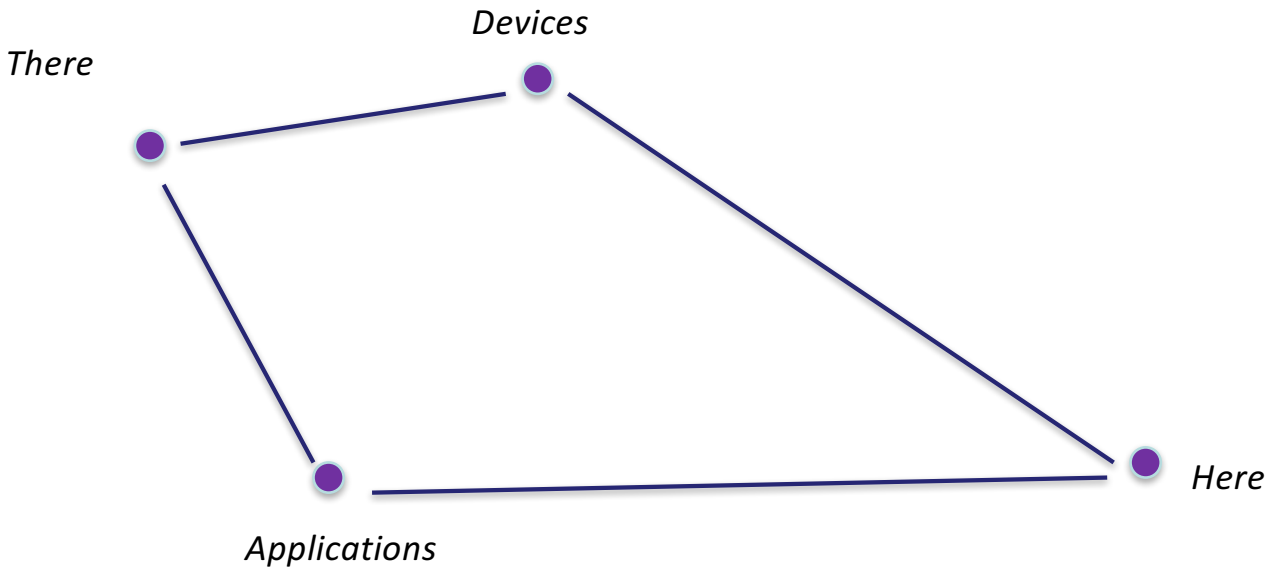
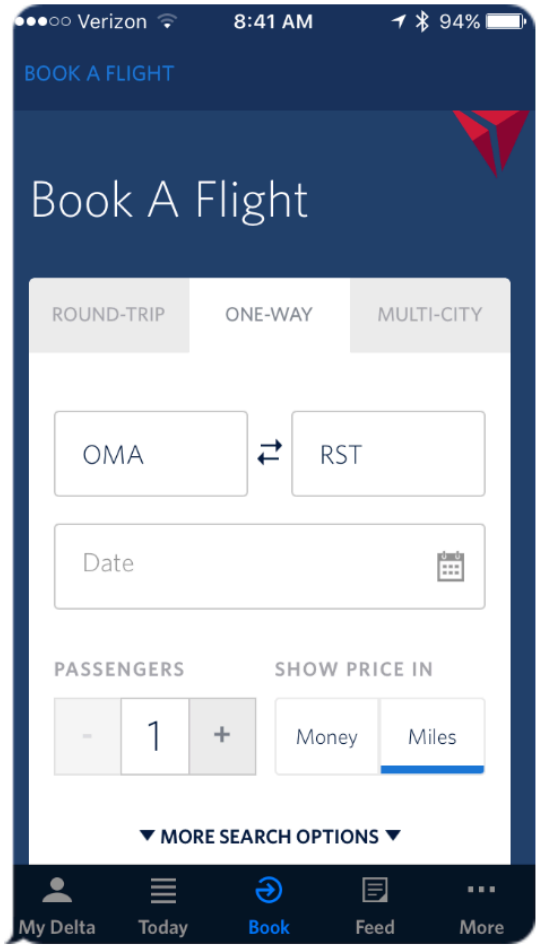

IBM
Cognitive Systems
SOAP/ REST and IBM i & Watson
Tim Rowe- timmr@us.ibm.com
Business Architect Application Development
for Business
for Business

