Practical Web Services for RPG
IBM Integrated Web services for i

Tim Rowe – timmr@us.ibm.com

Presentation RPG Web Service
Business Logic

- Integrated Web services for IBM i – Project
- Intro to SOA/WS on IBM i
- Web Services coding for RPG considerations
- Create, Deploy and Testing an RPG Web service
- Consuming Web services with RPG on IBM i
Web Services Made Easy
An IBM i Project

- IBM Integrated Web Services for i
  - “An Easy Step to starting with SOA on System i”
- Simplify the process of externalizing RPG/COBOL business logic as a service.
- Externalize various RPG/COBOL business tasks as services.
- Abstracts the hidden complexities of Web services for IBM i.
- Provide RPG/COBOL Developer easy to use Web interface, not requiring additional tools or skills

http://www.ibm.com/systems/i/software/iws/
Introducing - IBM Integrated Web Services Server on i

Key Features of IBM i Web Services server
- Integrated into IBM i
- WS-Basic Profile – Compliant
- Developed for RPG/COBOL
- Open Source Technology
- Removes Complexities of Web Services for IBM i Developer
- Easy to Use - Web Admin Interface
- IBM i Web Services Test Client
- Externalizes IBM i Program Objects
- Tracing – WS-Message & Program Objects
- Scripting Support

IBM Integrated Web services server for i

- Merged the Development Process and Deployment Server
- 2 Steps to Create a Web services server on IBM i
- 7 Steps to Deploy an RPG/COBOL Service
- Built on IBM Integrated Web Application Server for i
  - Note: 2 Methods returned for every procedure or program
    - XML - Automated Data (Including Data Structures)
    - Standard Serialized objects.
- Embedded Axis 2 Engine into IBM i
- NO External tools
- NO Additional products
How do I get to it

- **IBM i 5.4**
  - 5722SS1  30  Qshell
  - 5722SS1  33  Portable App Solutions Environment
  - 5722DG1  *BASE  IBM HTTP Server for i5/OS
    - Group PTF SF99114 Level 29
  - 5722JV1  8  J2SE 5.0 32 bit
    - Group PTF SF99291 Level 27

- **IBM i 6.1**
  - GA Cum
    - Group PTF SF99115 Level 22
    - Group PTF SF99562 Level 17

- **IBM i 7.1**
  - GA Cum
    - Group PTF SF99368 Level 10
    - Group PTF SF99572 Level 6

Yea, its in V5R4!!!!
What is a service?

A repeatable business task – e.g., check customer credit; open new account

... service oriented architecture (SOA)?

An IT architectural style that supports integrating your business as linked services

"SOA impacts every aspect of IT and business."

Gartner

Web Services Overview

Definition: Self-Contained with well-defined interfaces that provide functionality that is accessible over the Internet/Intranet

Key Technologies: XML, WSDL, SOAP, UDDI
SOA Is Like Musical Notes…

Each musical notes represents a business service

Checking Credit
Opening Account
Checking Inventory
Placing an Order

SOA allows for flexible composition of music

Web services are a good start…

Turn this …

...into this (web services).

- Rich business abstractions describe the application interface
- Decouples the interfaces from the business applications
- The number and complexity of the interfaces is reduced
- Business applications and their interfaces become reusable
IWS & Android on IBM i

- IWS Enables Android Applications

RPG Coding Considerations

POWER = INTEGRATION + SOLUTIONS
RPG Best Practices – Quick Reference

- Use Free Form
- Utilize ILE Techniques
  - Procedures
  - Binding Directories
  - Service Programs
  - Exports – Hints & Tips
- Centralize Declarations
- Expand Naming Conventions
- Write Indicatorless Code
- Use Structured Programming Techniques
- Use Comments
- Avoid Obsolescence

What is a RPG Service?

