



CICS Transaction Server

## Extending SOA and CICS with Web 2.0

Peter Havercan  
([peter\\_havercan@uk.ibm.com](mailto:peter_havercan@uk.ibm.com))

11 Jun 2008

© 2008 IBM Corporation

## Important Disclaimer

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.

WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

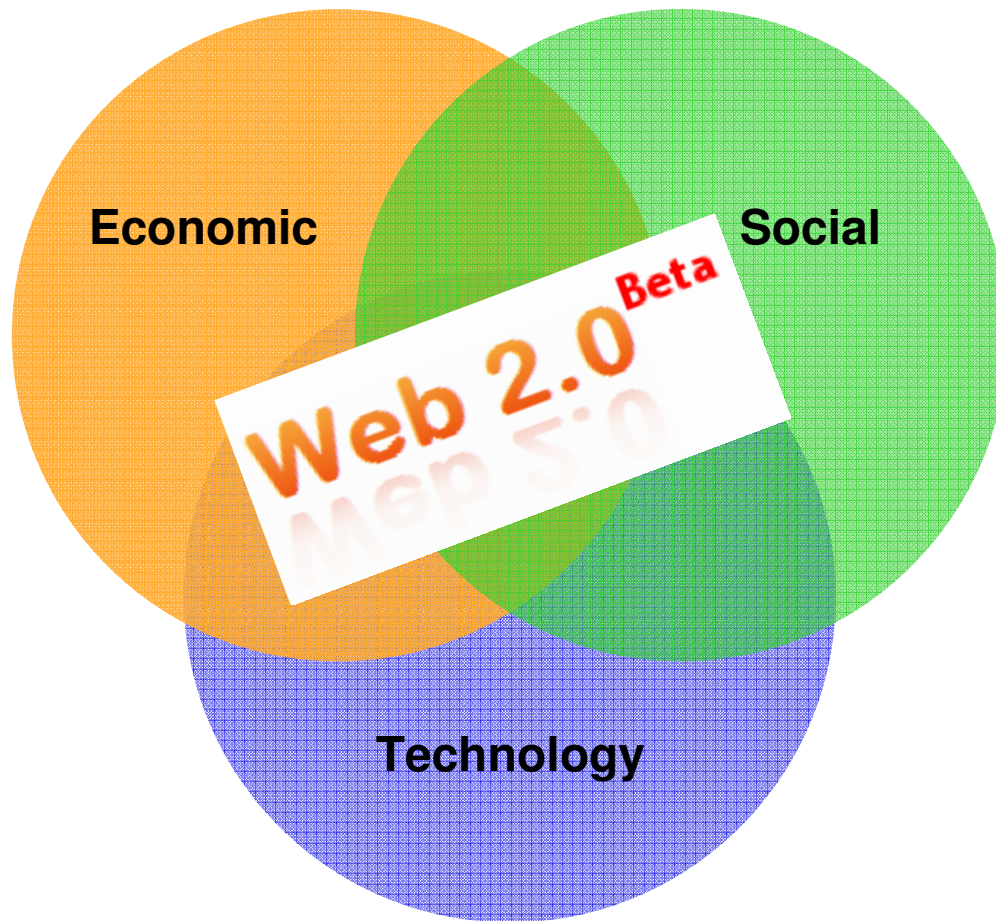
IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.

IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.

NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:

- CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR
- ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.

# Web 2.0 is why the internet is exciting today ... ... for everyone



Web 2.0 is a set of economic, social, and technology trends that collectively form the basis for the next generation of the Internet - a more mature, distinctive medium characterized by user participation, openness, and network effects.

Source: Web 2.0 Best Practices and Principles, O'Reilly Radar

Web 2.0 is a term that captures the widespread sense that there's something qualitatively different about today's Web.

Source: Tim O'Reilly

## Examples of technologies relevant to CICS

- **Clean and meaningful URLs**
- **RESTful access to CICS resources**
- **Access via AJAX**
- **Syndication and aggregation of data in RSS/Atom**
- **Mashups**
- **CICS is not likely to be directly concerned with the “social networking” aspects of Web2.0**

## URLs, RESTful and tagging

- **“Clean and meaningful URLs”**

- For everything!
- e.g. a file, a Database, a TQueue....
  - or a single record within each resource

- **REST**

- REpresentational State Transfer - See [http://en.wikipedia.org/wiki/Representational\\_State\\_Transfer](http://en.wikipedia.org/wiki/Representational_State_Transfer)
- How the Web has always worked...
  - HTTP POST | GET | PUT | DELETE are all the verbs you'll ever need
  - Everything else is a *resource* – with a representation of its state