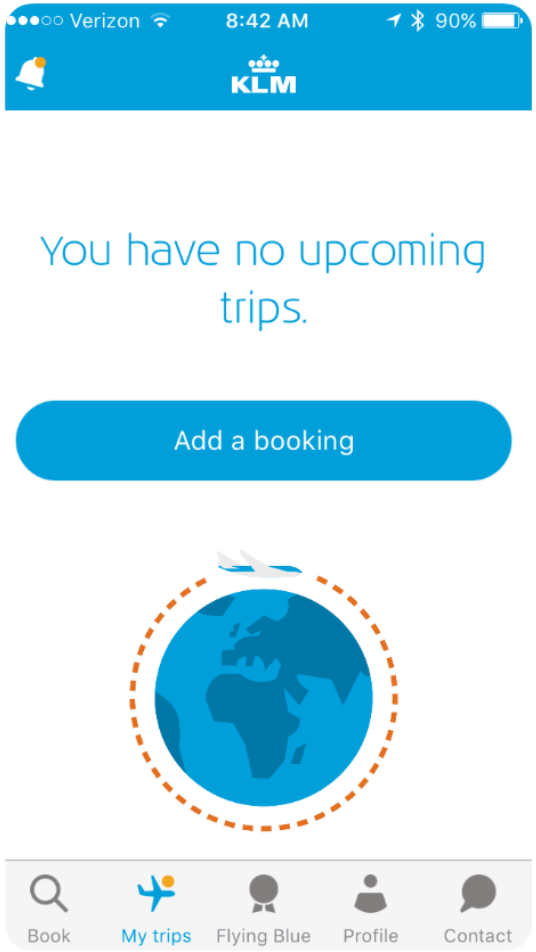
What is an API - Agenda

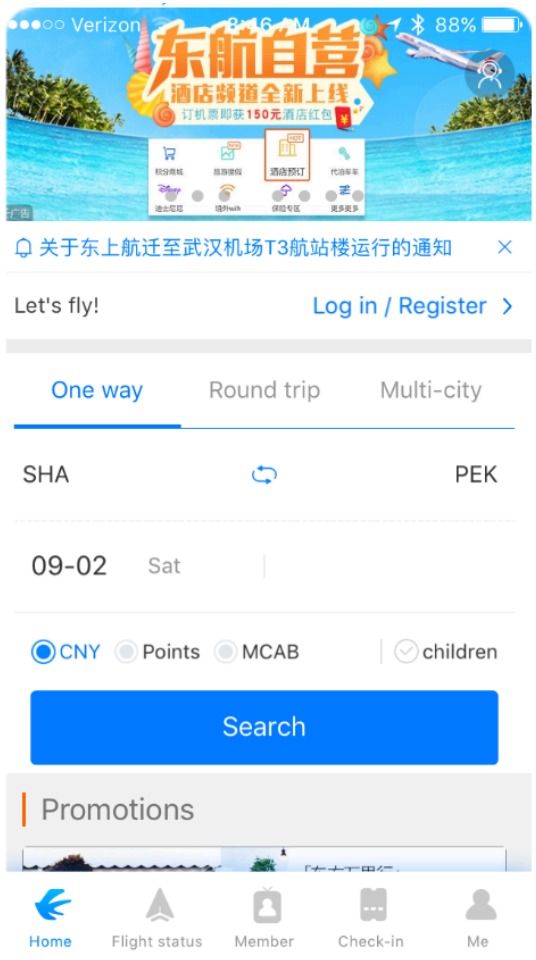
- What is an API
- What is a Web Service
- SOAP vs REST
  - What is SOAP
  - What is REST
  - Benefits
  - Drawbacks

