

Active Integrator – Questions and Answers

Subject: Questions that have been asked in the past **Date:** 26th Feb, 2001
Release MQ - Tuxedo

This document serves to bring together a number of questions and answers that we have encountered in the past with the Active Integrator MQ – Tuxedo release. This document may help to stimulate further thoughts on the functional and operational requirements and provide a mechanism for answering any early integration issues.

Questions on scalability and performance

1) **Current tested performance of the gateway.**

The gateway has been tested on an NT desktop single processor PC sub 500mhz with memory of 128mb. The gateway was required to pass a test of 10 substantial transactions per second. This requirement was met.

2) **You have a requirement for between 1,000 and 3,000 concurrent transactions through Clarify.**

We need to refine this transaction requirement in terms of, volumes in and out of Clarify and estimates of expansion. Also we need to explore the 3,000 number in more detail. For example if they are Clarify transactions then this throughput is based upon the time they take to be processed in Clarify. So on the assumption that this takes an average of 3 seconds then we are dealing with a throughput of 1,000 transactions per second. Regardless with these transaction levels we are confident that the Active Integrator can scale beyond even this level of transaction volume and that if provided with transaction type detail we would be able to accurately size the Hub required to support the required throughput.

3) **Where should the gateway reside.**

The question is specifically should the Active Integrator reside on the same hardware as the MQSI hub. For simplicity of architecture reasons we would prefer the system to reside on the Tuxedo server. The question should be asked as to whether the customer has an MPP Tuxedo environment; if this is the case then the Active Integrator could reside on any distributed box in the environment. Final architecture would be defined during the definition workshop process. We have experience of a number of different approaches in both the Tuxedo and MQ environment.

4) **What about the scalability of the gateway.**

If you refer to the model diagram, attached to this note, we have in effect, no limit on the number of queue process that we can run in or out of the Clarify environment.

a) **Transactions out of Clarify?** As a transaction at our gateway is "almost instantaneously" put onto the MQSI queue, i.e. the translation process is very rapid and there is no delay from the queues for Asynchronous messages.

b) **Transaction into Clarify?** The Tuxedo environment has a limit of 52 concurrent transactions per client instance. Therefore, we must ensure that we scale up the number of clients to remove this restriction. Hence the transaction volume into Clarify is particularly important.

c) The Gateway handles multiple queues to multiple clients without restriction, whichever process is ready picks up, converts and passes the message through. Therefore the configuration of the gateway itself will not create a bottleneck (indeed this is a specific benefit and ensures that available resources are optimised).

5) **We will need to run a performance test as part of the implementation, either at the customer site or Strategic Thought.**

We have a number of utilities already available to help in the measurement of performance. For example on the NT platform we would need to build a simulator that can fire controllable volumes of transactions at the gateway and then use the standard NT performance monitoring software to measure the environment performance and stress, based on the known volume of transactions coming from the transaction simulator.

6) **In the testing process we will also want to look at "round trip" transactions.**

How long does it take to process a message from the user screen and back again measuring the time taken at each stage; screen to queue, MQSI to gateway, Gateway to Clarify, Process, Clarify to Gateway and gateway back out the MQ environment back to the user.

7) **What's the performance in regards to how many MQ messages can be processed per second? How does Active Integrator deal with persistent messages?**

The performance is very good. On one site a single NT work station was achieving over 10 messages a second throughput. The gateway has been specifically designed to address issues associated with scalability. TUXEDO is not thread safe so you cannot have a multi-threaded application. However you can start multiple instances of the TUXEDO server supporting outbound messages either all within the same MSSQ or else multiple separate servers, potentially in separate TUXEDO groups and potentially on different machines in an MP installation if you want to.

Similarly you can have a single server monitoring many queues, you can have several servers all monitoring the same set of queues or you can have multiple servers all monitoring separate queues. Its highly flexible if very serious scalability is required. Basically the gateway is a very light weight high performance and throughput engine.

We also have a performance test rig for the gateway, therefore we can store a representative mix of transactions and fire them at the gateway on various servers that we have in our offices or at the customer site. We are happy to include the performance requirements as part of the Acceptance Criteria, because we can test accurately prior to agreement. This also enables effective recommendations on Hardware

Questions on Technology

1) **How's the XML DTD created?**

Actually it's not, as a consequence of long and heated debates with customers! We have three mechanisms that are alternatives, plus a fourth option:

The **first** and most popular option after long debate is that when it is required to translate a TUXEDO buffer into XML, the service has a default message name and the document is created with that as the root. Then all other attributes are read out of the FML buffer and created as a separate element in the XML message. Thus it is a long and flat XML message. While this appears to be a somewhat bizarre way to structure it the message most of our customers using XML are also using MQSI. In this circumstance they actually specifically did not want to define DTDs for their documents but wanted the documents structures controlled within the ESQ of MQSI as they translated from one format to another.

The consequence is that in MQSI it is very easy to see an array of fields and to extract data from those fields and place it in the appropriate place. In addition this flat XML structure is an exact representation of the FML being provided. Consequently we have used this mechanism in most places.

If you start to think about the issues of optionality in messages you will immediately determine that there is a significant challenge in determining the mapping between FML and XML particularly bearing in mind that most clarify installations are on TUXEDO 6.4. TUXEDO 7.1 does actually allow the use of proper records in FML.

The **second** mechanism we support is a pass through mechanism. If an XML message is prepared in the clarify environment it can be segmented into sections (the largest string field appears to be about 2k in TUXEDO) and multiple instances of this field are transmitted. The FML fields are then retrieved and concatenated back into the original XML message and transmitted.

The **third** mechanism we support is to use a 'dummy' XML message and just traverse that searching the FML buffer for fields of a particular name, and counting the number of times we have searched for each field, so that we extract the correct value. However that does not deal with optionality, and neither does a DTD.

Basically at the end of the day if there is any optionality in a DTD then you have a problem. Further, if the document contains records and you have an indeterminant number of records in them then you have a bigger problem. All of these issues can be resolved (because we did on another large TUXEDO project (however it is a pig, and requires large amounts of additional control information in the FML buffer to contain array counts and such like, or placing of values in the FML structures with holes in them).

Therefore we went the way we have, and just act as a pass through so a flat FML structure is turned into a flat XML record and passed on to MQSI for processing.

With TUXEDO 7.1 we will officially support XML buffers, if and when Clarify does.

- 2) **In the MQ Tuxedo Gateway product specification, it mentions two XML parsers(DOM XML API and XERCES C++ XML parser). Are both being implemented and where are they exactly used? If so, which version of XML DOM is being used?**

The parser we use is the XERCES-C parser which we down load from <http://xml.apache.org> where it is available. It is derived from the IBM equivalent. We only use the DOM model, and this handles very large messages of over 2M without any problem. We use this because the gateway is written in C/C++ for performance and because you cannot use Java for a TUXEDO service. You do not need to do anything as we will provide the DLL as part of the installation.

3) **Have you any sample Clarify code that uses the Adapters?**

We do but these are all components that are provided by customers. In having to be careful about NDA and IPR issues we are unfortunately unable to pass them on. However there are examples of using Clarify services to call and be called by the gateway. These are very straight forward and follow the form defined in the clarify manual on page 233 and 234. Providing you use the servicemessage mechanism for services you wish to have invoked by the gateway, and similarly the servicemessage mechanism to call services that will be intercepted by the TUXEDO server part of the gateway to send messages out.

Given our comments above on XML what is probably more interesting are worked examples of MQSI 2.0 nodes that will structure flat XML back up into structured XML and that we can do very easily and provide to you in an MQSI export format.