# FAQ

**Why does the Resolver Service fail with 503 Service Unavailable message?**

There can be many reasons for this. One thing to do is to check the Windows Event log for the following error:

"The Module DLL C:\Windows\system32\inetsrv\rewrite.dll failed to load. The data is the error."

This indicates that the IIS URL Rewrite module needs to be repaired. The reason this problem occurs is unclear, but it is an issue that is encountered in a variety of scenarios, and is not specific to the ESB Libraries. Go to Control Panel/Programs and Features and locate the program entry for IIS URL Rewrite Module 2. Right-click and select 'Repair'. After the repair is completed, restart the resolver service application pool in the IIS administration console.

If the problem persists, try uninstalling and re-installing the Rewrite module. 32 and 64-bit versions are available from <http://www.iis.net/downloads/microsoft/url-rewrite>.

**Why am I forced to use continuation across multiple BAM steps in an orchestration?**

The ESB Libraries support a ‘graduated’ approach where BAM steps can be applied at four levels:

1. ‘All at once’, by calling the OnStep() method of a TrackpointDirectiveEventStream or an OESTrackpointEventStream.
2. ‘Grouped by function’ by using the extended API of the event streams. For example, all data updates can be applied at once.
3. Individually, by using overloaded versions of the standard event stream methods.
4. Directly by using the standard event stream API. In this last case, the BAM Steps defined in BRE policies are not used.

Developers may wish to define a number of BAM steps to be used at different locations in a single orchestration. In this case, each step, except for the first, must be defined as a continuation using the standard continuation mechanism implemented by the BAM Event Observation API. This is supported in the BAM Steps defined using BRE policies. This may feel awkward and unnecessary, given that all the code runs in the context of a single orchestration.

Unfortunately, there is no safe or sensible alternative. There are several considerations here, but the main two are as follows:

1. In order to ensure that the BAM Steps are defined in a logical and consistent way, the code needs to validate each step as a logically complete unit of work. This validation is built into the BAM Interception Framework, but only supports the notion of continuation and relationships across multiple steps. If the ESB libraries introduced a concept of ‘extension’ of an activity using the same activity ID, they would need to perform careful validation of steps to ensure that the ‘extensions’ are logically consistent with the steps they extend. This, in turn, would require developers to model extensions explicitly in the BAM Step rules. The programmatic model would be almost identical to continuation, but with additional validation rules. This would offer no effective simplification of the existing continuation model.
2. The ‘continuation’ mechanism handles the complexity of out-of-order delivery of event information to the BizTalk message box when using buffered event streams. A lightweight ‘extension’ mechanism would be fundamentally unsafe. Consider a scenario where a developer continues an activity across multiple orchestrations. BAM continuation is safe. Use of a lightweight ‘extension’ mechanism would not be safe.