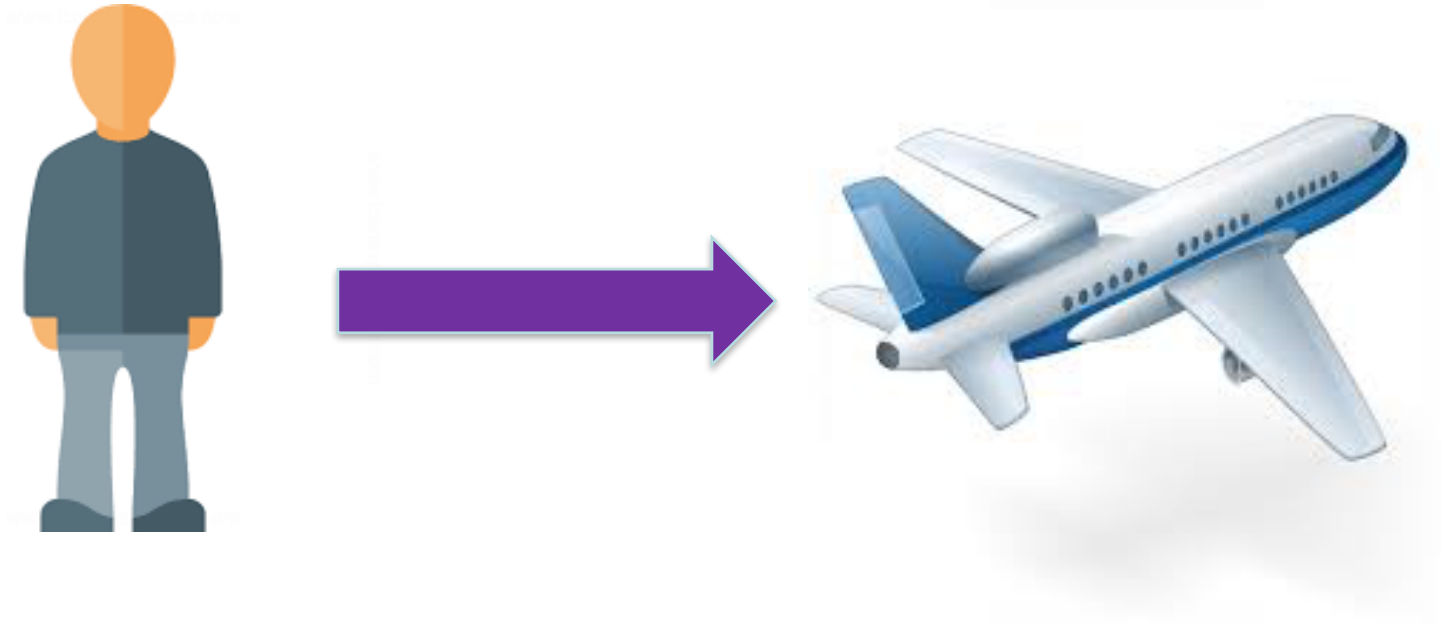
Connections

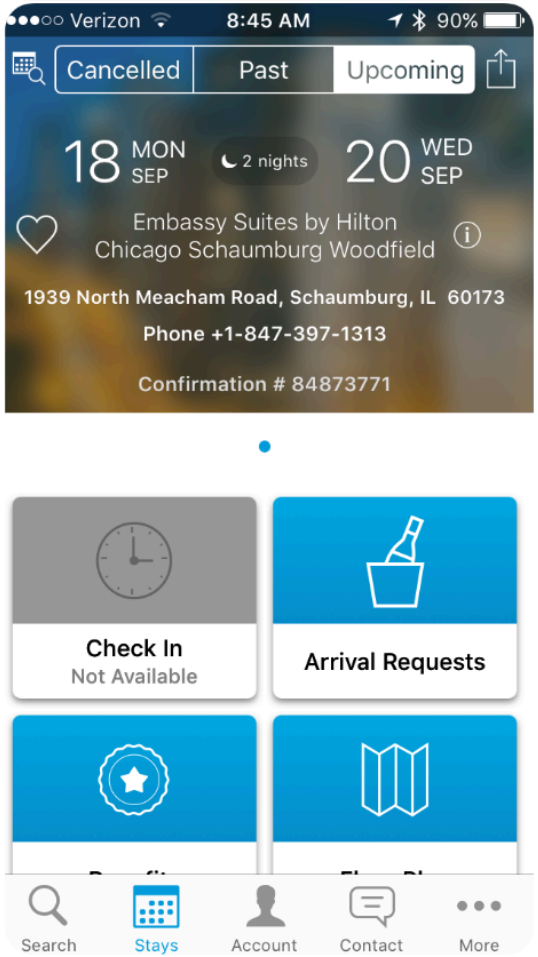


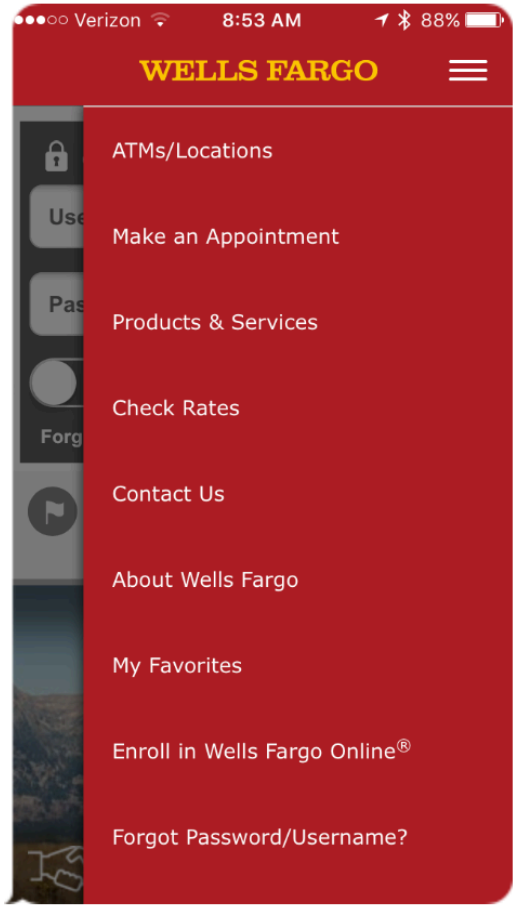












API Definition

Application

Programming

Interface

API Definition

Application programming interface

From Wikipedia, the free encyclopedia

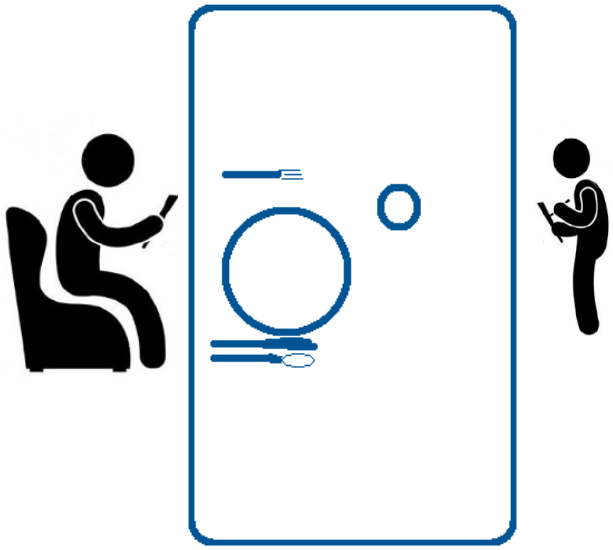
"API" redirects here. For other uses, see API (disambiguation).

In computer programming, an **Application Programming Interface** (**API**) is a set of subroutine definitions, protocols, and tools for building application software. In general terms, it is a set of clearly defined methods of communication between various software components. A good API makes it easier to develop a computer program by providing all the building blocks, which are then put together by the programmer. An API may be for a web-based system, operating system, database system, computer hardware or software library. An API specification can take many forms, but often includes specifications for routines, data structures, object classes, variables or remote calls. POSIX, Microsoft Windows API, the C++ Standard Template Library and Java APIs are examples of different forms of APIs. Documentation for the API is usually provided to facilitate usage.

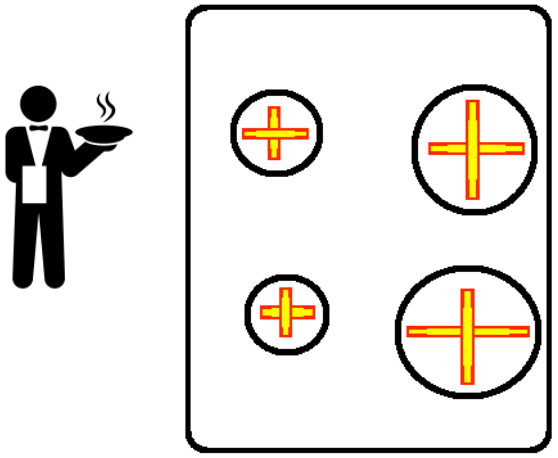
APIs - Simple

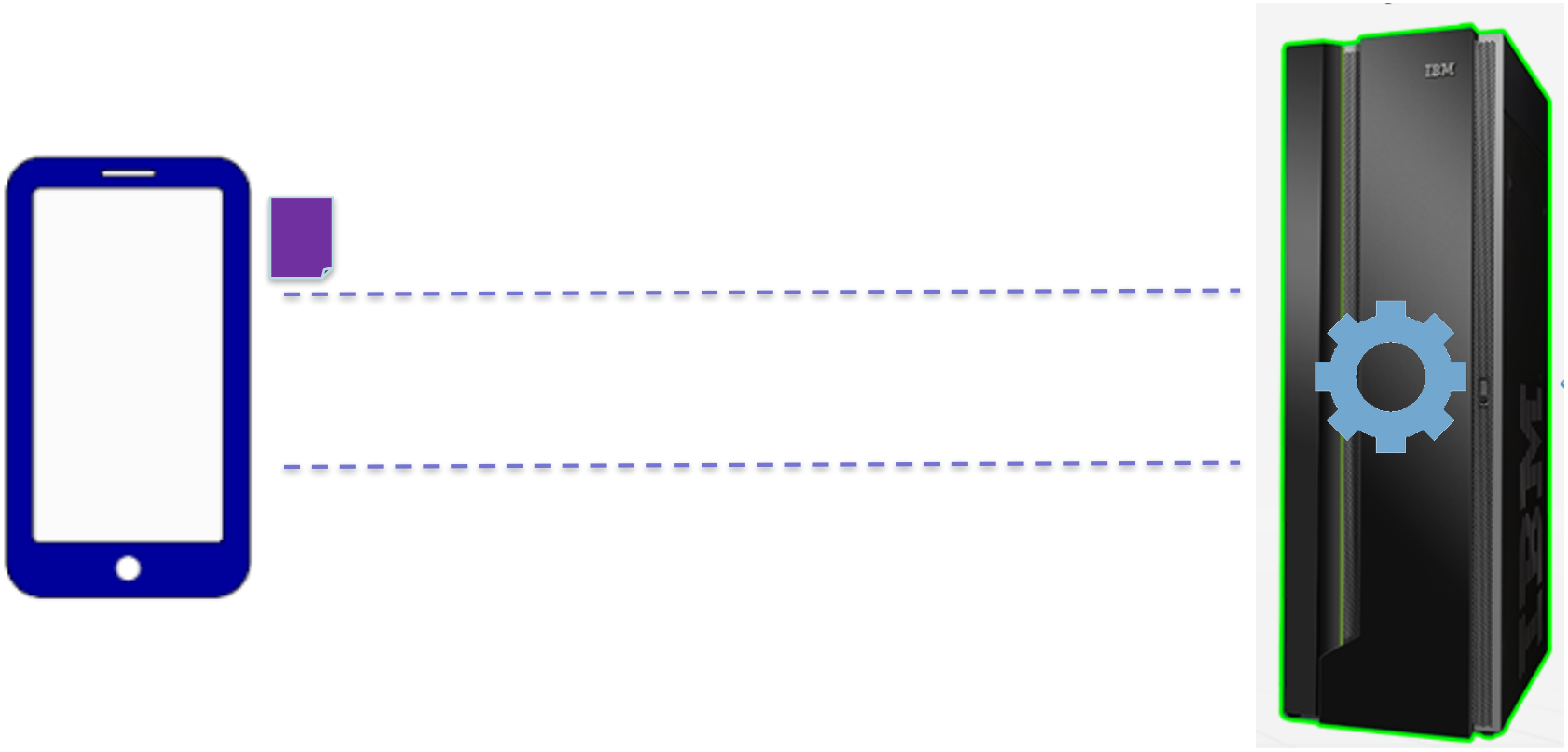
Simple way to connect endpoints. Send a request and receive a response.

