

27 Modeling Web Services in Transaction Whiteboard

You can model Web Service applications quickly and easily using imported WSDL (Web Services Definition Language) files. WSDL provides a model and an XML format for describing web services. A WSDL file defines the operations performed by a specific web service and the messages, data types, and protocols used by that service.

This feature enables you to

- Create traffic flows that more closely resemble the traffic patterns in the network
- Predict the performance of web service applications in QuickPredict and (if available) discrete event simulations
- Create web service messages by dragging operations from the WSDL Operations Palette into Transaction Whiteboard.

These derived messages reflect the operation times and message/argument sizes defined in the WSDL Operations Palette. Message sizes also reflect SOAP and HTTP overhead and default sizes for specific argument types, which you can define in Transaction Whiteboard.

- Visualize the set of operations defined in a WSDL file. The WSDL Operations Palette uses a treeview to show the following:

- Operations

Most operations consist of a request message (client -> web service provider) and a corresponding response message (web service provider -> client). Every operation has a default processing time at the provider tier.

- Messages

Every message has a set of arguments and a default message size. You can specify the message size as an aggregation of its arguments or as a bulk size.

- Arguments

Each message has a set of arguments; each argument has a specified type and size (in bytes). The number and size of arguments can determine the total size of a message.

You can save your custom operation/message/argument settings and reload them into the Operations Palette.

- Design and create web-service operations from scratch (that is, without importing from a WSDL file). You can define your own operations in the Template Palette and then drag them into Transaction Whiteboard.

You can also customize and configure application characteristics such as service-provider processing time, SOAP overhead, and message/argument size.

Modeling Web Service Applications: Workflow Description

The following steps outline the most common workflow. This workflow is described in detail in Procedure 27-2.

- 1) Import one or more WSDL files into Transaction Whiteboard.

If you do not have a WSDL file, or if you want to create a Transaction Whiteboard model without one, you can create web service operations from the Template Operations Palette.

- 2) Define the default settings that determine the sizes and processing times of new web service messages in Transaction Whiteboard:

In the WSDL Operations Palette:

Operation processing time

Message size

Argument size

In Transaction Whiteboard:

Processing delay at web service tier

- 3) Create web service messages by dragging and dropping from the WSDL Operations Palette (or the Template Palette) into Transaction Whiteboard.
- 4) Optionally, edit message sizes and processing times for individual messages using the Message Editor in Transaction Whiteboard (just below the Data Exchange Chart)
- 5) Predict the application performance using QuickPredict or discrete event simulations.

Creating Web Service Operations in Transaction Whiteboard (with WSDL Data)

Procedure 27-1 Creating a Web Service Application in Transaction Whiteboard

- 1 Choose File > Open Model > Transaction Whiteboard.
- 2 In the first window of the Transaction Whiteboard Startup Wizard, select “Create web service model from WSDL files.”
- 3 In the second window, specify the names of the tiers; for example, enter “Web Client” in the first row and “Web Server” in the second row.
- 4 In the third window, do the following:
 - 4.1 Select the HTTP Version (either 1.0 or 1.1).
For more information, see HTTP Version.
 - 4.2 Specify each WSDL file that you want to import, and the Transaction Whiteboard tier that will act as the provider for the services defined in that file.

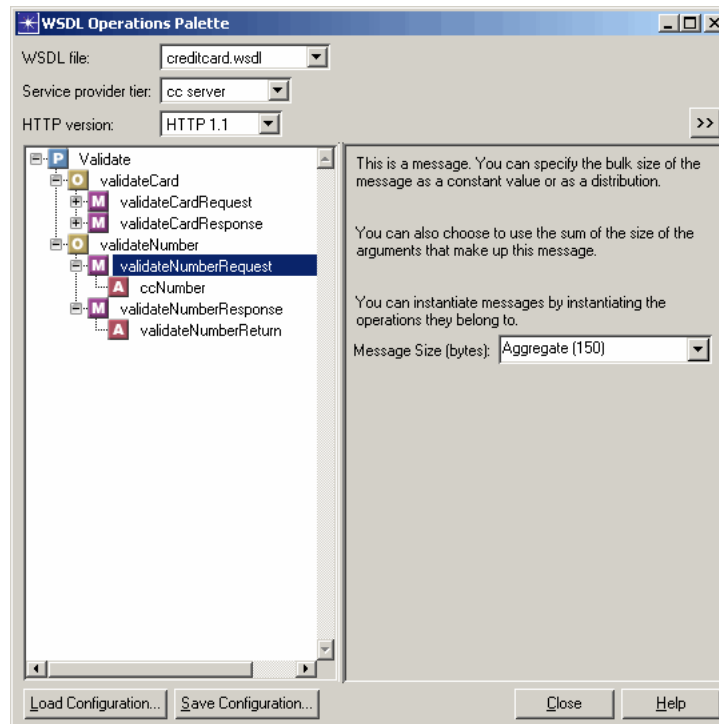
Figure 27-1 Transaction Whiteboard Startup Wizard