- Function
  - Get information
  - Perform action
- Properties
  - Encapsulated
  - Reusable
  - Stateless
  - Event driven
  - Loosely coupled
- Modular
Properties Comparison Review – Service

**Service**
- Encapsulated
  - Access through interface
- Reusable
  - Write once – use everywhere
- Stateless
  - Information not retained
- Event driven
  - No required order
- Loosely coupled
  - Callable from anywhere

**Traditional – Sub Routine**
- Global data
  - Access directly
- Reuse by copy
  - Maintain everywhere
- Stateful
  - Information retained in job
- Application driven
  - Fixed order
- Tightly coupled
  - Tied to application

Embedding Web services information with RPG on IBM i
Example: RPG Find Customer

RPG Find Customer

V5R4 - Enablement

```
0801.00 h nomain bnddir('FLGHT400R') PGMINFO(FCRM:MODULE)
0801.01
0801.02 fCUSTOMER IF E K DISK
0801.03 fCUSTOMER IF A E K DISK RENAME(CUTR:CUSTOMERR)
0801.04 fCUSTOMERZ IF E K DISK
0801.12
0801.13 /copy nfs405pr
0801.61 ************************************************************
0803.58 p FindCustomers b export
0803.57 ************************************************************
0803.59 d FindCustomers pi
0803.59 d Position 64 const
0803.60 d ListType 1 const
0803.61 d CountReq 101 0 const
0803.62 d CountRec 101 0
0803.63 d CustList liks(CustInfo) dim(180)
0803.64 d options(*vsize)
```
Binding RPG Business Logic to Program/Service Program

- Service Info embedded with RPG or COBOL program objects (PCML)
  - For IBM i 6.1 & 7.1 need to recompile specifying:
    CRTRPGMOD PGMINFO(*PCML *MODULE)
    CRTCBLMOD PGMINFO(*PCML *MODULE)
  - For V5R4 – recompile specifying following option in the source:
    • For RPG
      H PGMINFO(*PCML:*MODULE)
    • For COBOL
      PROCESS OPTIONS PGMINFO(PCML MODULE)

- Service Information can alternatively be generated in IFS

- Restrictions:
  - Program objects must be ‘Stateless’

Externalizing Web services with RPG on IBM i
(Demo) - Simplification and Ease of Use
Web Admin: Install Web Service


Click on the Create New Web Services Server link

And now for Breaking News…
Web Integration New Feature

In the past, any user wanting to use Web Admin they were required to have *ALLOBJ and *IOSYSCFG special authority!

System Security policy just does not allow this!

New ‘Permissions’ Support integrated!!!
- Now a *USER granted ‘permission’ can use the GUI
- Group profiles are now supported

Integrated GUI interface now available to Developers and Operators without compromising your system security
Creating a Web Service

HTTP Server
IAS Web Server
Axis 2 runtime
CreateCust
FindCust
Java
IBM i Operating System

RPG runtime
CreateCust.rpg
PCML – data definition
CreateCust WSDL
FindCust.rpg
PCML – data definition
FindCust WSDL

Create Web Service – Integrated in IBM i

- Where the Web Service runs
  - Web Server - part of the IBM i operating system
    HTTP servers
    IAS server
  - Runs on Axis2 runtime engine

- Create the Web Service artifact
  - Based on a compiled RPG object
  - Uses the imbedded PCML object
  - Creates a java artifact
    Creates necessary RPG environment – User profile, library list
    Runs the specified RPG program
Web Admin: Specify Name for Server
- Step 1: Create the server to run Web services.

Web Admin: Install Web Service
- Step 2: User Profile for Web Container.
Web Admin: Install Web Service

- Step 3: Select to Install a new Web service

Web Admin: Install Web Service

- Step 4: What program contains the services?
Web Admin: Install Web Service

- Step 5: What should we call this new Web service?

