DataPower in a Web 2.0 World

Brien Muschett, Lead DataPower Web 2.0 Engineer
Robert Peterson, DataPower Specialist, Master Inventor
Adolfo Rodriguez, STSM, WebSphere DataPower Architect


MAY 2-7 Las Vegas, NV
Agenda

• Understanding DataPower
• Extending DataPower value into Web 2.0
• DataPower use cases
  • REST proxy
  • REST protocol bridging
  • AJAX/JSON bridging
  • RSS/ATOM feed aggregation
• Recent DataPower enhancements
• Summary
DataPower SOA Appliances Product Family

Low Latency Appliance XM70
- High volume, low latency messaging
- Enhanced QoS and performance
- Simplified, configuration-driven approach to LLM
- Publish/subscribe messaging
- High Availability

Integration Appliance XI50
- Hardware ESB
- “Any-to-Any” Conversion at wire-speed with WS-TX
- Bridges multiple protocols
- Integrated message-level security

B2B Appliance XB60
- B2B Messaging (AS2/AS3)
- Trading Partner Profile Management
- B2B Transaction Viewer
- Unparalleled performance
- Simplified management and config

XML Security Gateway XS40
- Enhanced Security Capabilities
- Centralized Policy Enforcement
- Fine-grained authorization
- Rich authentication

Impact 2010: The Premier Conference for Business and IT Leaders
Typical DataPower Use Cases

Monitoring and control
   Example: centralized ingress management for all Web Services using ITCAM SOA

Deep-content routing and data aggregation
   Example: XPath (content) routing on Web Service parameters

Functional acceleration
   Example: XSLT, WS Security

Application-layer security and threat protection
   Example: XML Denial-of-Service protection, WS Security

Protocol and message bridging
   Example: Convert to WS to legacy Cobol/MQ
Why an Appliance for SOA?

- Integrated
  - Many functions integrated into a single device
  - Addresses the divergent needs of different groups (architects, operators, developers)
  - Integrates well with other IBM SWG and standards-based products
- Hardware reliability
  - Dual power supplies, no spinning media, self-healing capability, failover support
- Security
  - Higher levels of security assurance certifications require hardware (HSM, government criteria)
  - Inline application-aware security filtering and intrusion protection
- Higher performance with hardware acceleration
  - Wire-speed application-aware parsing and processing
  - Ability to perform costly XML security operations without slow downs
- Consumability
  - Simplified deployment and management: up in minutes, not hours
  - Reduces need for in-house SOA skills & accelerates time to SOA benefits
The DataPower Secret Sauce

Specialized compiler technology creates optimized executable object code from transformations (e.g., XSLT) that execute natively on hardware.

Everything is viewed as a transformation that is extensible via DataPower custom extension functions.

High-performance throughput-optimized engine yields wire-speed capabilities.

Purpose-built hardware to execute SOA workloads and transformations.
Why Web 2.0?

• Web 2.0 represents the next generation of web applications which have several characteristics:
  • **Rich and Immersive** – replace traditional desktop applications
  • **Constantly Changing** – new versions can be rolled out seamlessly and the browser is a ubiquitous platform
  • **Collaborative** – utilize contacts, the “wisdom of crowds,” and allow for real-time interaction
  • **Integrated** – aggregate data from other sources, mashup content, and the like

• *All* of these concepts apply to business applications

• IBM is committed to Web 2.0 and considers it a complimentary approach to SOA and our ongoing strategy to make modern businesses successful in the marketplace
Reconciling Web 2.0 Architecture and SOA

- Proxy
- Bridging
- Security
Web 2.0 and DataPower

• DataPower 3.8.0 provides a comprehensive set of Web 2.0 enhancements at the forefront

• DataPower now enables and compliments a Web 2.0 architecture in several ways:
  • First-class support for Web 2.0 technologies like AJAX and REST
  • Modernization of legacy applications to be accessed by Web 2.0 clients/consumers
  • Hardended security, performance, and high availability for Web 2.0 applications
Web 2.0 and SOA Deployment Trends
DataPower Web 2.0 Use Cases
**REST Proxy**