	WSDL File Name	Provider Tier Name
1	airport.wsdl	Airport Server
2	company.wsdl	Company Server
3	creditcard.wsdl	Credit Card Server
4	<Click to add WSDL file>	

< Back Next > Cancel Help

5 In the fourth (final) window, click Finish.

➤ The Transaction Whiteboard model file is created, and the WSDL Operations Palette appears.



5.1 Create operations from the WSDL Operations Palette as described in Procedure 27-2.

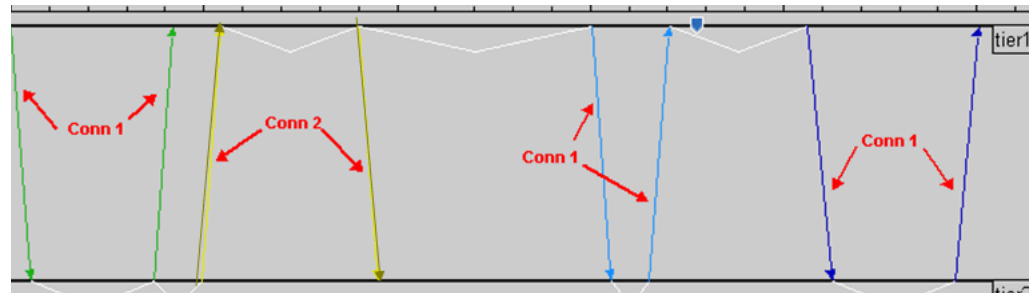
End of Procedure 27-1

HTTP Version

You can specify the HTTP version to be used by the web service model. The HTTP version determines how connections are reused by the SOAP messages.

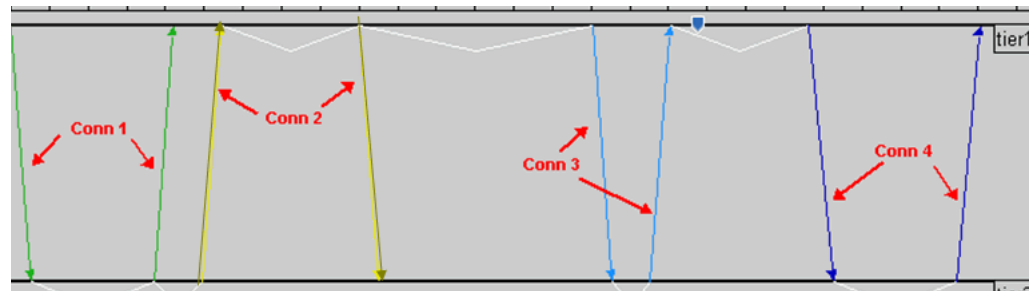
For HTTP 1.1, SOAP messages that originate from the same tier use the same initial connection. In this example, the third and fourth transactions use the same connection as the first transaction.

Figure 27-2 HTTP 1.1



For HTTP 1.0, SOAP messages establish a new connection for each operation regardless of the communicating tiers. In this example, there is no connection reuse.

Figure 27-3 HTTP 1.0



You can set the HTTP version in two places:

- In the Transaction Whiteboard Startup Wizard when creating a Transaction Whiteboard model file
- In the WSDL Operations Palette when viewing or creating operations