- Step 6: What in the ILE program should be externalized as a Web service?
Web Admin: Install Web Service

- Step 7: Specify User for the Web service

Create Web Services Server

Deploy New Service: Specify User ID for this Service - Step 7 of 9

The service requires an IBM i user ID to run the program object that contains the Web service business logic.

Specify User ID for this Service:
- Use server’s user ID
- Specify an existing user ID

The user ID must have the necessary authority to this program object and any other additional program objects.

User ID: MyUser

Update the server’s user ID to have *USE authority to this user ID.

Web Admin: Install Web Service

- Step 8: Specify Library List for the RPG Program

Create Web Services Server

Deploy New Service: Specify Library List - Step 8 of 9

The functionality of the IBM i program you want to externalize as a Web service may depend upon other IBM i programs in the system. Specify all libraries in which programs exist that the Web service programs depend on. If a library is specified, a default library list is used.

Specify library list position for this Web service:
- Input libraries in front of your library portion of the library list
- Insert libraries at the end of your library portion of the library list

Library list entries:

Library name
- IMS
- GGPL
- QTMP
- FLCHEF00

Add Remove Remove All Move up Move down Continue

Web is performance redefined © 2011 IBM Corporation
Web Admin: Install Web Service

- Step 9: Ready to deploy the new Web service – Server tab

Create Web Services Server

Summary - Step 9 of 9

When you click Finish everything necessary to run the Web services server is created.

Servers | Services | Operations

Web Services Server Information

- Server name: WSERVICEx2
- Server description: Web services server created by the Create Web Services Server wizard.
- Internal port range: 10032 - 10031
- Server root: www/WSERVICEx2
- Server URL: http://lp03rx18.richland.ibm.com:10032
- User ID for server: qwSERVICE
- Context root: /web

HTTP Server Information

- HTTP server name: WSERVICEx2
- HTTP server description: Web services server created by the Create Web Services Server wizard.
- Port: 10032
- Document root: www/WSERVICEx2/docs
- Server root: www/WSERVICEx2
- Server association: WSDRICE

Back | Finish | Cancel

Printable Summary

Web Admin: Install Web Service

- Step 9: Ready to deploy the new Web service – Services tab

Create Web Services Server

Summary - Step 9 of 9

When you click Finish everything necessary to run the Web services server is created.

Servers | Services | Operations

Sample services:

<table>
<thead>
<tr>
<th>Sample service name</th>
<th>Sample service definition URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>ComentTemp</td>
<td><a href="http://lp03rx18.richland.ibm.com:10032/web/services/ComentTemp?_web">http://lp03rx18.richland.ibm.com:10032/web/services/ComentTemp?_web</a></td>
</tr>
</tbody>
</table>

User deployed service:

- Name: ComentTemp
- Description: "ComentTemp"
- Service install path: www/WSERVICEx2/services/ComentTemp
- User ID for service: qwSERVICE
- Program: C:\Program Files\IBM\WebSphere\webapp\comentTemp\program
- Service library list: comentTemp
- Service definition URL: http://lp03rx18:10032/web/services/ComentTemp?_web

Back | Finish | Cancel

Printable Summary
Web Admin: Install Web Service

- Step 9: Ready to deploy the new Web service – Operations tab

- Create server - after short period of time (seconds) server created and service deployed
Web Admin: Install Web Service

- Once created, the server is started and deployed service started

Web Admin: Install Web Service

- Manage installed Services – view and install new services to this server
Web Admin: Install Web Service

- View the WSDL file

```xml
<definitions xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             targetNamespace="http://findcustomers.webbeans.ileries.com">
    <element name="FindCustomers">
        <complexType>
            <sequence>
                <element minOccurs="0" name="paramC" nillable="true"/>
            </sequence>
        </complexType>
    </element>
</definitions>
```