- Centralized monitoring, and management point for all traffic
  - Out-of-the-box service-level monitoring and throttling
  - Centralized logging and auditing

- Enforcement point for centralized security policies
  - HTTP basic auth or SSL requirements
  - AAA and DoS security
  - Message filtering
Simplified REST-to-SOA Bridging

PUT /library/part/1234 HTTP/1.1
Host: acmeparts.com
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
Content-Type: text/xml
Content-Length: 12
<qoh>32</qoh>

Web Service (SOAP) Provider

SOAP
GetHandle
UpdateQOH
ShutDown

Hide underlying “conversational” exchange from the REST interface
AJAX Bridging

Perform format translation from JSON to SOAP (and vice versa)

...{"Task": "Dry cleaning: shirt, pants, and 20% discount coupon"}...
News Feed Aggregation

- Composition of new services using integration (a.k.a., “mash-up”) technology to “wire” together REST services
- A solution may be rapidly assembled by combining existing feeds and services that aggregate, enrich, sort, and filter data in a pipeline
- Asynchronous support enables parallelized networked overhead:

![Graph showing latency vs. number of actions for asynchronous and synchronous scenarios.](image-url)
DataPower Web 2.0 Support
Manipulating URIs

Key pattern element – URIs

RESTful URIs are verbose while SOAPful URIs are terse

Example:

RESTful: http://[host]/movie/IndianaJones/RaidersOfTheLostArk
SOAPful: http://[host]/soap/servlet/messagerouter

Solution: Use a URL rewrite policy to bridge URI differences.
Manipulating SOAP Headers

RESTful services use a variety of HTTP headers while SOAPful services use mainly SOAPAction

Example:

RESTful: not applications

SOAPful: “SOAPAction : GETLastTradePrice”

Solution: Use the Set Variable action to set the SOAPAction header
Enabling RESTful responses

RESTful responses include headers and response codes

Example:

RESTful: DELETE http://[host]/movies/StarWars returns 201

RESTful: POST http://[host]/movies/StarWars returns Location Header

Solution: Match SOAP responses using XPath expressions to tie front side request to back side response then use style sheets to set Location header, response codes and so forth, and to translate payloads.
DataPower Web 2.0 Functional Enhancements
Recent DataPower Web 2.0 Improvements

- RESTful Message Processing
  - Ability to handle differing payload requirements within a single policy definition
  - Bypass “One way exchange pattern”

- RESTful HTTP Method Enhancements
  - Better accessibility to the HTTP verbs

- RESTful Bridging / Proxy support
  - Ability to rewrite HTTP Request line (method (new) + uri)

- JSON and JSONX
Empty Body Requests

RESTful message requests might not have a message body

Example:

RESTful: DELETE http://[host]/movies/Fridaythe13thPartXXIIXX
Still need processing rules to execute to bridge requests

Solution: Use the Advanced Configuration Option “Process Messages Whose Body is Empty”
Process Empty-Body Messages

- New MPGW/XMLFW configuration option
- Useful for RESTful message patterns empty messages are common
- Bypasses the built in “One Way Exchange Pattern” in multistep
- Request/response types are XML and JSON
Matching on HTTP Method

RESTful URIs are overloaded – the same URI supports multiple verbs

Example:

RESTful: http://[host]/movies/StarWars

HTTP GET returns the movie details i.e., its representational state
HTTP DELETE removes the movie details (and subsequently the movie)
HTTP PUT updates the movie details i.e., its representational state
HTTP HEAD will provide the movie meta data

Solution: Use the HTTP Method type for the match action in the matching rule
HTTP Method Match on Match Action

- New matching HTTP Method rule type
- Supports HTTP HEAD, DELETE, PUT, POST, GET
- Can be combined with other match criteria e.g., URL, Xpath
HTTP Method on dp:url-open

