

TITLE: ``Distributed CICS in practice to maximize your IT investment'

HOST: Hi, and welcome to the 'Did you say Mainframe?!' podcast series. This is where we regularly interview IBM technical experts who can help you to understand important IBM mainframe hardware and software issues. I'm your host Sherrie Abshire.

HOST: Today we're going to talk about TXSeries - the CICS product for distributed platforms.

HOST: Our guest today is Iain Boyle from the CICS team in the Hursley laboratory. Iain it's great to have you here.

SME: I'm delighted to be here.

HOST: Before we begin, I'd like to mention to listeners that there's a great opportunity to test drive TXSeries. I'll be giving out more information at the end of the discussion.

SME: [optionally responds]

HOST: So Iain, remind us, what's TXSeries and why is it relevant to the mainframe?

SME: A long standing member of CICS family, TXSeries for Multiplatforms is a CICS based transaction processor designed to run on distributed platforms. TXSeries shares many of the same design principals as mainframe CICS Transaction Server whilst providing a carefully selected subset of functionality. For example, the newest release of TXSeries, Version 6.2, is continuing to enhance its CICS Application Programming and System Programming command coverage whilst providing new functionality. In fact many IBM mainframe customers will use TXSeries to support their mainframe solutions and their CICS implementations because of benefits such as common Application Programming Interfaces and extensive intersystem communications.

If we talk about capabilities, there are 2 typical usage patterns for TXSeries users. The first is for its distributed transaction-processing capabilities, effectively running fully functional CICS applications on distributed platforms such as AIX. The second and perhaps more commonly seen, is to use TXSeries for its

transactional-integration capabilities, as TXSeries has extensive support for many Enterprise Systems, including CICS, IMS, DB2, WebSphere MQ and WebSphere Application Server. In this usage scenario, TXSeries provides the ability to run intelligent business logic using locally available business data whilst also integrating and communicating in a transactional manner with enterprise systems.

HOST: Sounds good, can you give me some examples?

SME: Yes of course. The 1st example is where customers typically use the mainframe version of CICS (that is, CICS Transaction Server for z/OS running on a System z mainframe), to exploit the exceptional reliability and quality of service, or perhaps to manage vital business data. They then use TXSeries to consolidate and group terminals, thereby controlling traffic and mainframe access and removing the burden of terminal processing from the mainframe system. This has the advantage of reducing cost and complexity to the business whilst introducing some level of manageability to collections of terminal users.

Another example might be to use TXSeries for local transaction processing in line with local customer behaviour, while also allowing some access to corporate information stored in a central office location.

Another example of TXSeries usage is where a time-critical or cost-driven project is required to deliver a simple and rapidly deployed transaction processing environment. Once the project is successfully deployed, the customer then has the option, at a later date and in a controlled manner, of moving the whole environment to the mainframe.

HOST: So, considering the vast number of CICS usage, what benefits do customers see when using TXSeries to do this kind of mediation? How specifically does TXSeries work with CICS TS that's different from other products?

SME: This is done by using a very secure, direct and consistent method called CICS Inter-system Communications. This is the ability for two CICS systems to synchronously or even asynchronously communicate across a network or remote interface. The CICS systems do not have to be of the same type or even running on the same operating system. So it's possible for TXSeries running on System P hardware to

communicate with CICS Transaction Server running on System Z hardware.

For example, it is possible to start a transaction in TXSeries which updates data in a locally attached database, and then, still within the same transaction, access data or programs on a CICS Transaction Server system. At the end of the transaction, TXSeries will ensure all data updates, regardless of where the data is located, are consistent.

HOST: You say there's a latest release? Tell me more..

SME: I am pretty excited about this release, it is a result of listening and responding to our customers. Version 6 first became available at end of 2005, and is the most significant version in recent history. Following further releases in the past few years, the latest delivery, V6.2 builds on earlier releases. Functionality previously provided by DCE and Encina, and was once a prerequisite for TXSeries, has been replaced with additional functionality in TXSeries. The whole TXSeries environment has now become more robust and resilient. For example, enhanced recovery processing now makes TXSeries regions more resilient against abnormal terminations due to asynchronously aborted tasks, and FORCEPURGE requests can withstand unplanned region terminations caused by events such as timeouts and network failures.

Notably, the Web based administration console, introduced in V6.1 has been enhanced in V6.2 with advanced features including more easy-to-use workload management configuration and monitoring facilities. For example, you can see a detailed monitoring profile. It has a number of similarities with the WebSphere Application Server Administration Console, so it helps users of both products (of which there are a huge number of users exploiting the web interface provided by WebSphere) to be more efficient.

So TXSeries V6.2 further increases user effectiveness, system resilience and interpretability for many businesses worldwide.

Furthermore, earlier this year, TXSeries extended its platform support to include HP Itanium. TXSeries is also the CICS testing and debugging component supplied with Rational Application Developer for System Z. This allows

CICS application developers to test and debug their CICS applications before deploying to CICS Transaction Server.

HOST: Tell me more about the integration options with WebSphere - I thought WAS only runs J2EE code, how can a Java class running inside a Java Virtual Machine (JVM) access CICS programs running in a TXSeries system on a potentially remote machine?

SME: The answer is to use the J2EE Connector Architecture (or JCA) provided by the CICS Transaction Gateway. Let me give you an example of a European financial customer I worked closely with.

Their challenge is to integrate their J2EE™ web applications, running on WebSphere Application Server, with back-office core applications for processing and updating customers' insurance policies, which runs on TXSeries. They need the core processing to continue smoothly whilst making sure their process involving these web applications provide continuous availability to customers. As you can imagine, their systems are quite complex having evolved over many years with different technologies and acquisitions. So TXSeries for Multiplatforms provides an "intelligent" gateway linking the customer's Java based WebSphere Application Server Web environment with Oracle databases and ERP based solutions. TXSeries provides COBOL-based core business applications and coordinates all information and application updates, both locally and remotely.

HOST: Great! Sounds like TXSeries is an extensible and useful component of application infrastructure then.

SME: Yes, you can utilise your existing C, C++, COBOL and PL/I skills and assets, while providing a future proof solution. As businesses grow, some of our customers may choose to run some of their TXSeries CICS applications on CICS Transaction Server. And, as the European insurance example show, TXSeries can integrate data and applications between a variety of enterprise systems whilst reliably processing business transactions.

HOST: Iain that was really interesting - thanks for joining us. We'll be looking to learn more.

SME: Thanks for giving me the opportunity to talk about TXSeries.

HOST: That wraps up this discussion. To find out more about the TXSeries test drive I mentioned earlier, go to the description for this podcast at:

<http://www.ibm.com/software/os/systemz/podcasts/websphereonz/>

HOST: There you will also find the URL for registering for the IBM Impact 2008 event which is being held in Las Vegas in April.

HOST: Join us next time as we talk about another important mainframe topic. For now, this is Sherrie Abshire saying “Thanks for listening