- **Tagging**

- “Remembering in public”
- Associating audience-defined terms to things so that dynamic collections are formed

## REST and SOAP comparison

- **REST over HTTP:**
  - All objects/resources have a URI (URL)
  - Operations on the objects are done through the four defined HTTP methods:
    - POST, GET, PUT, DELETE
- **SOAP over HTTP:**
  - Objects are imbedded in the SOAP message
  - Operations have a URI
    - (Or a verb within a namespace with a URI)
  - Only uses the HTTP POST method

## Access via AJAX, what's that?

- **Aynchronous Javascript and XML**
- **Mainly a browser/UI technology**
  - Javascript spruced up in some new clothes
  - Allows a client-side script to interact directly with a server
- **Provides a more responsive UI for web apps**
  - More like a locally installed application
- **Asynchronous is the key distinguishing feature**
  - HTTP requests sent to the servers asynchronously – not in direct response to normal web page requests
  - Few (if any) whole-page requests
  - **LOTS** of smaller requests more often

## Syndication: RSS and Atom


- **Syndication: publishing short items of text**
  - News headlines
  - System status changes
  - New customer records
  - . . .
- **Client can be a web browser or a dedicated “feed aggregator”**
- **Two main syndication flavors**
  - RSS is very popular but has several ad hoc implementations
  - Atom is an emerging standard defined in two RFCs



## WEB 2.0 and ATOM

- **ATOM feeds can be basis for other Web 2.0 technology**
  - Mashups, feed aggregators, blogs all support ATOM
- **Comes in two levels of complexity:**
  - RFC4287 - Atom Syndication Format (2005)
    - Feeds and Entries
    - Targetted at producing feeds
  - RFC5023 - Atom Publishing Protocol (2007)
    - Services and Collections
    - Targetted at creating and updating resources

## WEB 2.0 summary

- **Web 2.0 is mainly about more dynamic forms of web interaction - “Rich Internet Applications”**
  - Mashups
    - Content from multiple servers composed in a single web page
    - Uses AJAX (a.k.a. **XMLHttpRequest** from JavaScript)
    - Archetypal example is GoogleMaps
  -  Syndications
    - Menu of short messages, possibly linking to other content
    - Podcasts
  - Blogs
    - Online “web log”, diary, etc, possibly collaborative

# Web 2.0 Support in CICS

## Why Web2.0 in CICS, and why now?

- **Web 2.0 is clearly an important new style of accessing data**
- **Potential to “unlock” data currently stored in CICS**
- **We want to show that CICS has an interest in the technology**
- **Gauge customer interest by shipping a SupportPac™**
  - Demonstrate feasibility
  - Test the market
  - Gather user feedback

## SupportPac CA8K characteristics

- **Targeted at delivering Atom feeds**
- **Runs on CICS TS 3.1 and 3.2**
- **Available now (released on 20 Mar 2008)**
  - Download from:  
<http://www.ibm.com/cics/supportpacs>
- **Needs a prerequisite D-type APAR (PK58721)**

## CICS Atom SupportPac design objectives

- **Only a portion of the CICS resource needs to be exposed as an ATOM entry**
  - For example a single file record, or queue item, or a range of records or items
- **The selection of the resource portion is identified by the URL querystring:**

`http://winmvs2c.hursley.ibm.com/atom/cicsfile/filea  
?s=66666`

## CICS Atom SupportPac design objectives (cont)

- **RESTful API requires “CRUD” interfaces:**
  - CREATE, READ, UPDATE, DELETE
- **These are mapped from the four HTTP methods:**

<u>Operation</u>	<u>HTTP method</u>	<u>CICS API (example)</u>
<b>CREATE</b>	<b>POST</b>	<b>WRITE FILE</b>
<b>READ</b>	<b>GET</b>	<b>READ FILE</b>
<b>UPDATE</b>	<b>PUT</b>	<b>READ FILE UPDATE + REWRITE FILE</b>
<b>DELETE</b>	<b>DELETE</b>	<b>DELETE FILE</b>

## CICS Atom SupportPac design objectives (cont)

- **READ for UPDATE cannot hold a lock over an HTTP GET/PUT sequence, so a “conditional PUT” has been implemented:**
- **GET returns an HTTP `Etag` header**
  - Etag value uniquely identifies the response instance returned
  - can be a timestamp, sequence number, or checksum value
- **PUT request sends the same Etag value in an `If-Match` header**
  - If resource has changed, HTTP response 412 (Precondition failed) is returned, and update is not performed