Power is performance redefined

IBM Power Systems

Web Admin: Install Web Service

- Manage installed Services – Test Service
### Web Admin: Web Service Properties

- Manage installed Services – General Properties

**Service Properties**

<table>
<thead>
<tr>
<th>General</th>
<th>Operations</th>
<th>Library List</th>
<th>WSDL</th>
<th>Connection Pool</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name:</strong></td>
<td>FindCustomers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description:</strong></td>
<td>NFS400_THR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Start type:</strong></td>
<td>Automatic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Program:</strong></td>
<td>/QSYS.LIB/WSRM.LIB/NFS400_THR.SRVPGM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Web service definition URL:</strong></td>
<td><a href="http://10.0.0.10:9002/web/services/FindCustomers?wsdl">http://10.0.0.10:9002/web/services/FindCustomers?wsdl</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>WSDL target namespace URI:</strong></td>
<td><a href="http://findcustomers.wsbeans.iteries">http://findcustomers.wsbeans.iteries</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>User ID for this service:</strong></td>
<td>TIMMR</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Update the server's user ID to have *USE* authority to this user ID.]

**OK** | **Apply** | **Cancel**
Web Admin: Web Service Properties

- Manage installed Services – Update Library List

Security Discussion

Security Concerns
Messages are being sent over the intranet/internet – Are they Secure?

Web Services – Security
- HTTP
- WS-Security

Available Security Mechanism’s
- HTTP - SSL encryption and authorization (Certificates) (Basic authorization)
- WS-Security (No support on IBM Integrated Web services server for i)
IWS - Recent Improvements: IBM i 7.1

- Performance Enhancements...
- iASP Support
- CCSID Support via Scripts
- Connection Pool Properties
- Static WSDL
- Tracing

Static WSDL Support
Static WSDL Support

IBM Power Systems

Web Admin: Web Service Properties

- Manage installed Services – Connection Pool

IBM Power Systems
Simplified IWS Tracing….

Consuming Web services with RPG on IBM i

The Unlimited Potential

Power = Integration + Solutions
Web Services – a client view

Definition: Self-Contained with well-defined interfaces that provide functionality that is accessible over the Internet/Intranet

Key Technologies: XML, WSDL, SOAP, UDDI

Code Generation Tools (cont.)...

If RPG or COBOL program is going to use C proxy code:

Important point: It is up to the RPG/COBOL programmer to map the C structures and arrays to what is appropriate for their specific language. Can get complex when dealing with nested structures, arrays and pointers.
Integrated Web services client for IBM i

- Natural for ILE RPG/COBOL Developer
- 4 Step Static Development Process
  1. Use WSDL to generate Web service proxy code in C or RPG
  2. Build RPG stub code from C proxy code
  3. Compile/Bind RPG & Web service stub code
  4. Invoke RPG/COBOL Web service client
     \[http://www.ibm.com/systems/i/software/iws/\]

Web Services Client for ILE

- Overview
  - Based on Apache AXIS C++ Version 1.5+
  - Consists
    - Tools
      - Convert service’s WSDL to C/C++ APIs
      - SOAP client
      - ILE Service Program
    - Supported today
      - Packaged with 5733-XT1
      - V5R1, V5R2, V5R3, V5R4
    - Availability
      - 4Q/07 – IBM i SS1 Option 3
      - V5R4 and later
- Supports
  - C, C++, RPG, COBOL
  - Web Services Description Language (WSDL) - document literal only
  - Web Services Invocation (WSI) 1.1 basic profile compliance
  - Secure Sockets Layer (SSL)
Web Service Deployment Review

Where to find my WSDL
FindCustomers.WSDL

Defines the message structure for the operations

Service Interface

Operation and Binding Protocol

URL

31
Web Service Client Support

- Integrated Web services client for ILE makes consuming Web services from ILE languages (RPG, COBOL, C, C++) easier
  - Web service communication layer is simplified
    Consumers don’t need to worry about any of the underlying TCP/IP, HTTP, or SOAP protocols because of Apache Axis client
  - Not fully automated
    Automated – stub code easily generated to talk with a Web service
    Not Automated - Web service consumer applications still need to have invocation logic implemented to properly consume deployed Web services
- Note:
  The ILE client supports consuming Web services deployed in any language (Java, PHP, ILE RPG, ILE C, etc) on any platform
Web Service Flow - Client