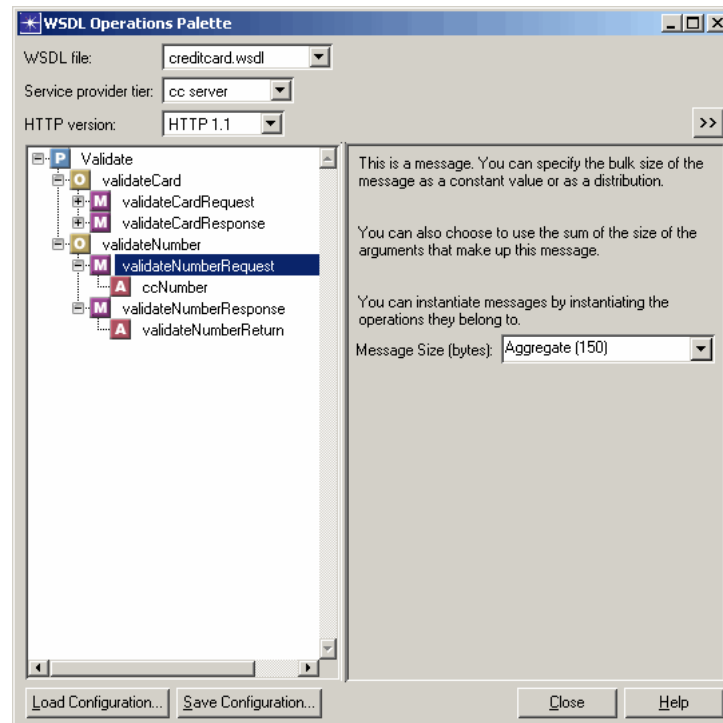
Creating Web Service Operations in Transaction Whiteboard

You can create web service operations based directly from WSDL files (Procedure 27-2) or without using WSDL files (Procedure 27-3).

Procedure 27-2 Creating Web Service Operations from WSDL Data

- 1 Import one or more WSDL files into Transaction Whiteboard.
 - To import into a new Transaction Whiteboard model file, perform Procedure 27-1.
 - To import into an existing Transaction Whiteboard model file, choose Web Services > Import WSDL Files.
- 2 Open the WSDL Operations Palette (Web Service > Open Operations Palette) if it is not already open. This window uses a treeview to show all defined operations (O), messages (M), and arguments (A) defined in an imported WSDL file.

Figure 27-4 WSDL Operations Palette



- 3 In the WSDL File menu, select the file that contains the operation of interest.
- 4 In the Service Provider Tier menu, select the Transaction Whiteboard tier that will act as the provider tier for that operation.
- 5 Optionally, configure the operation processing time or message/argument sizes for the operation of interest. To do this, select the item in the treeview and set the corresponding field in the right pane. If you have a saved configuration that you want to use, click Load Configuration and select it.

For more information, see *Specifying Message and Argument Sizes*.

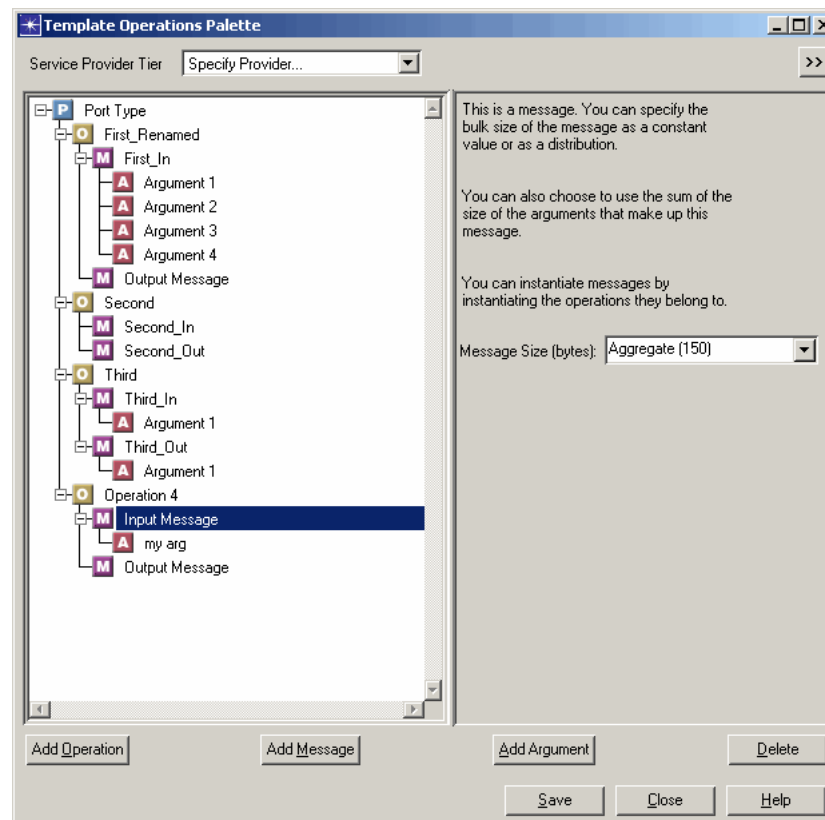
- 6 Optionally, click **Save Configuration** to save any custom operation, message, and/or argument settings.
- 7 Select the operation (**O**) from the treeview, drag it to the Data Exchange Chart in Transaction Whiteboard, and drop at the requester tier (not the Service Provider tier).
 - A new set of messages (in most cases, one request message and a corresponding response message) is created between the requestor tier and the Service Provider tier.
- 8 Optionally, edit the characteristics of the new messages. To edit a message, select it in the Data Exchange Chart and edit the fields in the **Message Editor** table.