## Mapping XML to CICS resource binary data

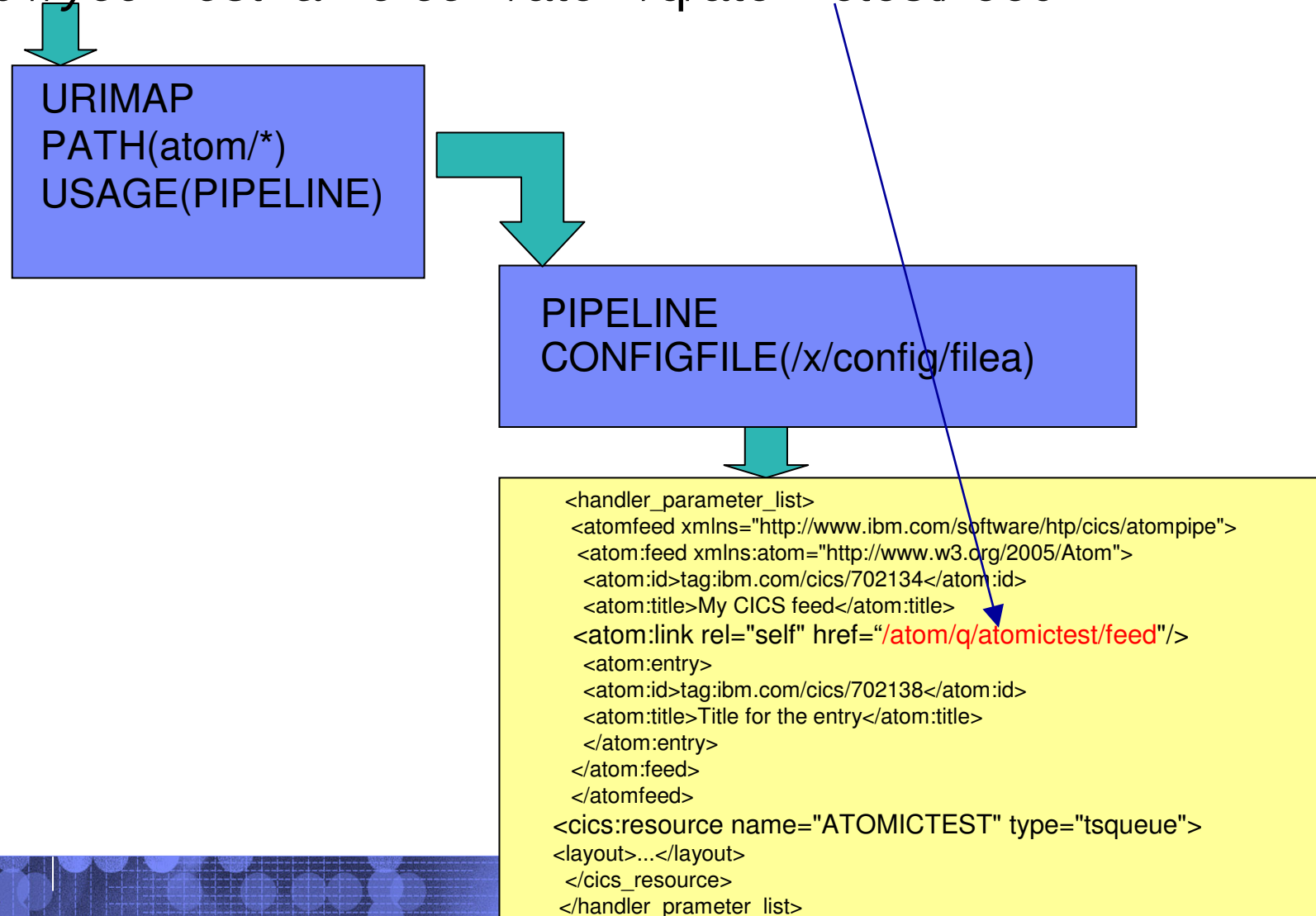
- **Data from Atom is character-based data imbedded within XML**
- **Data in the CICS resource is in binary or packed-decimal form**
- **Bi-directional conversion is needed**
- **This is similar to the problem that is already addressed in existing CICS Web Services support**
  - A similar approach is required for Atom feeds

## Design for SupportPac CA8K

- **Design uses *existing* RDO resource of PIPELINE**
- **Demonstrates that Web 2.0 can be done using *existing* CICS Web Support**
- **Exploits the z/OS XML System Services**
  - Namespace-aware XML parser
  - Available on z/OS 1.7 (with UA31443) and later
  - No open source licensing issues