- New attribute on dp:url-open
  - http-method
  - Supports HTTP HEAD, DELETE, PUT, POST, GET
HTTP Method/Body on Fetch Action

- Fetch Action includes HTTP Method
- Content (Input Context)
- Supports HTTP HEAD, DELETE, PUT, POST, GET
- Content can be used to set headers
HTTP Method on Results Asynchronous Action

- Results Async includes HTTP Method
- Supports HTTP DELETE, PUT, POST
HTTP Method on Results Action

- Results Action includes HTTP Method
- Supports HTTP DELETE, PUT, POST
- Method applies to all result targets
HTTP Method on Log Action

- Log Action includes HTTP Method
- Supports HTTP DELETE, PUT, POST
Dealing With HTTP Verbs

RESTful services use a variety of HTTP verbs while SOAPful services use mainly POST.

Example:

RESTful: GET http://[host]/movies/StarWars

SOAPful: POST http://[host]/soap/servlet/messagerouter

Solution: Use the advanced Method rewrite action to bridge Method differences.
Method Rewrite Advanced Processing Action

- New Method Rewrite Action
- Rewrites HTTP method request to backend
- Useful for bridging e.g., RESTful to SOAPful
- Supports HTTP HEAD, DELETE, PUT, POST, GET
- Advanced action is shorthand for setting service variable
Method Rewrite using Set Variable action

- Updated service variable
  - var://service/protocol-method
Converting JSON to XML

RESTful services can use JSON whereas SOAPful services use XML

Example:

RESTful: [“Milk”, “Eggs”, “Tuna”, “Beer”]

SOAP:

```xml
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <l:setList xmlns:l="http://groceries.com">
      <list-id>Milk</list-id>
      <list-id>Eggs</list-id>
      <list-id>Tuna</list-id>
      <list-id>Beer</list-id>
    </l:setList>
  </soap:Body>
</soap:Envelope>
```

Solution: Invoke JSON as a native data type or use the HTTP Convert action with a default JSON encoding to convert to JSONX.
JSON as a Native Data Type

- Parses and validates the incoming JSON against the RFC 4627
  - Result of this validation is the content in the INPUT context
- Generates an ancillary context called __JSONASJSONX for further mediation
Processing JSON payloads

- New HTTP Input encoding
- JSON encoding
- Converts JSON to JSONX
- Specified using the Convert HTTP action as default encoding
- JSONX to JSON stylesheet provided
- JSONX XML schema provided
What is JSONX?

• IBM Internal Standardization of JSON modeled in XML
  – Spec was developed by DP with input from the Data Web Service Team (DB2)
  – Strict model of RFC 4627 – application/json
  – Productized by DP and Data Web Service Team

• Developed to be generically schema validateable (json is validated)
  – Not an arbitrary representation of JSON data as XML
  – Not at attempt to model any XML as JSON

• Developed to be a non-lossy transformation of JSON types/data

• Useful for everything DataPower
  – RESTful json bridge to SOAP
  – Threat protection for Ajax clients that use eval(json)

• Data Web Service details:
  https://w3.tap.ibm.com/w3ki07/display/pureXML/jsonxArticles
A simple JSON object with two properties

```json
{ "First" : "John",
  "Last" : "Wayne" }
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<json:object xsi:schemaLocation="http://www.datapower.com/schemas/json json.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <json:string name="First">John</json:string>
  <json:string name="Last">Wayne</json:string>
</json:object>
```
We Value Your Feedback!

- Please complete the session survey for this session by:
  - Accessing the SmartSite on your smart phone or computer at: http://imp2010.confnav.com
    - Surveys / My Session Evaluations
  - Visiting any onsite event kiosk
    - Surveys / My Session Evaluations

- Each completed survey increases your chance to win an Apple iPod Touch with daily drawing sponsored by Alliance Tech
Copyright and Trademarks

© IBM Corporation 2009. All rights reserved. IBM, the IBM logo, ibm.com and the globe design are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml. Other company, product, or service names may be trademarks or service marks of others.