Example



Kitchen





The API Economy

Not just a buzz-word, but rather the evolution of services-oriented IT. Allows users, businesses & partners the ability to interact in new and different ways resulting in the growth (in some cases the revolution) of business.

What is the API economy?

- Cloud, mobile and social - business as-a-service economies
- Data has considerable value and can be monetized given an easy-to-consume API
- APIs standard building block for doing developme
  - Mobile
  - Web
- Easy integration with other apps and services
- APIs provide consumability for 3rd Parties – broaden your reach

What is the API Economy

Creating value by offering APIs that others want

Using APIs to help your developers innovate freely

Supporting your mobile development team with APIs

What is the API Economy

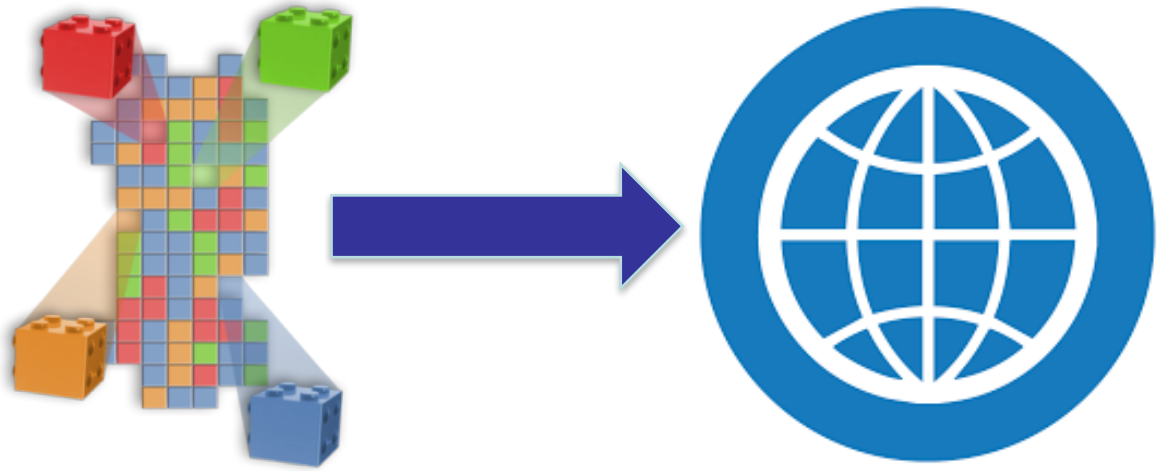
Making APIs the common language in a hybrid world

Linking devices to data on the Internet of Things (IoT)

What is a Web Service ?

… a service?

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



SOAP vs REST

Simple

Object

Access

Protocol

What is SOAP

- Exposes **operations** that implement logic
- Designed for distributed computing
- Standardized
- Aligns with Enterprise Application needs
  - Support multi transport
  - Enterprise security – WS.Security
  - Governance with strong typing
  - Broad Development tooling support
- XML Based message protocol
- Uses WSDL as a contract between consumer and provider

REpresntational

State

Transfer

What is REST

- Architectural Style as described by Roy Fielding
- Resource focused
- Every request is via hyperlink ie http request
- Easily consumed by any client, especially web clients
- Light weight
  - Uses JSON vs XML
  - No required header for each message
- Resources are driven by HTTP Specification
  - GET, PUT, DELETE, POST