## Configuration for SupportPac CA8K

<http://yourhostname.com/atom/q/atomicctest/feed>



## CA8K sample pipeline config data

```
<?xml version="1.0" encoding="UTF-8"?>
<provider_pipeline
  xmlns="http://www.ibm.com/software/htp/cics/pipeline"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ibm.com/software/htp/cics/pipeline/provider.xsd">
  <service>
    <terminal_handler>
      <handler>
        <program>DFH$W2FD</program>
        <handler_parameter_list>
          <cics:feed xmlns:cics="http://www.ibm.com/software/htp/cics/atompipes"
            <!-- CICS Atom feed metadata .... -->
          </cics:feed>
        </handler_parameter_list>
      </handler>
    </terminal_handler>
  </service>
</provider_pipeline>
```

## Describing CICS feed data

```
<cics:feed xmlns:cics="http://www.ibm.com/software/htp/cics/atompipeline">
  <atom:feed> <!-- Metadata for Atom feed --> </atom:feed>
  <cics:resource name="ATOMICTEST" type="tsqueue">
    <cics:fieldnames id="atomid" updated="datetime"
      title="atomtitle" subtitle="atomstitle"
      summary="atomsumm" />
    <cics:layout>
      <!-- description of resource record layout -->
    </cics:layout>
  </cics:resource>
</cics:feed>
```

## Mapping XML to CICS resource binary data

- **The SupportPac does not exploit existing CICS XML conversion services**
  - Too heavily SOAP oriented
- **Instead, the CICS resource layout is described by a DFDL specification within the pipeline configuration file**
- **DFDL (Data Format Description Language)**
  - Superset of XML schema language
  - Being specified by Open Grid Forum working group
  - See <http://forge.gridforum.org/projects/dfd1-wg/>

## Example of DFDL specification

```
<xs:element name="sbyte" type="xs:byte"
 />

<xs:element name="comment" type="xs:string"
 dfdl:length="16" />

<xs:element name="packnum" type="xs:decimal"
 dfdl:length="16"
 xs:fractionDigits="2"
 dfdl:decimalFormat="packed"
 dfdl:decimalSigned="false"
 />
```

## Adding a customized Service routine

- **If the configuration file specifies a CICS resource type of PROGRAM, the SupportPac will LINK to a program you specify (called a Service routine)**
  - Information about the Atom feed request will be passed in an ATOMPARAMETERS container
    - Including record layout information from the configuration file
  - If the request is a POST or a PUT, update information is in the DFHREQUEST container
  - Your customized application must return feed content in the ATOMCONTENT container
- **There is a sample Service routine in the SupportPac**
  - DFH0W2FA accesses the sample file FILEA



## CA8K has prerequisite D-type APAR

- **Current PIPELINE support is SOAP oriented**
  - HTTP methods GET and DELETE cause errors
    - Pipeline expects Content-Type header
  - MEDIATYPE of HTTP response is assumed to be SOAP
  - HTTP status code cannot be set within the handler application
  - Also need to recognize JavaScript and JSON mediatypes as text
    - for correct codepage conversion
  - Fixes for above are available in CICS APAR **PK58721**

## Pipeline changes for PK58721

- **Two new control containers can be specified, which will be processed by the non-SOAP pipeline HTTP support**
- **DFHHTTPSTATUS**
  - used to specify the HTTP status code in the final HTTP message.
  - Format is: “HTTP/1.1 *nnn* *xxxxxxxx*”
    - where *nnn* is the status code and *xxxxxxxx* is the status text
    - For example: HTTP/1.1 412 Precondition Failed
- **DFHMEDIATYPE**
  - used to specify the media type of the data returned in the DFHRESPONSE container
  - Format is “*type/subtype*”
    - where *type* and *subtype* are standard media definitions
    - For example: application/atom+xml

## Summary

- **CICS is interested in moving into Web 2.0 technology**
- **SupportPac CA8K shows how this can be done today**
  - Can deliver Atom feeds containing CICS data
  - Can receive requests from Ajax clients
  - Can access CICS data using REST interfaces
  - Can potentially deliver CICS data into mashups

# Thank You for Joining Us today!

**Go to [www.ibm.com/software/systemz](http://www.ibm.com/software/systemz) to:**

- ▶ Replay this teleconference
- ▶ Replay previously broadcast teleconferences
- ▶ Register for upcoming events