End of Procedure 27-2

Procedure 27-3 Creating Web Service Operations without WSDL Data

- 1 In Transaction Whiteboard Open the WSDL Operations Palette (Web Service > Open Operations Palette) if it is not already open. This window uses a treeview to show all defined operations (**O**), messages (**M**), and arguments (**A**).

Figure 27-5 Template Operations Palette



- 2 Create a new operation, or edit an existing operation so that it has the characteristics you want. You can specify the following characteristics of an operation:
 - Operation processing time—Select the operation (**O**) and set the “Processing Time (sec)” field
 - Messages—In most cases, an operation consists of one request message (first) and one response message (second). To add and remove messages, use the Add Message and Delete buttons.
 - Message size—Select the message (**M**) and set the “Message Size (bytes)” field.

You can specify the message size as either an aggregate size (that is, as a sum of all argument sizes for that message) or as a bulk size (which overrides the argument sizes). For more information, see Specifying Message and Argument Sizes.
 - Arguments—To add or remove arguments, use the Add Argument and Delete keys.
 - Argument size—Select the argument (**A**) and set the “Argument Size (bytes)” field.
- 3 Optionally, click Save to save the current operation, message, and/or argument settings.
- 4 In the Service Provider Tier menu, select the Transaction Whiteboard tier that will act as the provider tier for that operation.
- 5 Select the operation (**O**) from the treeview, drag it to the Transaction Whiteboard window, and drop at the requester tier (not the Service Provider tier).
 - A new set of messages (in most cases, one request message and a corresponding response message) is created between the requestor tier and the Service Provider tier.

End of Procedure 27-3

Specifying Message and Argument Sizes

An operation definition in WSDL has a request and/or a response message. A message can include a list of arguments; each argument has a name and type (name:type), but not a size. Transaction Whiteboard requires a size (in bytes) for every message, and assigns a default size for every message and argument defined in the WSDL file. Transaction Whiteboard computes a message size based on the arguments and argument sizes defined for that message.

Note—When you create a web-service argument in Transaction Whiteboard, the resulting message includes SOAP overhead (based on the definitions in the Web Service Parameters. Thus, the actual size of a web service message in Transaction Whiteboard will be:

$$\text{<actual_message_size>} = \text{<message_size_in_palette>} + \text{<soap_overhead>}$$

You can specify Argument size in the following places:

- **Web Service Parameters**—Use this to specify default sizes for SOAP overhead and individual argument types (string, float, and so on).
- **WSDL Operations Palette**—Use this to define message sizes for specific message and argument definitions.
- **Message Editor**—Use this to change the size of messages in Transaction Whiteboard.

Web Service Parameters

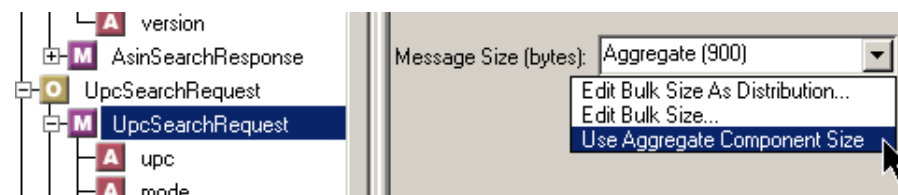
Transaction Whiteboard has a set of defaults defined for SOAP overhead and specific argument types. These settings help determine the default message/argument sizes calculated for a WSDL file when you import it into Transaction Whiteboard. For more information, see SOAP and HTTP Overhead.

Note—If you change these default settings, they will affect new messages only. These changes will not affect messages that have already been created in Transaction Whiteboard.

WSDL Operations Palette

To override the defined size of an operation or argument, select the item (**O** or **A**) in the treeview and edit the size in the right pane. You can specify the message size as a sum of the argument sizes or as a bulk size that overrides the argument sizes.