Why one vs the other ? Philosophical Difference

SOAP

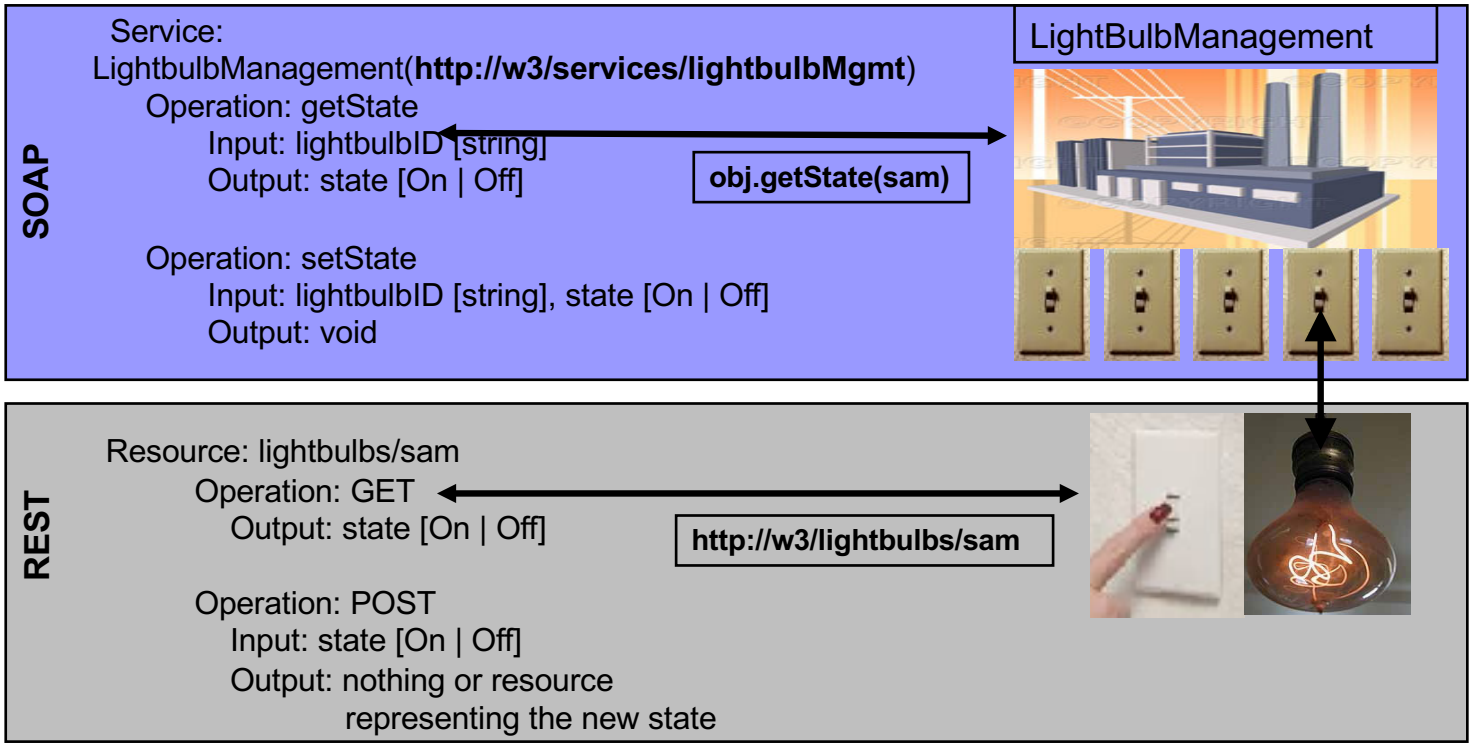
- Enterprise Driven
- Contract based
- Robust Infrastructure
- More Security Options

Rest

- Simplicity
- Small packet size
- HTTP focused
- Easy to call from JavaScript

SOAP vs. REST example

Is the light bulb currently on?



SOAP vs. REST example data flows

SOAP request

```
POST /services/LightBulbManager HTTP/1.1
Host: example.com
Content-Type: text/xml; charset=UTF-8
SOAPAction: "LightBulbManager#getState"

<?xml version='1.0' ?>
<env:Envelope xmlns:env="…">
<env:Body>
  <ns1:getState xmlns:ns1="…">
    <in0 xsi:type="xsd:string">SAM</in0>
  </ns1:add>
</SOAPenv:Body>
</env:Envelope>
```

REST request

```
GET http://w3/lightbulbs/SAM HTTP/1.1
Host: example.com
Accept: application/xml
```


IBM
Cognitive Systems
Creating Web Services on IBM i
Tim Rowe - Business Architect for Application Development
timmr@us.ibm.com
for Business

Using a REST API with Watson

https://ibm-i-watson-test.mybluemix.net/

What are the characteristics of a web service?

