

# OCEAN

June 21, 2011

## XML and Web Services

Presented by:  
Sharon L. Hoffman  
shoffman@techreflections.com

1

## Agenda

- **Web services definition and concepts**
- **XML overview**
- **Web services standards**
- **Invoking web services**
- **Hosting web services**

2

## **Web Services Definition and Concepts**

3

## **Web Services Defined**

- **Conceptually a web service is an application that can be invoked without any knowledge of how the application works**
- **A program-to-program (versus user-to-program) communication**
- **Web services are agnostic concerning platform, database, and programming language**
- **Communicating/understanding the web service requirements is an important task**
  - **Expect to invest some time working with your counterparts at your business partners**

4

## Typical Web Service Characteristics

- **There are lots of exceptions, but most often web services ...**
- **Are business-to-business communications**
- **Deliver information via the Internet**
- **Utilize synchronous round-trip communications**
- **Use XML for data transfer**

5

## Web Service Roles

- **Invoking a web service**
  - **Use a web service provided by somebody else**
- **Hosting a web service**
  - **Make a web service available to others**
- **Both hosting and invoking a web service require**
  - **Creating XML documents**
  - **Extracting data from XML documents**
  - **Communicating via HTTP**
- **The programming to use a web service is different from the programming to provide a web service**

6

## Invoking a Web Service

- **Send a request for the specific operation to the web service URL**
- **Process the web service response**

7

## Hosting a Web Service

- **Make business logic available via a neutral interface**
- **Accept a request for a one or more operations (methods) at a particular URL**
- **Process the request**
- **Return a response if required**
  - **At a minimum the request should be acknowledged**
- **Provide specifications for using the web service**

8

## Working with Web Services

- Traditional i5/OS languages currently have no direct support for web services
- Languages such as PHP and .Net languages (C# and VB) have built-in web service capabilities
- i5/OS ILE languages include some basic XML capabilities
- V7R1 SQL has additional XML support
- Tools are available for generating web services and web service requests

9

## Getting Started with Web Services

- Learn basic XML syntax
- Get specifications for the web service you want to invoke
- Plan for error-handling and reporting

10

## XML Overview

11

## Basics of XML

- **eXtensible Markup Language**
- **A tag-based language that can be used to describe any type of information**
- **Platform, database, and programming-language neutral**

12

## Essential XML Structures

- An XML document is self-contained and typically represents a single transaction
- An XML element defines a single piece of information – similar to field/column in a database
- XML elements can be nested multiple levels
- An XML attribute specifies information about an element
- It's up to the XML designer to determine whether the data resides in elements, attributes, or a combination of both

13

## XML Architecture

- **XML document**
  - Contains both tags (data description) and data
  - Usually a small number of transactions or a single transaction
- **Schema (xsd) or Document Type Definition (DTD)**
  - Use to communicate XML structure for a particular type (e.g., purchase order) of XML document
  - Use for basic validation of XML documents against a defined structure
  - Additional validation is usually required

14

## XML Elements

- **Elements**
  - a pair of tags and its data  
`<name>Sharon L. Hoffman</name>`
- **Sub elements**
  - break elements into parts  
`<name>  
  <first_name>Sharon</first_name>  
  <middle_initial>L</middle_initial>  
  <last_name>Hoffman</last_name>  
</name>`
- **Root element**
  - one per XML document

15

## XML Attributes

- Name/value pairs within the opening tag of an element
- Quotes (single or double) are required for values
- Must be unique within an element
- Attribute example:  
`<purchase_order po_number = '12346'  
  customer_name = 'Acme Company'>`

16

## General XML Syntax

- **Element tag names**
  - case-sensitive
  - no spaces
  - start with a letter or underscore but cannot start with xml
  - after the first character, names can also contain numbers, hyphens, or periods
- **Use entity references for reserved characters within data**
  - &gt; &lt; &quot; &apos; &amp;
- **Elements must be properly nested**
- **An XML document that obeys syntax rules is “well-formed”**

17

## Additional XML Document Components

- **Empty elements**
  - contain only one tag
  - often used for elements that are composed exclusively of attributes
  - example: `<region assignment="Northeast"/>`
- **Comments**
  - same format as HTML comments
  - `<!-- this is a comment -->`
- **CDATA**
  - everything is treated as character data
  - `<![CDATA[Treat this information as text ]>`

