Working with Transactions

A transaction defines a logical 'unit of work'. For example, a fictional BankingService's creditAccount and debitAccount methods may use a transaction. A transaction allows the application to ensure that any database changes made by the application server while executing the two methods will be made permanent if and only if both are successful, thus maintaining consistency. Refer to the NonStop TM/MP Application Programmer's Guide for further information about transactions on NonStop Server systems.

SOAPam Server supports NonStop TM/MP (TMF) transaction management. You, as the Web service developer, specify in your Service Definition File (SDF) whether a method requires an active transaction when its Pathway or Guardian server is accessed and whether the Web service client or SOAPam Server will control it.

When SOAPam Server alone controls the transaction, it uses the settings specified in the SDF to determine whether to commit or abort the transaction. When a Web service client controls the transaction, it includes information in the SOAP request that dictates if the SOAPam Server should suspend, commit or abort the transaction. Client applications may also use the built-in transactionControl service's TransactionBegin, TransactionCommit and TransactionAbort methods to control transactions. In either case, SOAPam Server performs the necessary NonStop transaction management, but with client-controlled transactions, the client application determines when to begin, commit or abort a transaction.

Allowing the SOAPam Server to control the transaction hides the details of the transaction from the client application. However, when the Server controls the transaction, the transaction can consist of only a single method request-response. When a Web service client controls the transaction, the transaction can include multiple method request-response exchanges from multiple server applications running on one or more NonStop Server systems in the same Expand network.

The following summarizes the features of SOAPam Server controlled versus client controlled transactions:

Server controlled transaction

- consists of a single request-response
- commit/abort criteria coded in the Service Definition File
- client application includes no transaction management code or logic

Client controlled transaction

- may consist of any number of request-response exchanges
- suspend/commit/abort criteria coded in client
- multiple concurrent transactions supported
- client application includes transaction management code and logic
- SOAPam Server makes the actual NonStop transaction management API calls on behalf of the client

Enabling Transaction Support

If you use the Service Definition Wizard to create your SDF, select the *This method requires a transaction* check box on the Wizard's Method Definition page to access its Transaction Control page. Then click the *Allow SOAPam Server to manage the transaction state* or *Allow the SOAP client to manage the transaction state* as appropriate.

If you're building your own SDF, you specify that a method uses a transaction by including a <transactionControl> element within a parent <method> element. If the <transactionControl> element specifies 'clientControl="1"', allowing client applications to control the transaction, you must also define a <header> or a <parameter> of type txControlRequestType to be passed in the SOAP request and a header or parameter of type txControlResponseType to be passed in the SOAP response. If both client and Server are allowed to control the transaction, the <header> option will allow the method to be invoked with or without a txControlRequestType header.

Controlling a Transaction

Using txControlRequestType, a client application can specify that, on method completion, that SOAPam Server should "suspend", "commit" or "abort" the transaction. A suspended transaction can be used in a subsequent method invocation of the same or another method that allows client transaction control. Typically, with the final method call in its unit of work, the client application specifies that the transaction be committed or aborted. However, there may be a situation in which the client may want to end a suspended transaction without having to invoke an application server method. SOAPam Server provides the built-in transactionControl service for this purpose. It supports the TransactionCommit and TransactionAbort methods, which a client application can use to commit or abort a suspended transaction without any interaction with the application server.

Whether server-controlled or client-controlled, SOAPam uses the specified list of 'validReplyCodes' when determining whether to suspend, commit or abort a transaction. At run time, SOAPam Server compares the reply code (the signed integer value contained in the first 2-byte word) of the server's reply interprocess message to the list of validReplyCodes. This behavior is designed to take advantage of the convention, commonly used by Pathway servers, of setting the first word of the reply message to a value that indicates the success or failure of the request.