



IBM Software Group

TPF Users Group Spring 2005

TPF Web Services Support

Jason Keenaghan
AAA Task Force

AIM Enterprise Platform Software

IBM z/Transaction Processing Facility Enterprise Edition 1.1.0

© IBM Corporation 2005

Any references to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

Purpose for Presentation

- Provide an update on IBM's current undertakings for Web Services support
 - Enhancements to current components
 - Introduction of new components
- Receive feedback on current direction and implementation plans
 - Evaluate the relevancy of the proposed work items
 - Identify important pieces that are missing
- Ensure the correct prioritization of new Web Services work items
 - Encourage the creation of new formal requirements
 - Align IBM's delivery of Web services enhancements with user needs

What is IBM working on?

■ WS-I profile conformance

– Web Services Interoperability Organization (WS-I)

- An open industry organization chartered to promote Web services interoperability across platforms, operating systems, and programming languages
- <http://www.ws-i.org>

– Allow TPF to better interoperate with a larger array of products and platforms through web services

- This is achieved by conforming to the profiles and by making use of the testing tools that the WS-I provides
- Conformance level will be selected on a per service basis

■ Make web services deployment/development comparable to other platforms

– This is accomplished with a combination of online additions and offline tooling

- Without this, users would be required to learn, and stay current with, the details of each of the standards that make up Web services
- As these standards evolve, we will probably need to make some online additions/changes, but a considerable amount of effort will be saved if TPF can rely on common offline tooling as is done on other platforms

– Coexist with TPF's current SOAP server support

WS-I Conformance

- Over 200 interoperability issues associated with implementing the core Web services specifications
 - Example:
 - SOAP allows for the creation of custom fault codes using the “dot” notation
 - Client.Authentication where Client is a fault value defined in SOAP1.1, but Authentication is a further clarification of the fault but is not standard
 - Interoperability issues can arise when the same non standard names are used to convey different meaning by different implementations
 - To prevent these issues, the Basic Profile specifies that only the SOAP1.1 defined fault values are valid and describes a mechanism to provide further fault clarification without introducing interoperability issues
- Conform to WS-I Basic Profile 1.1 & Simple SOAP Binding Profile 1.0
 - Can claim conformance on “artifacts” and “parties”
 - Artifacts – SOAP messages, WSDL Descriptions, UDDI registry data, ...
 - Updates to TPF SOAP server
 - Creation of offline tooling
 - Parties – SOAP Processor (server), SOAP Client, UDDI registry, ...
 - Updates to TPF SOAP server

Tasks to Deploy TPF Application as a Web Service

- Create WSDL document that describes the interface to the TPF application
 - Today: Done with any text editor or tool that TPF users can find
 - Tomorrow:
 - **Minimum:** Provide a WSDL editor/tool plug-in to TPF Toolkit to validate, and check for WS-I conformance, of a WSDL document
 - **Ideal:** Include a WSDL generator that starts with C/C++ interface
- Create a driver/wrapper to route an incoming request to the actual TPF application
 - Today: Done by updating `tpf_soap_appl_handler()`
 - Tomorrow:
 - **Minimum:** Provide a mechanism to deploy new applications without having to hard code `tpf_soap_appl_handler()`. Additionally, provide a SOAP bridge for accessing applications that expect input messages in AMOSG-format.
 - **Ideal:** Include tooling that generates TPF application wrappers and deployment descriptors that perform this routing

Tasks to Deploy TPF Application as a Web Service (continued)

■ Extract input parameters from SOAP message

- Today: Done by updating TPF application to extract the parameters from the infoNodes structure created by the B2B scanner
- Tomorrow:
 - **Minimum:** Provide a high-level API, to extract input parameters, to be used by TPF applications
 - **Ideal:** Include tooling that generates TPF application wrappers that extract parameters
 - Examples of APIs:

```
getElementsByTagName  
getFirstElementByTagName  
getNextElementByTagName  
getFirstChild  
getNextChild  
positionToElement
```

APIs for Accessing Parsed XML Document – Using Tag Name

getElementsByTagName	Returns a pointer to an array that contains the element tag name and element content value in the requested format for the requested element tag name. The special value of '*' matches all tags.
getFirstElementByTagName	Returns a pointer to an array that contains the element tag name and element content value in the requested format for the first element matching the requested element tag name.
getNextElementByTagName	Returns a pointer to an array that contains the element tag name and element content value in the requested format for the next element matching the requested element tag name

APIs for Accessing Parsed XML Document – Using Relative Position

getFirstChild	Returns a pointer to an array that contains the element tag name and element content value in the requested format for the first child element
getNextChild	Returns a pointer to an array that contains the element tag name and element content value in the requested format for the next child element
positionToElement	Positions the XML data structure pointer to the elementNode of the position (index) that is requested.

Example of Using XML APIs

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!--Sample SOAP message -->
<SOAP-ENV:Envelope xmlns:SOAP-
  ENV="http://www.w3.org/2002/06/soap-envelope">
<SOAP-ENV:Body>
<ns1:addPassengerNameRecord
  xmlns:ns1="http://com.ibm.pnr.PNRService">
  <m:Passengers xmlns:m="some-URI">
    <m:PassengerName title="Mr.">
      <m:FirstName>John</m:FirstName>
      <m:MiddleName>Lawrence</m:MiddleName>
      <m:LastName>Sayers</m:LastName>
    </m:PassengerName>
    <m:PassengerName title="Mrs.">
      <m:FirstName>Karen</m:FirstName>
      <m:MiddleName>A</m:MiddleName>
      <m:LastName>Sayers</m:LastName>
    </m:PassengerName>
  </m:Passengers>
  <PhoneNumber countryCode="44" areaCode="340"
  pNumber="6353343" />
  <o:FormOfPayment xmlns:o="http://com.ibm.financial">
    <o:cclssuer>MasterCard</o:cclssuer>
    <o:ccNumber>1234567899999999</o:ccNumber>
    <o:ccExpiration>05/03</o:ccExpiration>
  </o:FormOfPayment>
</ns1:addPassengerNameRecord>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- Example: `getElementsByTagName`
 - Application passes in a data type and element tag name: `"STRING_FORMAT"."PassengerName"`
 - Traverse the XML data structure for all elements with a tag name of `PassengerName`. Returns array with pointers containing the element tag name and content values as strings.