18

## Sample XML

```
<?xml version="1.0" encoding="UTF-8"?>
<purchase_order po_number='12346' customer_name='Acme Company'>
  <po_item_list>
    <po_item>
      <item_number>8876-5</item_number>
      <item_quantity>10</item_quantity>
      <item_unit_price>12.36</item_unit_price>
    </po_item>
    <po_item>
      <item_number>2234-4</item_number>
      <item_quantity>5</item_quantity>
      <item_unit_price>10.00</item_unit_price>
    </po_item>
    <po_item>
      <item_number>7886-1</item_number>
      <item_quantity>22</item_quantity>
      <item_unit_price>5.35</item_unit_price>
    </po_item>
  </po_item_list>
</purchase_order>
```

19

## Sample DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT policies (policy+)>
<!ELEMENT policy ( policy_number,
  customer_name)>
<!ELEMENT customer_name (#PCDATA)>
<!ELEMENT policy_number (#PCDATA)>
<!ATTLIST policy
  region CDATA "national"
  type (auto|homeowner|life|health) #REQUIRED>
```

20

## Sample Schema (partial)

```
<complexType name="Address">
  <sequence>
    <element name="name" type="string" minOccurs="0"/>
    <element name="street" type="string"/>
    <element name="city" type="string"/>
  </sequence>
</complexType>
```

21

## Processing an XML Document

- Requires *parsing* -- extracting the data from the XML tags
- Two primary parser models
  - Document Object Model (DOM)
  - Simple API for XML (SAX)
- System i XML parsers are supplied in the no-charge XML Toolkit LPP
- Parsing a complex XML document manually is cumbersome

22

## **i5/OS Parsing Options**

- **Use a language that includes parsing support**
  - Java, PHP, C# etc.
  - V5R4 or higher RPG
  - V5R3 or higher COBOL
- **V7R1 SQL support for parsing XML**
- **XML-to-Database/Database-to-XML tools**
- **Use XML APIs to access the XML Toolkit parsers**
- **Use the stream file APIs and parse manually**
- **Convert the XML document stream file to a database file and parse manually**

23

## **i5/OS Options for Creating XML**

- **Use a language that includes XML support**
  - Java, PHP, C# etc.
  - V5R4 COBOL
- **V7R1 SQL support for building XML documents**
- **XML-to-Database/Database-to-XML tools**
- **Use APIs to write an XML document file from an RPG or COBOL program**
- **Create a database file with XML “text”**
  - Convert the database file to a stream file using CPYTOSTMF

24

## Web Services Standards

25

## Web Services Standards

- **WSDL – Web Services Description Language**
  - Standard for describing web services
  - Schemas and DTDS may be provided instead of or in addition to WSDL
- **SOAP – Simple Object Access Protocol**
  - Standard for web service request and response messages
  - The web service must support this capability
- **REST – Representational State Transfer**
  - An alternative to SOAP
  - The web service must support this capability
- **UDDI – Universal Discovery and Description Information**
  - Standard for web services directories
  - Not widely used

26

## WSDL

- **Web Services Description Language**
- **One alternative for documenting a web service**
- **WSDL documents are XML documents**
- **Specifies the operations the web service supports**
- **Specifies the format for incoming and outgoing messages for each method**
- **Specifies communications options such as the message protocol and security standard**
- **Specifies the web service URL**

27

## SOAP

- **Simple Object Access Protocol**
- **A widely-used format for web service messages**
- **A SOAP message is an XML document**
- **A SOAP message contains both standard elements and application-specific elements**
- **The format of SOAP messages used by a particular web service is usually specified using WSDL**

28

## Key SOAP Elements

- **<Envelope>**
  - The root element of a SOAP message
  - Specifies the name of the web service
  - Does NOT specify where the message is sent
- **<Body>**
  - Specifies the name of the operation
  - Contains operation-specific parameters

29

## REST

- **Uses HTTP (usually Get or Put) to send and receive web service messages**
  - Usually XML
  - May include other types of data
- **May be documented in the WSDL**
- **Many REST web services don't have WSDL**
  - Some other documentation must be provided

30

## Invoking Web Services

31

## Information Required to Invoke a Web Service

- **Determining the exact requirements to invoke a web service may be the most time-consuming task**
- **You'll need to know:**
  - The web service URL
  - The method (aka operation) you are requesting
  - The format of the request XML document
  - The format of the response XML document
- **The web service provider should supply all of this information**
  - May be a WSDL, a schema, a DTD, or less-formal documentation such as sample XML documents

32

## Invoking a Web Service

- **Create an XML request**
- **Send the request to a URL**
- **Process the XML response**
  - Check for errors
  - Report results
- **Reporting results**
  - Display on a web page
  - Send an email
  - Store selected data or an entire XML document

33

## Coding Options for Invoking Web Services

- **Use a language with built-in web-service support**
- **Several solutions are available that let RPG function as an HTTP client**
  - Scott Klement's HTTPAPI
  - Commercial products
  - These solutions also require you to build and parse XML

34

## Invoking REST Web Services

- **Build the XML document**
- **Create an HTTP client**
- **Send the XML to the specified URL with Get or Post**
- **Check for HTTP errors**
- **Retrieve the response**
- **Parse the response**

35

## Invoking a SOAP Web Service using PHP

- **Create a SoapClient object specifying the URL of the web service**
- **Invoke the specific web service method using PHP's object-oriented syntax**
- **Build the request parameters**
  - **Create an associative array for simple parameters or use objects to define request XML structures**
  - **Simple parameters do not require any XML coding – they will be properly nested when passed to the web service**
- **Parse the response to extract the required information**

36

## Invoking a SOAP Web Service using PHP