Web 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 subroutine

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

Simple View

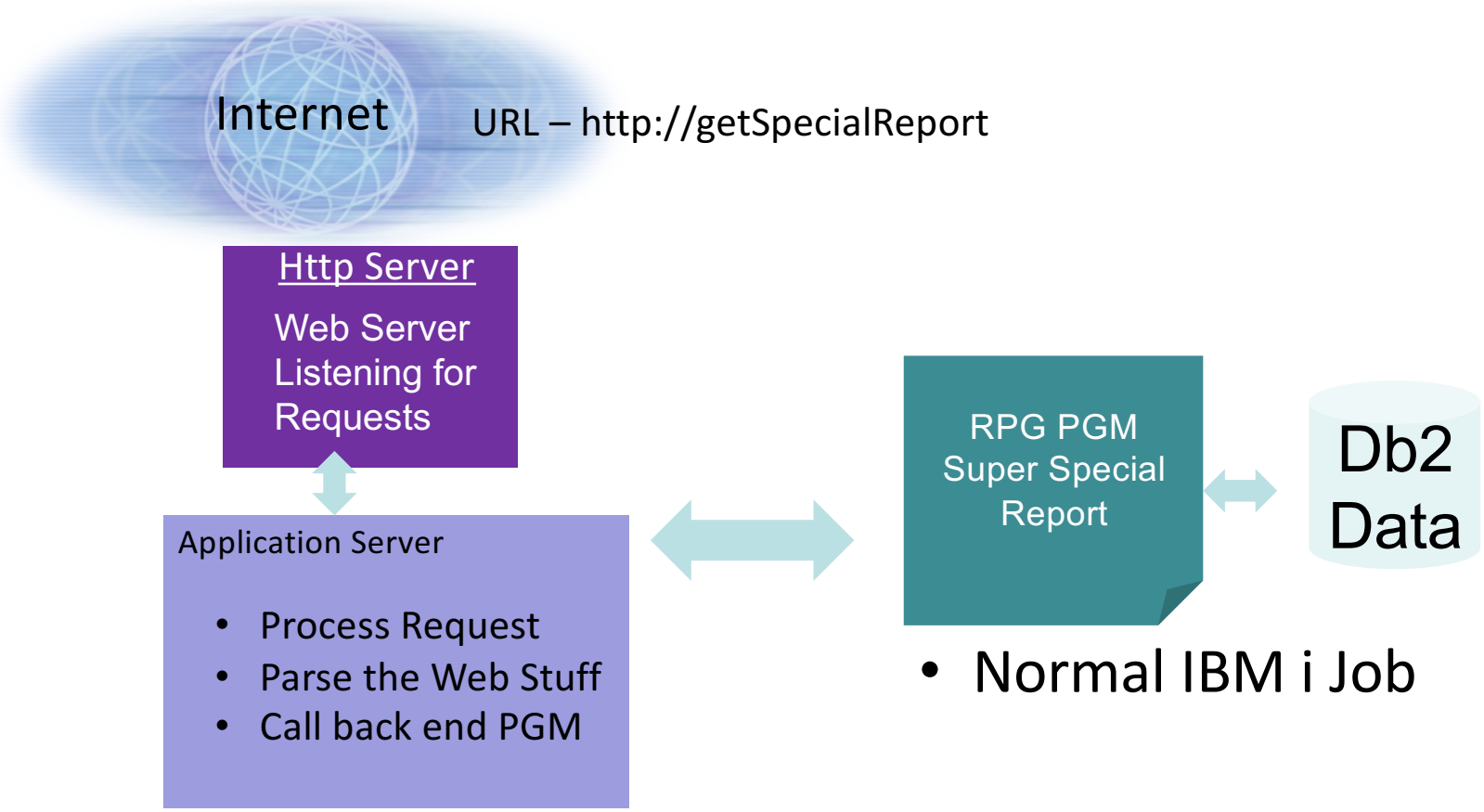
IBM i

RPG PGM
Super Special
Report

???

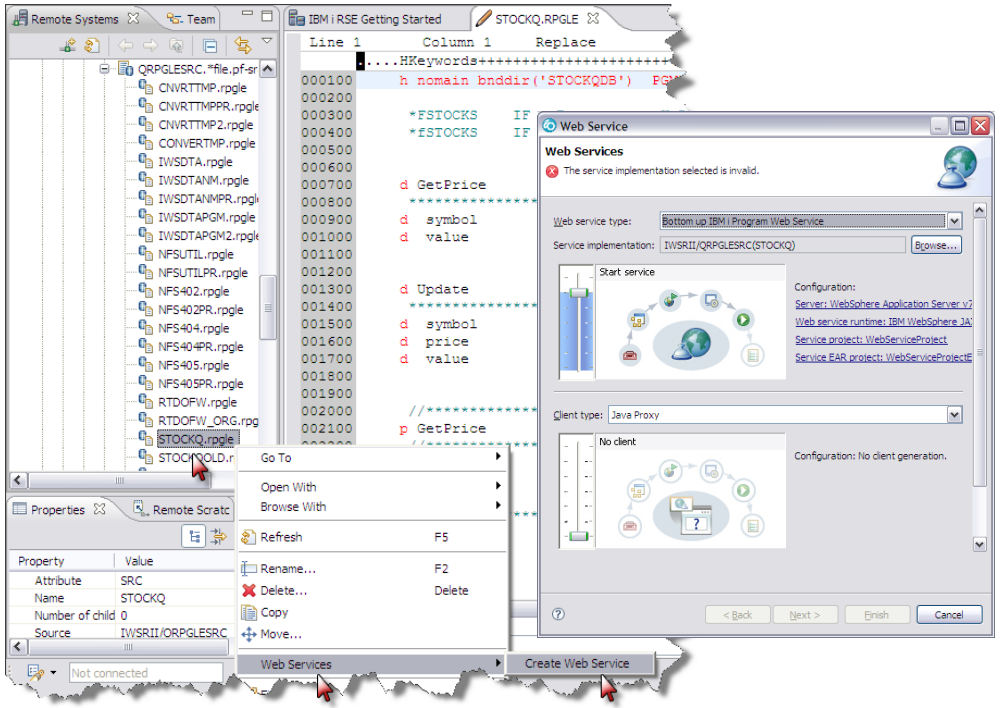
How do you connect your IBM i to the outside world ?