Figure 27-6 Editing the Size for a Message Type



Message Editor

To change the size of a message in Transaction Whiteboard, select the message in the Data Exchange Chart; then click in the Bytes field. You can edit the size of individual arguments or specify a bulk size for the entire message.

You can change the default size of individual messages and arguments in the WSDL Operations Palette. You can also save and reuse your custom definitions using the Save Configuration and Load Configuration buttons.

SOAP and HTTP Overhead

This section describes the settings for SOAP/HTTP overhead, which affect the size of web service messages that are created in Transaction Whiteboard. To change these settings, choose Web Services > Specify Web Service Parameters.

Note—Specifying custom SOAP overhead sizes is an advanced feature. Do not change these default settings unless you know the specific details of your SOAP implementation and the messages that are generated by the tiers of interest.

If you change any of the default settings, the new web service parameters are stored in the following files:

- `<user_home>/op_admin/ace_import_configs/ace_wb.gdf`
(HTTP and Soap overhead)
- `<user_home>/op_admin/ace_import_configs/ace_soap_elements.gdf`
(Default SOAP argument length)

To restore the default settings, exit AppTransaction Xpert, delete one or both of these files, and then restart AppTransaction Xpert.

Table 27-1 SOAP Overhead Options and Default Settings

SOAP Section Name	Description	Default Overhead Size (bytes)
HTTP Header (Request)	The average total size of HTTP headers present in the request. This includes elements such as cookies, hostname, etag, and so on.	203
HTTP Header (Response)	The average total size of all the HTTP headers present on the response.	74
XML Tags	Length of the <code><?XML ?></code> processing statement at the start of the SOAP message	38
SOAP Header	Length of the header section of the SOAP message not including the header tags	0
SOAP Envelope Tags	Total length of the tags that make up the SOAP message envelope	362

Table 27-1 SOAP Overhead Options and Default Settings (Continued)

SOAP Section Name	Description	Default Overhead Size (bytes)
SOAP Body Tags	Total length of the SOAP body tags	23
Namespace Tags	Total length of namespace declarations in the SOAP message	4
Message Name Tags	Length of the tags that define a message. This specifies the length of the tags only, and does not include the message body.	75
Argument Type Tags	Length of the tags that define an argument type.	23
Array Tags	Length of the tags that define an array.	30
Complex Type Tags	Length of the tags that define a complex type	30
Parameter Namespace Tags	Length of the tags that define a namespace for the parameters	0

Computing SOAP Message Sizes: Example

This example shows how Transaction Whiteboard computes the size of a SOAP message. When distributions occurs, this example uses mean outcomes.

Figure 27-7 Example SOAP Message

```
POST /creditcard.asmx HTTP/1.1
Host: webservicess.primerchants.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.paymentresources.com/webservicess/GetBankCardDebitStatus"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetBankCardDebitStatus
      xmlns="http://www.paymentresources.com/webservicess/">
      <TransID>string</TransID>
      <MerchantID>string</MerchantID>
      <RegKey>string</RegKey>
    </GetBankCardDebitStatus>
  </soap:Body>
</soap:Envelope>
```

Computing the Argument Sizes There are three string arguments. Suppose the user specified the following sizes for the arguments:

- TransID: Constant (16)
- MerchantID: Uniform (20, 40) = 30 bytes
- RegKey: Constant (32)

The total size of the arguments is obtained by adding the mean outcomes of the three distributions, which gives us a value of $16 + 30 + 32 = 78$ bytes.

Computing the SOAP Overhead The following SOAP sections are present in the previous message:

- HTTP Request Header: 203 bytes
- XML Tags: 38 bytes
- SOAP Envelope Tags: 362 bytes
- SOAP Body Tags: 23 bytes
- Message Name Tags: 75 bytes
- Argument Type Tags: $23 * 3 = 69$ bytes
- Argument Name Tags: $\text{strlen}(\text{arg name}) * 2$
 - TransID: $7 * 2 = 14$
 - MerchantID: $10 * 2 = 20$
 - RegKey: $6 * 2 = 12$
 - Total = $14 + 20 + 12 = 46$

Total SOAP overhead = $203 + 38 + 362 + 23 + 75 + 69 + 46 = 826$ bytes.

Computing the Overall Size of the SOAP Message The overall size of the message is the sum of the size of the arguments and the SOAP overhead.

Size of SOAP Message = $78 + 826 = 904$ bytes.