```
<?php
$webServiceClient = new SoapClient
("http://www.webservicex.net/CurrencyConvertor.as
mx?wsdl");
$parameters =
array('FromCurrency'=>"EUR",
      'ToCurrency'=>"USD");
$webServiceResponse =
$webServiceClient->ConversionRate($parameters);
print_r($webServiceResponse);
?>
```

37

## Parsing the Response using PHP

- The response is returned as an object or as a simple variable
- Depending on the complexity of the response you can parse it using different techniques
  - Drill down to the data you need by accessing object properties (variables)
  - Load part of the response into a SimpleXMLElement object
  - Load the entire response into a DOMDocument object
- **\_\_getLastResponse()** function of the SoapClient Object retrieves the response as XML
  - Very useful for debugging

38

## Hosting Web Services

39

## i5/OS Options for Hosting Web Services

- Write the web service using a language with built-in web-service support
- Register an i5/OS program as a stored procedure and invoke it using a language with web service support
- Generate a “wrapper” for an RPG (or COBOL) program

40

## Integrated Web Services for i5/OS (IWS)

- Free component of i5/OS beginning with V6R1 and via PTF for V5R4
- i5/OS tools for creating and testing web services
  - Also available in RDp SOA but not part of the base RDp product
- Generates Java components to invoke the underlying ILE RPG or COBOL for a web service
- Generates WSDL
- Includes testing tools
- Uses the lightweight web application server developed for DB2 Web Query

41

## IWS Web Services Wizard

- Builds a web service based on an ILE program using PCML
- Connect by entering:
  - `http://99.999.999.999:2001/HTTPAdmin`
  - Replace the 9's with your IP address
- You may need to start the HTTP server
  - `strtcpsvr server(*HTTP) httpsvr(*admin)`
- See the IWS web page for step-by-step instructions

42

IBM Web Administration for i

Setup **Manage** Advanced | Related Links

All Servers | HTTP Servers **Application Servers**

Running Server: WSERVICE - V1.3 (web services)

Common Tasks and Wizards

- Create Web Services Server
- Create HTTP Server
- Create Application Server
- Create WebSphere Portal

Web Services Wizards

- Deploy New Service

Server Properties

- Properties
- Server Tracing
- View HTTP Servers

Services

- Manage Deployed Services

Web Performance

- Web Performance Advisor

variety of platforms and programming languages such as C, C++, Java and .NET. An easy to use wizard is provided to configure the Web services server and the services for IBM i program objects. Other management functions such as starting, stopping and deleting services are also provided.

For more information, please visit: <http://www-03.ibm.com/systems/i/software/iws/>

Server: "WSERVICE"

- ConvertTemp
- TAXCALC05
- TAXCALCTest

Note: To update the status, click Refresh

43

IBM Web Administration for i

Setup **Manage** Advanced | Related Links

