More advanced users can take advantage of Python scripting and the Transaction Whiteboard Scripting APIs to introduce sophisticated logic into the application definition.

This last use case is the subject of this tutorial.

Overview

In this tutorial, you will use Transaction Whiteboard to model the behavior of the AppTransaction Xpert Capture Agent when finishing a capture.

The AppTransaction Xpert Capture Agent is a utility for performing packet trace captures. When a user finishes a capture, the packet trace is automatically transferred to Capture Manager. However, if the capture is performed over a low-bandwidth link or on a heavily congested network, it may not be desirable to immediately transfer the packet trace from the agent to Capture Manager.

Objectives

- Create a new Transaction Whiteboard model
- Use QuickPredict to determine the expected response time to four remote locations

Understanding the Application's Behavior

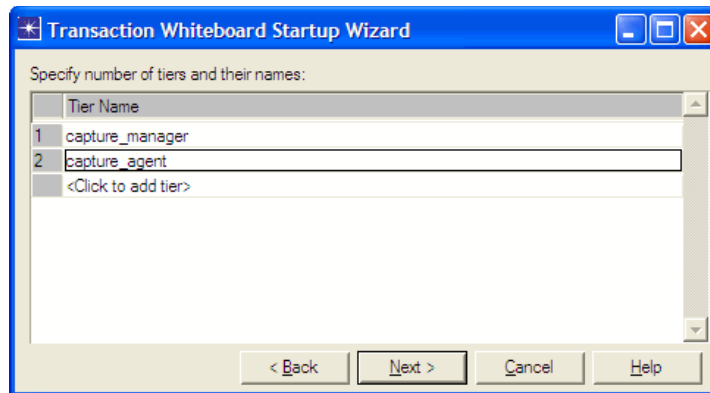
The following is a description of the application's behavior:

- Capture Manager sends a 50-byte message to the capture agent and receives a 100-byte response.
- Capture Manager sends a 600-byte request for the file; the Capture Agent responds by sending the packet trace.

Creating the Transaction Whiteboard Model

Procedure 11-1 Creating the Transaction Whiteboard Model

- 1 Choose **File > Open Model > Transaction Whiteboard...**
- 2 In the **Transaction Whiteboard Startup Wizard**, select **Create application manually** and click **Next >**.
- 3 In the dialog box that displays, specify the names of the tiers; enter **capture_manager** in the first row, and **capture_agent** in the second row, as shown in the following figure.



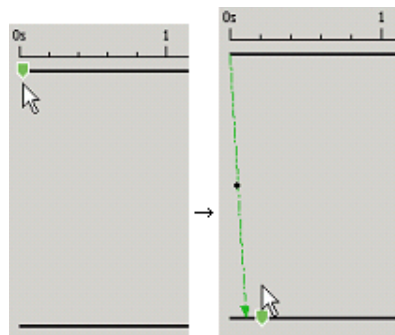
- 4 Click **Next >**.
- 5 Click **Finish** to close the startup wizard.

Notice that the Transaction Whiteboard model now contains the two tiers specified in the wizard: **capture_manager** and **capture_agent**. Next you will add messages to the model.

- 6 Choose **Insert > Message**.

Notice that the cursor in the **Data Exchange Chart** changes color from blue to green when you enter message creation mode.

- 7 Create the first message from the **capture_manager** to the **capture_agent**:

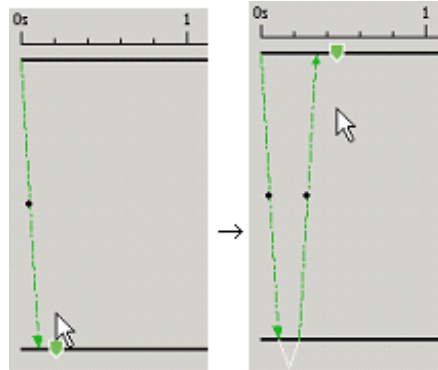


- 7.1 Click the **capture_manager** line to specify the source tier and start time of the message.

- 7.2 Click the **capture_agent** line to specify the destination tier. You should see a green arrow appear between capture_manager and capture_agent.

Note—Since this is the only message in the trace, the start time of the message is automatically shifted to 0.0 seconds.