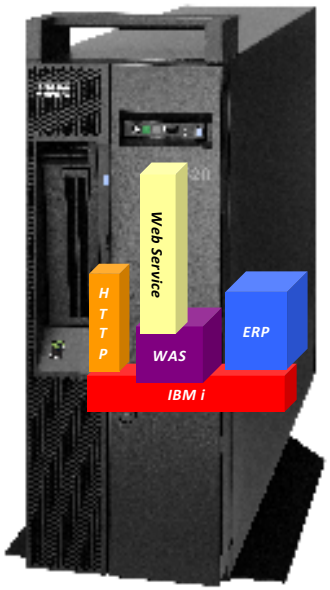
Logistics



Rational Developer for i

RDi & Modernization Tools: Web Services Wizard

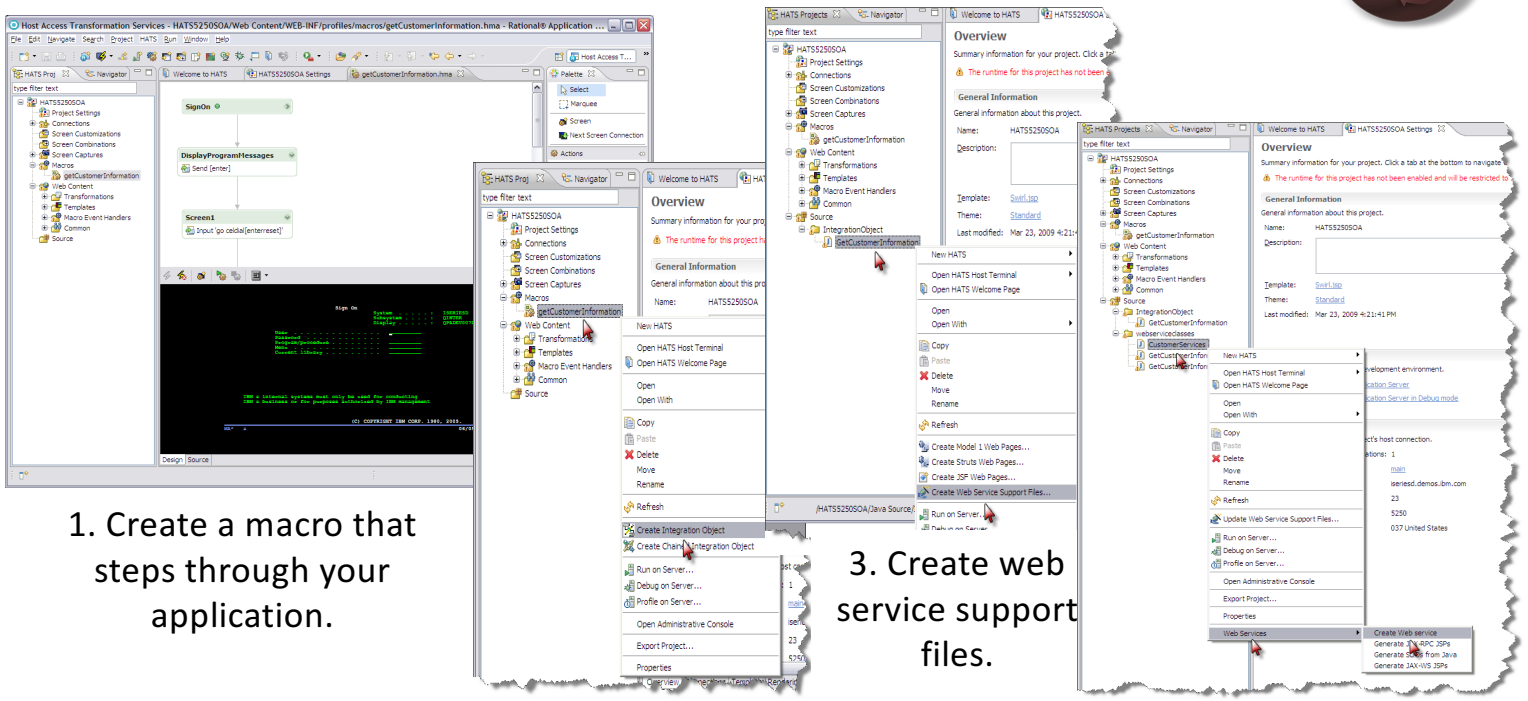




More Information:
http://www-01.ibm.com/software/awdtools/developer/rdisoa/

Host Access Transformation Services

IBM Rational HATS Toolkit



1. Create a macro that steps through your application.

2. Create an Integration Object.

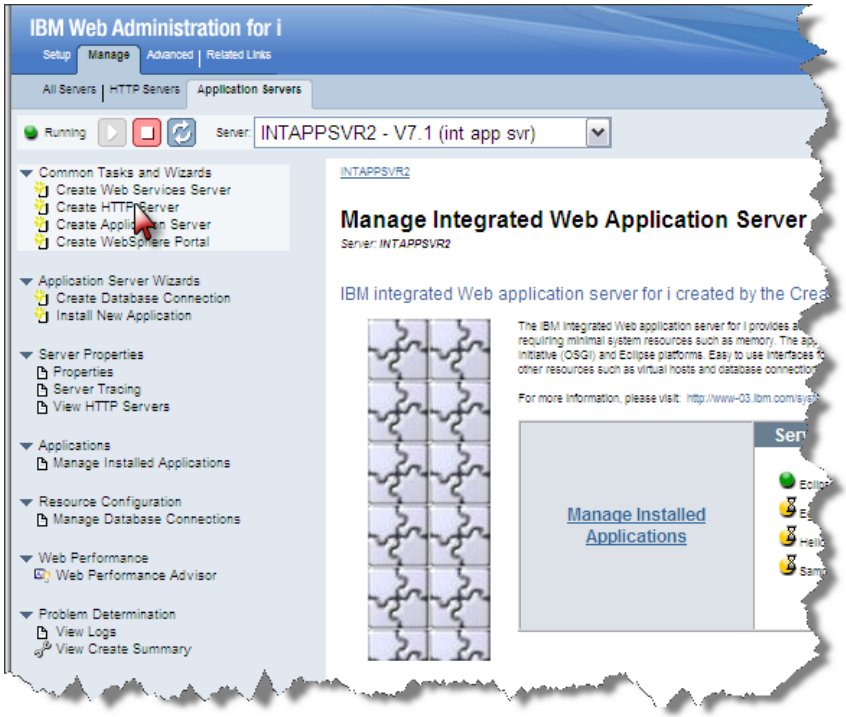
3. Create web service support files.

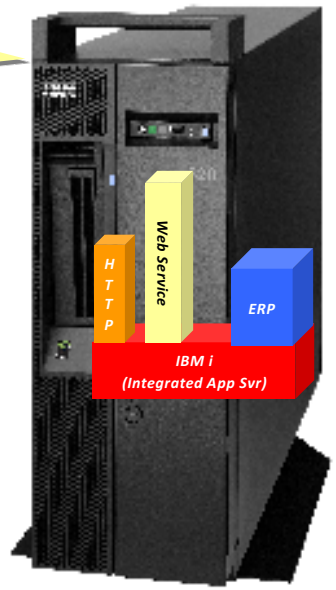
4. Create the web service and WSDL.