- Request
- C/C++ stubs (service, method)
- Axis Client
- SOAP Request
- SOAP Response
- Call/Return
- Server
- Call/Return
- Response

WSDL2WS Tool – Existing function today

- Generates C stubs, so a pure ILE solution
  - But NOT a pure RPG solution
- This is the only option available for ILE programmers today
- Can be difficult for RPG programs to call C stubs
  - Parameters can be difficult to pass
  - Complex structures are hard to map and prone to errors
  - Potential for memory leaks
- Many RPG programmers really want to stay in RPG
Generating C Stub Code

- ILE client applications need stub code to successfully communicate with a deployed Web service.

  **WSDL**

  Describes a Web service's details

  **WSDL2WS Tool**

  Only option available today

  **C Stubs**

  Callable from any ILE language (RPG, Cobol, C, etc)

Web Services Client for ILE

**Step 1: Stub Generation – Creating a Web Services Proxy**

- Developer generates stubs using:
  - Java tools (wsdl2ws.jar)
  - Qshell script - `wsdl2ws.sh -c STOCKQ.wsdl`

  **WSDL**

  WSDL passed into tool that generates C/C++ stubs

  **C/C++ stubs**

  Files generated:
  - CUSTINFO_Array.c
  - FindCustomersPortType.c
  - FINDCUSTOMERSResult.h
  - CUSTINFO_Array.h
  - FindCustomersPortType.h
  - RSPERIOD_T.rpgie
  - FINDCUSTOMERSInput.c
  - transform
  - FINDCUSTOMERSInput.h
  - FINDCUSTOMERSResult.c
  - FindCustomers.wsdl
Using RPG Stubs to Invoke Web Services

- **Web services client for ILE was enhanced to generate RPG stubs**
  - In December 2010, PTFs released on IBM i 5.4, 6.1, and 7.1 to support the generation of RPG stubs
  - **Before** support for RPG stubs, users had to:
    1. Generate C stubs
    2. Compile the C stubs into a service program
    3. Generate RPG prototypes for function calls
    4. Generate RPG data types corresponding to C structures and types
    5. Deal with pointers
    6. Keep track of dynamically allocated storage

- **With RPG stubs**, users will:
  1. Generate stubs and build service program automatically
  2. Code to 2 RPG include files that contain prototypes, data types, etc

  **No pointers or dynamic storage to manage!**

Solving the RPG Issue

- Provide an easier path for ILE RPG programs to consume a Web service

- New tool - **WSDL2RPG**
  Generate RPG stub code directly from a WSDL file

- Allow RPG programmers to decide whether or not they want to work with C stub code
WSDL2RPG Tool

- WSDL2RPG generate RPG stub code directly from WSDL files
  - C stub code no longer needs to be called by user
  - Pure ILE RPG solution at their disposal

- RPG Stub code is less error prone
  - Parameters are clearly defined as RPG data structures instead of pointers
  - No need to map RPG prototypes to C procedures names
  - Overly complex WSDL files can still cause problem
  - No more memory management issues

- RPG programmers can stay in RPG and avoid C altogether

Generating RPG Stub Code

- ILE client applications need stub code to successfully communicate with a deployed Web service

```
C Stubs

Used by RPG stubs behind the scenes

RPG Stub
Code
*SRVPGM

Service program containing RPG stub code procedures

Callable from any ILE language
```
WSDL2RPG Example

- **Description:** RPG Web service that accepts a customer number (string) as input and returns a customer record consisting of a customer number, customer name, customer phone number

- **Example QSHELL invocation:**

  ```bash
  > cd /home/examples/demo
  > /qibm/proddata/os/WebServices/V1/client/bin/wsdll2rpg.sh GETCUST.wsdl -p dem -f demo -lsrc demolib -lpgm demolib
  
  Code generation completed. Generated files in directory '/home/examples/demo'.
  ```
What RPG source code gets automatically generated?