- 8 Create the response from the capture_agent to the capture_manager:



- 8.1 Click the **capture_agent** line just to the right of the reception of the first message.
- 8.2 Click anywhere on the **capture_manager** line to complete the second message. (The reception time for the newly drawn message is determined by the **Simple Network Estimation** settings in the lower-left corner of the Transaction Whiteboard window). Also note that each new message that is created is automatically added to the **Message Editor**.
- 8.3 Right-click on a blank section of the **Data Exchange Chart** to exit message creation mode. The cursor changes from green to blue.

- 9 Edit the sizes of and the dependency between the first two messages:

- 9.1 Verify that the two messages are displayed in the **Message Editor**.

If the Message Editor is empty, select the two messages in the **Data Exchange Chart**.

- 9.2 Click in the **Bytes** cell for the first message (ID: 1). From the drop-down menu, choose **Edit...**, and then enter **50** as the new size.

➡ The color of the arrow changes to orange.

- 9.3 For the second message (ID: 2), change the **Bytes** value to **100**.

- 9.4 By looking at the value in the **Depends On** column for the second message, you can see that message 2 depends on message 1. Double-click on the **Processing Time** cell for the second message, and change the value to **0.005**, as shown in the following figure.

Message Editor:												
ID	Source	Destination	Bytes	Tag	Description	Subtask	Connection	Drop	Response	Depends On	Processing Time	User Time
1	capture_manager	capture_agent	50			<None>	1 (TCP)	N/A		Beginning	N/A	N/A
2	capture_agent	capture_manager	100			<None>	1 (TCP)	N/A		ID: 1	0.005000	0.000000

- 10 Create the next request/response pattern between capture_manager and capture_agent:

10.1 Click the **Insert Message** toolbar button.

10.2 Create another request message from capture_manager to capture_agent, following the procedure outlined in step 7.

10.3 Create the response message from capture_agent to capture_manager, following the procedure outlined in step 8.

10.4 Right-click on a blank section of the **Data Exchange Chart** to exit message creation mode.

10.5 Change the size of message 3 to **600**.

10.6 Change the size of message 4 to **100,000**.

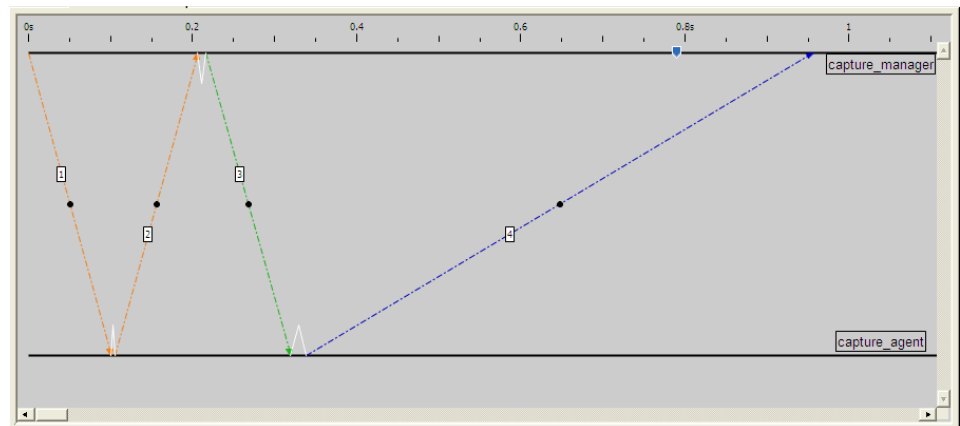
10.7 Change the Processing Time delay of message 3 to **0.01**.

10.8 Change the Processing Time delay of message 4 to **0.2**, as show in the following figure.

Message Editor:											
ID	Source	Destination	Bytes	Tag	Description	Subtask	Connection	Drop Response	Depends On	Processing Time	User Time
1	capture_manager	capture_agent	50			unnamed	1 (TCP)	N/A	Beginning	N/A	N/A
2	capture_agent	capture_manager	100			unnamed	1 (TCP)	N/A	ID: 1	0.005000	0.000000
3	capture_manager	capture_agent	600			unnamed	1 (TCP)	N/A	ID: 2	0.010000	0.000000
4	capture_agent	capture_manager	100000			unnamed	1 (TCP)	N/A	ID: 3	0.200000	0.000000

11 Change the zoom level to Full Zoom by right-clicking on a blank portion of the **Data Exchange Chart**, and then choosing **Full Zoom** from the menu.