"title"	"Mr."
"FirstName"	"John"
"MiddleName"	"Lawrence"
"LastName"	"Sayers"
"title"	"Mrs."
"FirstName"	"Karen"
"MiddleName"	"A"
"LastName"	"Sayers"

Tasks to Deploy TPF Application as a Web Service (continued)

■ Build SOAP response message

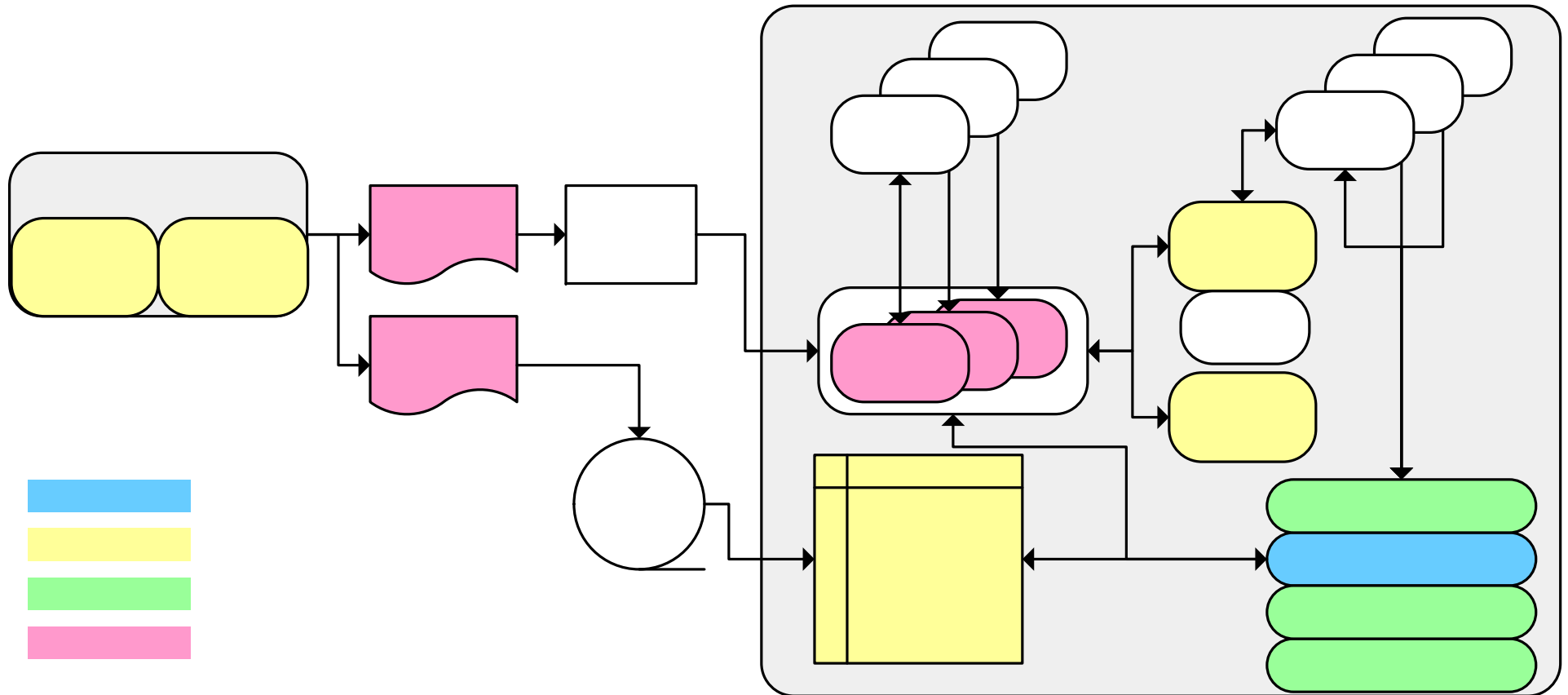
- Today: Done by updating TPF application to create a SOAP response in XML format that conforms to the WSDL document for the service, if one exists, and return this to the SOAP server
- Tomorrow:
 - **Minimum:** Provide a high-level API, to generate XML, to be used by TPF applications
 - **Ideal:** Include tooling that generates TPF application wrappers that generate XML
 - Examples of APIs:

```
createDocument  
createElement  
insertElement  
deleteElement  
buildXMLdocument
```

APIs for Creating XML Document

createDocument	Creates a response xml data structure starting with the documentNode. Creates all associated nodes for the documentNode that are requested (eg. namespaceNode, attributeNode(s), contentNode(s)).
createElement	Creates an elementNode and all associated nodes that are requested (eg. namespaceNode, attributeNode(s), contentNode(s)).
insertElement	Finds the requested existing element in the xml data structure, creates and inserts the new elementNode AFTER the existing element. Creates all associated nodes for the new element (eg. namespaceNode, attributeNode(s), contentNode(s)).
deleteElement	Deletes the elementNode and all associated nodes for the deleted element (eg. namespaceNode, attributeNode(s), contentNode(s)).
buildXMLdocument	Traverses the XML data structure and builds an XML document. Returns a pointer to the XML document

Future TPF Web Services Support



Trademarks

IBM is a trademark of International Business Machines Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Notes

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.