- `/home/examples/demoPR.rpgle`
  - Contains RPG data structure and prototype definitions
- `/home/examples/demo.rpgle`
  - Contains the implementation procedure interfaces for the RPG stub code
  - This should never need to be looked at by a developer
- `/home/examples/demomk.clp`
  - Contains the commands run to create the service program containing the RPG stub code procedures
  - This should never need to be looked at by a developer

RPG source code is created in both IFS and in source physical file members

Service program containing RPG stub code is automatically created. It just needs to be bound to the ILE client application consuming the Web Service.

C stub code still generated – used by the RPG stubs

Data structures found in the `demoPR` source file:

- Output Data Structure
  ```
  TRUNC
  0 demGETCUSTOMERResult_t...
  0 0 DS qualified based(Template)
  0 0 REC000 qualified based(Template)
  ```

- Input Data Structure
  ```
  TRUNC
  0 demGETCUSTOMERInput_t...
  0 0 DS qualified based(Template)
  0 0 CUSTNR1 128a
  ```

- Web Service Endpoint Data Structure
  ```
  TRUNC
  0 demThis_t 0 DS qualified based(Template)
  0 demEndPoint 1024
  ```

- Customer Record Data Structure
  ```
  TRUNC
  0 demQUOTREC_t 0 DS qualified based(Template)
  0 0 CUSTNO 128b
  0 0 CUSTNR 128a
  0 0 PHONE 128a
  ```
WSDL2RPG Example (continued...)

- Procedure found in the `demoPR` source file:

    ```rpg
    * (Public)
    d demogetcustomer PR in extr('demogetcustomer:P')
    d this likes(demThis_t)
    d result likes(demGETCUSTOMERResult_t)
    d end
    ```

  - Work with Web Service Endpoint Procedure (verifies the web service is available)
    ```rpg
    * (Public)
    d demoWebWith PR in extr('demoWebWith:')
    d this likes(demThis_t)
    ```

Example ILE Client (consumer) Application

```rpg
/* 
 *Example ILE Client (consumer) Application
 */

d demoxxx pr in extrm('demoxxx')

d temp 128a

d demo2 pr in extrm('demo2')

d temp 128a

copy demoPR

d this do likes(demThis_t)

d input do likes(demGETCUSTOMERInput_t)

d result do likes(demGETCUSTOMERResult_t)

d output  s 58a int(4blanks)

copy demoPR

d this do likes(demThis_t)

d input do likes(demGETCUSTOMERInput_t)

d result do likes(demGETCUSTOMERResult_t)

do result d

copy demoPR

d this do likes(demThis_t)

do result d

copy demoPR

d this do likes(demThis_t)

do result d

copy demoPR

d this do likes(demThis_t)

do result d

copy demoPR

d this do likes(demThis_t)

do result d

copy demoPR

d this do likes(demThis_t)

do result d

copy demoPR

d this do likes(demThis_t)

do result d

copy demoPR

d this do likes(demThis_t)
```
WSDL2RPG Example (continued...)

* Example ILE Client (consumer) Application (continued...)

```rpg
// web service is available
if dworkwith(this):

    // Assign input customer no. to the value passed into the program
    input.dem_custno = temp;

    // Call web service to get the customer record information
    if demgetcustomer(this,input,result):

        // Output customer no.
        output = 'Customer No.';
        output = trim(output) + result.dem_record.dem_custno;
        disp output;

        // Output customer name
        output = 'Customer Name:';
        output = trim(output) + result.dem_record.dem_name;
        disp output;

        // Output customer phone number
        output = 'Customer Phone:';
        output = trim(output) + result.dem_record.dem_phone;
        disp output;
        endif;

        *inrl = *on;
        endif;

/*end-free
```

Note: The bound service program (DEMOLIB/DEMO) contains the RPG stub code procedures generated by the WSDL2RPG tool.