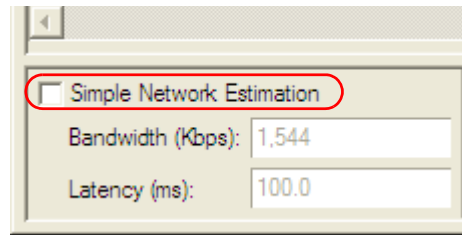
12 At this point, the **Data Exchange Chart** should look similar to the following figure.



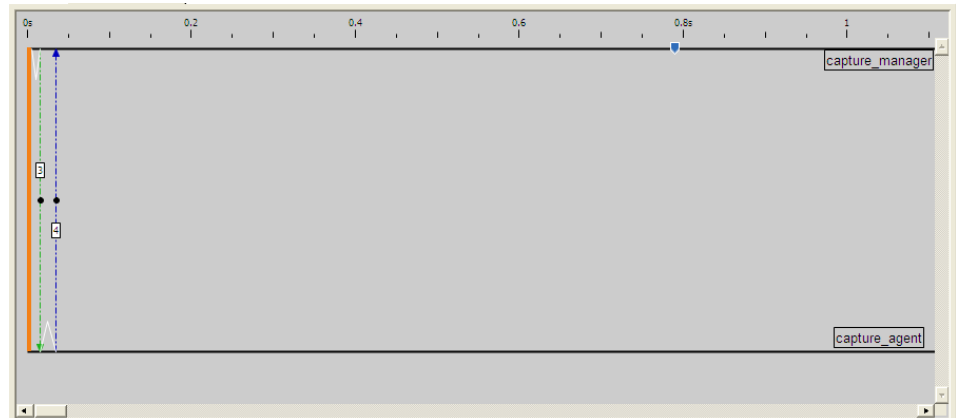
Notice that the large message from capture_agent to capture_manager takes longer to arrive than the three smaller messages. To make it easier to draw and edit messages, Transaction Whiteboard determines the reception time of each message based on a single bandwidth and latency value. This is known as **Simple Network Estimation**.

13 Temporarily turn off **Simple Network Estimation**:

- 13.1** In the lower-left corner of the Transaction Whiteboard window, uncheck the **Simple Network Estimation** checkbox, as shown in the following figure.



- 14** The **Data Exchange Chart** should look similar to the following figure.



With Simple Network Estimation turned off, each message displays straight up-and-down.

Note—The Simple Network Estimation settings can be used to make an application's behavior easier to visualize. The settings do not impact any simulation results.

- 14.1** Recheck the **Simple Network Estimation** checkbox.

- 15** Choose **File > Save As...** to save the Transaction Whiteboard model as **<initials>_tutorial_create_new_model**.

