

TITLE: CICS and Web 2.0

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.

Today we're going to talk about a new Web function in CICS Transaction Server.

Our guest today is Peter Havercan from CICS Development at the IBM Hursley Lab in England. Welcome, Peter.

SME: Thank you Sherrie. It's nice to be here.

HOST: Before we begin, I'd like to mention to our listeners that we have a teleconference titled 'Extending SOA and CICS with Web 2.0'. In addition we have a supportpac that is available to download that provides sample code to demonstrate how to generate atom feeds to publish data contained within CICS resources or obtained by running a CICS application. I'll be giving out more information at the end of this podcast.

HOST - Question 1: Peter, tell me about the new Web function in CICS.

SME - Answer 1: Well, first I want to talk about the new way of using the World Wide Web that has come to be known as Web2.0. This is a new presentation style that's distinguished by highly dynamic and interactive web pages, also known as Rich Internet Applications. It includes new concepts like syndication feeds, Mashups, Blogs, and Wikis. It's mainly driven by JavaScript programs that are downloaded with the web pages and executed as part of them.

HOST - Question 2: So how does CICS work with these new technologies?

SME - Answer 2: Most of these technologies operate within a Web browser, and interact with servers like CICS using existing HTTP protocols, so CICS is already well placed to operate with the new applications. CICS is not going to try to interact with all of these technologies, so we're just going to focus on a relatively new protocol called Atom, as it turns out that this can be the basis for several of the other technologies. Atom is a syndication feed protocol, similar to RSS, which is a way of publishing short items like news headlines to a client

program known as a feed reader or aggregator. Atom overcomes a number of deficiencies in RSS, and it's formally defined by a couple of standard definitions or RFCs.

Atom can also be used for publishing articles to a Web Log, also known as a blog. But an even more interesting application of Atom is to feed data into a mashup, which is a JavaScript program running in a browser that can compose the results from several different servers to produce a single composite graphical result.

HOST - Question 3: Are the new functions integrated into the CICS product?

SME - Answer 3: No. The new ideas of using Atom in CICS have only arisen in the last few months, and well after CICS Transaction Server 3.2 was completed. But we think that these new ways of interacting with CICS are so important that we don't want to delay them by waiting for a future release of CICS. So we have produced some early exploitation of Atom in a new CICS SupportPac, number CA8K, which will run on CICS TS 3.1 and 3.2. But I have to emphasize that this is experimental code, just to introduce CICS customers to the new technology and to gauge customer reaction.

HOST - Question 4: What's the main function of the SupportPac?

SME - Answer 4: The main idea behind the Atom SupportPac is that you should be able to easily publish data contained within CICS as an Atom feed. The main emphasis is on publishing the contents of a Temporary Storage Queue as a feed. This means that it becomes very easy to publish an Atom feed from CICS - just write data to a Temporary Storage Queue and let the SupportPac handle the details of constructing and publishing the feed. And although the SupportPac does not specifically handle CICS files, it should be fairly easy to extend the supplied code to do this as well.

And if you're ambitious enough to create a mashup application in a Web browser, you can use exactly the same ideas and interfaces to feed the CICS data into the mashup program. There are a number of tools available for creating mashup applications very easily; such as QEDWiki, which you can download from the Alphaworks website, or the soon-to-be-released Lotus Mashup product. Both of these can accept data from an Atom feed - such as the one supplied by this SupportPac - and integrate it

graphically with data from other sources, such as Google Maps, Yahoo! Weather, or a stock quotation service.

As well as being a simple feed server, the SupportPac also gives you a RESTful interface to CICS Resources.

HOST - Question 5: What does RESTful mean?

SME - Answer 5: RESTful is one of those new terms that goes along with all the other Web2.0 terminology. It's associated with the acronym REST, which means Representational State Transfer. It's a way of accessing resources over the Web by representing them as a URL. You can perform operations against the resource using the four HTTP methods POST, GET, PUT and DELETE. POST creates a new resource and assigns a new URL for it. GET retrieves the current version of the resource, PUT updates a resource, and, of course, the DELETE method deletes it. In the RESTful context a resource doesn't necessarily mean an entire CICS resource. It can identify just a single item within a queue, or a single record within a file, for example.

HOST - Question 6: How does the SupportPac work?

SME - Answer 6: Atom is mainly based on XML over HTTP, just like SOAP. So, in principle, all CICS has to do is compose a standard Atom XML document and deliver it over HTTP using CICS Web Support. So that's exactly what the SupportPac does. But to simplify the composition of the Atom document, it allows you to compose a prototype Atom document in a configuration file. It uses the CICS PIPELINE resource to do this, as that already uses an associated configuration file in the UNIX System Services file system.

So the main Atom support module executes as a custom handler module in a CICS PIPELINE. This interprets the incoming feed URL request and uses the configuration data to determine which CICS resource is to be accessed. The configuration data for the Atom feed is extracted from the handler parameter list section of the pipeline configuration file. After interpreting the URL and the configuration file, the handler module chooses and links to a resource-dependent service routine to access the selected resource. The service routine constructs the Atom content from the resource and returns it to the handler module, which complete the Atom feed response and returns it to the client.

HOST - Question 7: What does the SupportPac contain?

SME - Answer 7: The SupportPac is packaged as a number of CICS sample programs, together with some JavaScript and some HTML to give you an example of how they might be exploited. The main driver program is the pipeline handler module that I just described. There is device-dependent service module for handling Temporary Storage Queues and a sample Cobol program that is specifically for handling the FILEA sample file.

HOST: Peter, thank you so much, that was really interesting.

SME: Thanks for giving me the chance to talk about Web2.0 and CICS.

HOST: Well, that wraps up this podcast discussion. To find out more about the Teleconference and the SupportPac we talked about earlier, please go to

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

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

URL for SupportPac

http://www-1.ibm.com/support/docview.wss?rs=1083&context=SSGMGV&context=SSZMG3&context=SSNQZF&q1=ca8k&uid=swg24018619&loc=en_US&cs=utf-8&lang=en

Teleconf

<http://www-306.ibm.com/software/os/systemz/telecon/11jun/>