All Servers | HTTP Servers **Application Servers**

Running Server: WSERVICE - V1.3 (web services)

Specify IBM i library and ILE program object name (Recommended)

You can specify the program object location by entering the name of the library that contains the program object, as well as the name of the program object. This is the fastest and recommended way to locate the program object.

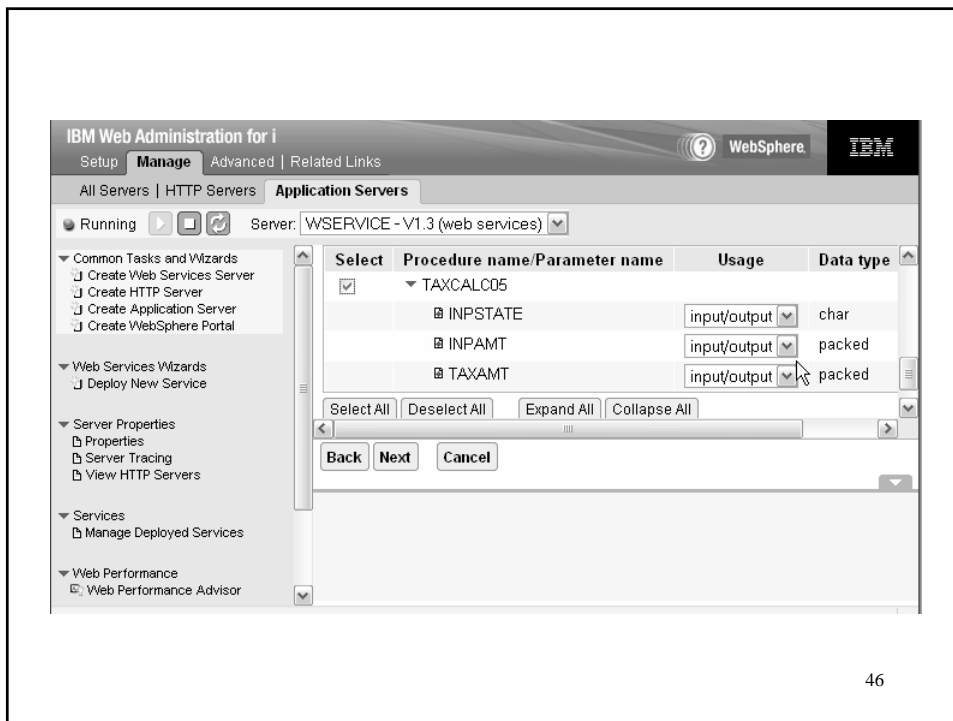
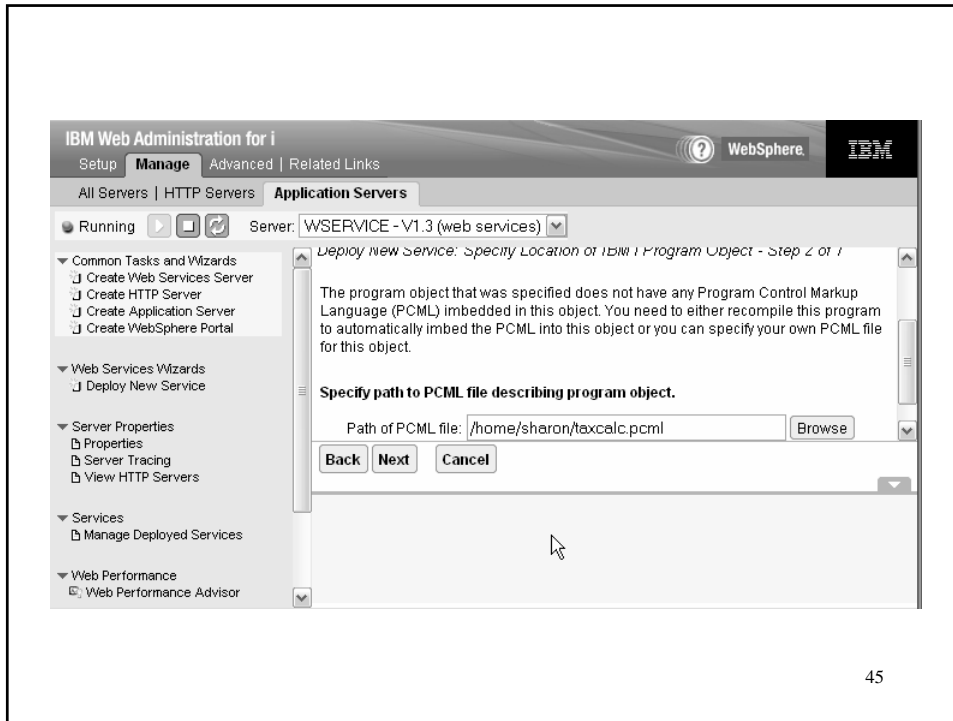
Library name: sharon1

ILE Object name: taxcalc05

ILE Object type:  \*SRVPGM  \*PGM

Back Next Cancel

44



The screenshot shows the 'Web Services Test Client' interface. On the left is a 'Navigator' pane with a tree view containing 'WSDL Main', 'http://62.176.136.111:10010/w...', 'TAXCALCTest03', and 'TAXCALCTest03SOAP11Bif'. The main area is titled 'WSDL Binding Details' and contains the following text: 'Shown below are the details for this SOAP <binding> element. Click on an operation to fill in its parameters and invoke it or specify additional endpoints.' Below this is an 'Operations' section with a table:

Name	Documentation
taxcalc05_XML	--
taxcalc05	--

Below the table is an 'Endpoints' section with a button 'Add Remove' and a list box containing one endpoint: 'http://62.176.136.111:10010/web/services/TAXCALCTest03?wsdl'. At the bottom, a 'Status' pane shows the message: 'IWAB0381I http://62.176.136.111:10010/web/services/TAXCALCTest03?wsdl was successfully c'.

47

The screenshot shows the 'Invoke a WSDL Operation' dialog box. It has a title bar 'Actions' and a subtitle 'Invoke a WSDL Operation' with a 'Source' link. The main text says: 'Enter the parameters of this WSDL operation and click Go to invoke.' Below this is an 'Endpoints' section with a text box containing 'http://62.176.136.111:10010/web/services/TAXCALCTest03' and a dropdown arrow. Underneath is a section for the operation 'taxcalc05' with a checkbox and a parameter 'param0 nil?' with 'Add Remove' buttons. At the bottom are 'Go' and 'Reset' buttons.

48

The screenshot shows a web service client interface. At the top, there is a header bar with a dropdown menu set to 'param0 nil?' and links for 'Add' and 'Remove'. Below this is a 'Content' section with a tree view. The tree view has three main nodes: 'INPAMT nil? decimal Add Remove', 'INPSTATE nil? string Add Remove', and 'TAXAMT nil? decimal Add Remove'. Each node has a 'Values' sub-section. The 'INPAMT' node has a value of '100.00'. The 'INPSTATE' node has a value of 'CA'. The 'TAXAMT' node is currently empty. At the bottom of the form, there are 'Go' and 'Reset' buttons.

49

## Providing a Web Service using PHP

- **Much of the actual coding is similar to invoking a web service**
  - You will need to parse XML requests and create XML responses
- **Provide documentation of your web service (WSDL, schemas, etc.)**
- **Create a SOAPServer object**
- **Retrieve the SOAPServer “handle”**

50

## Creating a Web Service