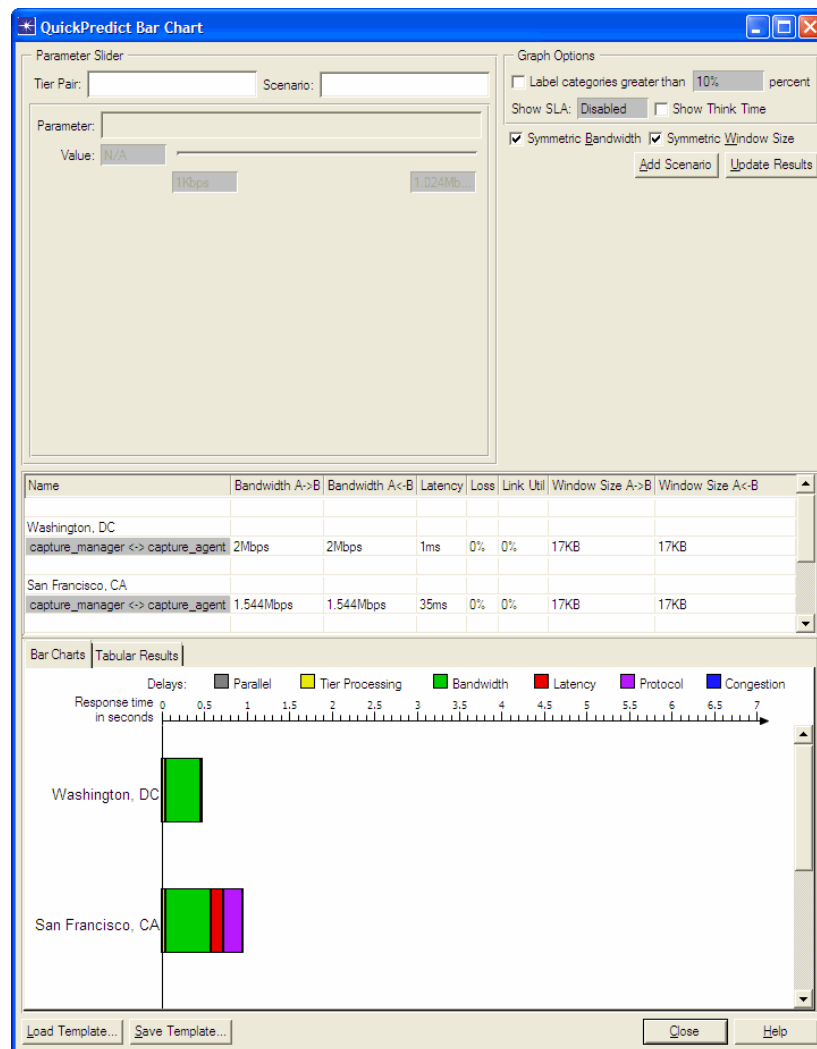
End of Procedure 11-1

Using Transaction Whiteboard, you created an application model that defined the communication between Capture Manager and Capture Agent when downloading a file. Next, you will use QuickPredict to evaluate how this application will perform on a variety of network conditions.

Using QuickPredict to Determine the Expected Response Time

Procedure 11-2 Determining the Expected Response Time with QuickPredict

- 1 Choose **Prediction > QuickPredict**.
- 2 Click **Load Template...**
- 3 Select the file **tutorial_new_model.ace.qpb** in
`<reldir>\sys\examples\AppTransaction Xpert\examples`.
<reldir> is the release directory where AppTransaction Xpert is installed. (In the Windows environment this is typically C:\Program Files\OPNET\<release number>.)
- 4 Click the **Open** button. The specified template is applied to the QuickPredict results and the following bar chart displays.



This chart shows the results of running this application from the four locations with the specified bandwidth and latency values. Notice that it takes significantly longer to retrieve the packet trace when running across a low-bandwidth link.

- 5 Click the **Close** button to close QuickPredict.
- 6 Save the model and close Transaction Whiteboard.

End of Procedure 11-2

Conclusion

In this tutorial, you learned how to use Transaction Whiteboard to model the behavior of the AppTransaction Xpert Capture Agent when finishing a capture. You also learned how to use QuickPredict to determine the expected response time to four remote locations.

Additional Details

- To add more flexibility to your Transaction Whiteboard model, you could have parameterized the size of the packet trace that the Capture Agent sends back to Capture Manager. You will see an example of this in the next tutorial.
- Instead of creating messages 3 and 4 using the **Insert > Message** operation, you could have copied and pasted messages 1 and 2, and then edited the message sizes and dependency values. This technique is useful when copying a large block of messages, especially when moving them from one file to another.
- You can create messages using keyboard shortcuts using Rapid Message Creation. For example, to create a message that starts at tier 1 and ends at tier 2, press CTRL-1 then CTRL-2. To create a message from tier 3 to tier 2, press CTRL-3 then CTRL-2.
- To create a repeating pattern of messages (such as a 120-byte SELECT statement and a 400-byte response, repeated 1,000 times), use a Message Group (**Insert > Message Group**).

For more information, see Message Group Operations.

- To change the same value on many messages in the Message Editor, use multi-message editing. This can be used, for example, to move 30 messages from one connection to another connection.

For more information, see Multi-Message Editing.