- Output when customer number '45731' is input:

```
DSPLY Customer No=45731
DSPLY Customer Name=Acme Toy Company
DSPLY Customer Phone=1.800.GET.TOYS
```
RPG Call Find Customers Program

```xml
<RPG Find Customer Web Service
  <?xml version="1.0" encoding="UTF-8"?>
  <FINDCUSTOMERS>
    <CUSTLIST>
      <NAME>Brown, Jacqueline</NAME>
      <NUMBER>1781</NUMBER>
    </CUSTLIST>
    <CUSTLIST>
      <NAME>Brownigg, Daphne</NAME>
      <NUMBER>2313</NUMBER>
    </CUSTLIST>
    <CUSTLIST>
      <NAME>Brownigg, Ellen</NAME>
      <NUMBER>5315</NUMBER>
    </CUSTLIST>
  </FINDCUSTOMERS>
</RPG>
```
Quick Review of Steps

Process to Run RPG Web Service:
1. Stub Generation – Creating a Web Services Proxy ("intermediary" for ILE RPG, COBOL)
2. Use Stubs to build RPG prototypes
3. Compile/Bind and Invocation
4. Run Program

• Prerequisites
  – C++ Compiler (Compiler - ILE C++, licensed program product ID 5722WDS, option 52)
  – Java (IBM Developer Kit for Java, JDK 1.4, licensed program product ID 5722JV1, option 6+)
  – C Compiler (Compiler - ILE C, licensed program product ID 5722WDS, option 51)
    Only needed if generating C stubs

IBM Systems Lab Services and Training

Composed of experts who develop and deploy solutions across IBM's systems family offerings. From in-depth product expertise, to knowledge transfer, to platform-specific hardware and software solutions, we're here for you!

Web Site:
http://www.ibm.com/systems/services/labservices

Sample Power Services
• Web Services for RPG
• Sockets Communications Programming
• Optical Storage Migration
• Performance Tuning
• External Disk Implementation
• IASP Implementation
• Security
• HA Consulting

Contact Us
stgls@us.ibm.com
NEW!! IBM i zone on developerWorks

✔ New resource for the IBM i community
✔ Wide variety of technical information

ibm.com/developerworks/ibmi

IWS - Web site Information:

- Integrated Web Services for IBM i - Website:
  http://www.ibm.com/systems/i/software/iws/

- Server/Client Tutorials:

- Limitations and Restrictions:

- How to Order: - Group PTF - for V5R4 and V6R1
  http://www.ibm.com/systems/i/software/iws/support.html
Published Articles: - Chronologically

- Integrated Web Services for IBM i - June 2008

- A Walk in the Park - RPG developer Perspective: Dec 2008

- "Cover Article" - Diving Into Web Services and SOA on IBM i: Dec 2008

- Getting Started With the Integrated Web-Services Server - Dec 2008

- Power Under the Hood - Dec 2008

- Securing Web-Service Messages: - January 2009

- Integrated Web Services Server for IBM i - Feb 2009
  http://www.mcpressonline.com/application-software/integrated-web-services-server-for-ibm-i.html

Other - Resources

- Integrated Web Services for IBM I
  http://www.ibm.com/systems/i/software/iws/

- IBM Technical Information and Example
  http://www.ibm.com/developerworks/

- WebSphere Enterprise Service Bus
  http://www.ibm.com/software/integration/wsesb/

- WebSphere Process Server
Summary

- Integrated Web services for IBM i – Project
- Intro to SOA/WS on IBM i
- Web Services coding for RPG considerations
- Create, Deploy and Testing an RPG Web service
- Consuming Web services with RPG on IBM i

Questions?

Thank You