```
<?php
class webServiceHelloWorld {
public function getGreeting() {
    $greeting = "Hello World";
    return $greeting;
}
}
$webServiceOptions = array('uri' =>
    'http://www.techreflections.com/e-learning/webService/');
$classLocation = "webServiceHelloWorld";
$testSOAPServer = new SoapServer(NULL,
    $webServiceOptions);
$testSOAPServer->setClass($classLocation);
$testSOAPServer->handle();
?>
```

51

## Invoking the Web Service (No WSDL)

```
<?php
$webServiceOptions = array('uri' => 'http://www.techreflections.com/e-
learning/webService/', 'location' => http://www.techreflections.com/e-
learning/webService/slhsoapserver.php');

$webServiceClient = new SoapClient(NULL, $webServiceOptions);

$webServiceResponse = $webServiceClient->getGreeting();

print_r($webServiceResponse);
?>
```

52

## Conclusions

53

## Web Service Concepts

- **Web services are independent of the platform, database, and programming language used to implement a solution**
- **Web services are most commonly used for business-to-business communications**
- **Most often data is transferred as XML documents over the Internet**
- **WSDL is an XML-based standard that helps define web service interactions**

54

## **i5/OS Support for Web Services**

- **Traditional i5/OS languages have limited XML support and no web services support**
- **Take advantage of IWS to generate a web service wrapper for an ILE program**
- **Utilize languages such as PHP or C# to work with web services**
- **V7R1 greatly expands XML capabilities with built-in SQL support for reading and writing XML**

55

## **More Information**

- **IWS home page**  
<http://www-03.ibm.com/systems/i/software/iws/index.html>
- **Integrating XML with DB2 for i 7.1**  
<http://systeminetwork.com/article/integrating-xml-db2-i-71>
- **i5/OS PHP information from Zend**  
<http://www.zend.com/en/solutions/modernize-ibm-i/>

56