More Information:
http://www-01.ibm.com/software/awdtools/hats/

IBM i Integrated Web Services Environment

IBM i: Integrated Web Services Server *SOAP & REST*



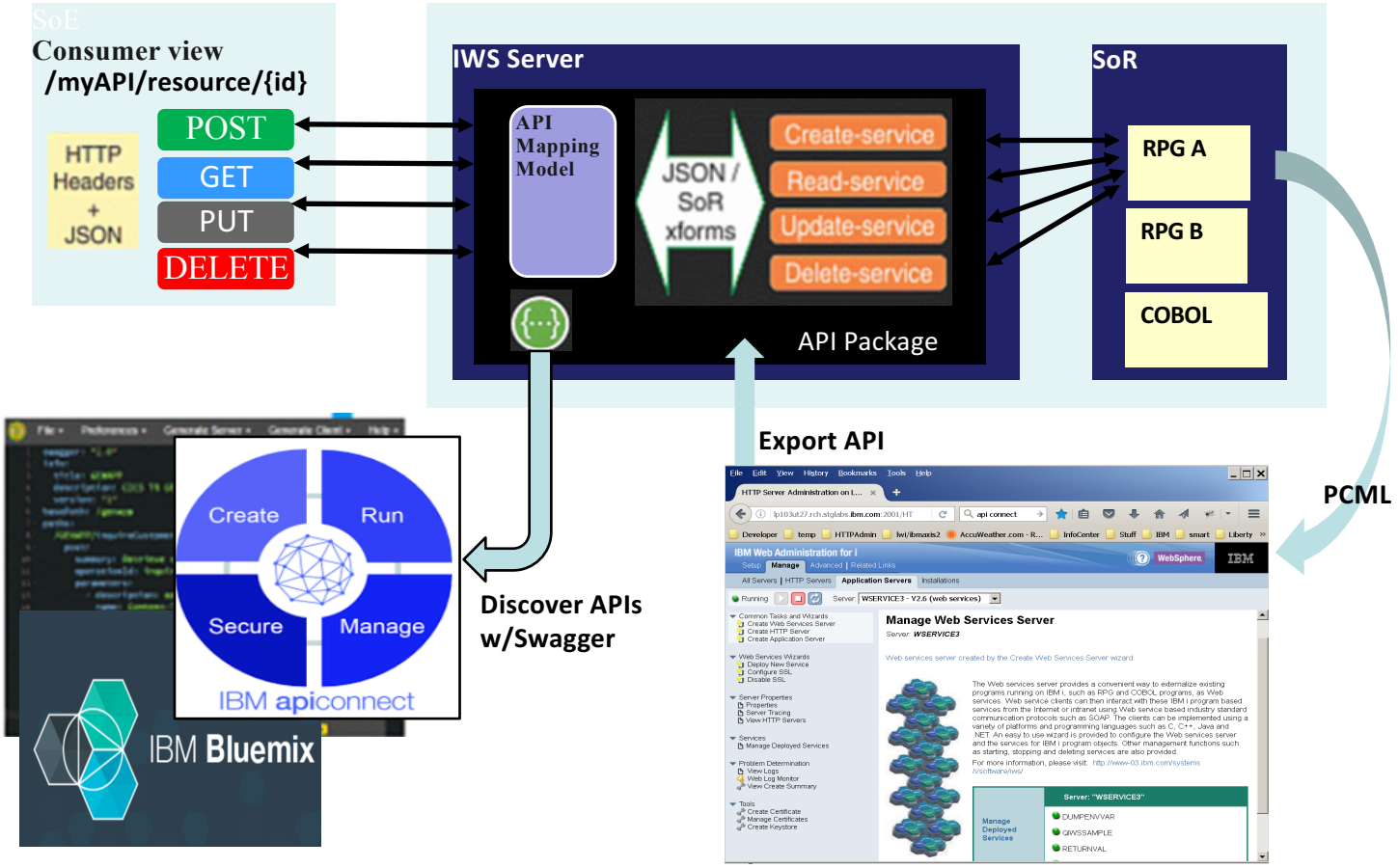


About integrated web services

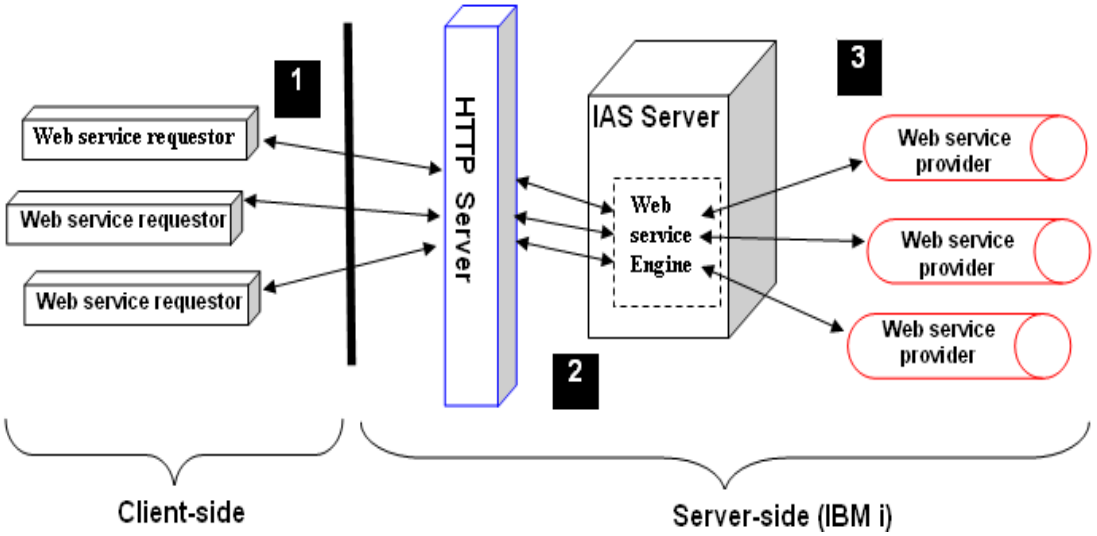
- Released December of 2007 on IBM i 5.4, 6.1, and 7.1
  - Installed as part of base operating system option 3
  - Always load latest HTTP Group PTF for latest fixes and enhancements
- Consists of two separate entities
  - Integrated web services client for ILE
  - Integrated web services server
- Latest information, including product prerequisites, can be found at http://www.ibm.com/systems/i/software/iws/

- Continues to be re-invented and enhanced on 7.2 & 7.3

IWS server API requests – mapping & transformation



Web service client/server flow does not change



- When a web services server is created, an associated HTTP server is also created
  - You can go straight to the web services server, but if you need SSL or basic authentication, you need to do it via the HTTP server
- The web service provider implementation code (i.e. RPG or COBOL programs or service programs) are run in separate jobs

What are all the parts…

- HTTP Apache Server
  - Connector to the IAS server
- IAS Server (Liberty)
  - JAX-RS (REST)
  - JAX-WS (SOAP)
  - Java program
    - Handles Inputs
    - Calls the backend ILE Program
    - Converts Output back to Web format

About integrated web services server REST support

- Supported in IBM i 7.1 and 7.2 and 7.3
  - On version 2.6 of integrated web services server
  - Server will handle both SOAP and REST services
- Uses JAX-RS
  - Java API for RESTful Web Services
- Two ways to deploy a REST service
  - IBM Web Administration GUI updated
    - Deploying a REST service will require more user input than when deploying a SOAP service
  - QShell script installWebService.sh updated to support REST

How to Get Started - Rest

- Identify the RPG / Cobol / Java pgm
- Figure out the HTTP methods
  - GET - read activities
  - POST - create entries
  - PUT - update an entry
  - DELETE – remove
- Determine the URI - Identifiers
  - Use Nouns vs Verbs
  - Keep it simple

Getting Started

Examples

- GET /defects: list all bugs.
- GET /defects/123: Retrieve bug 123.
- POST /defects/123: Create a new defect 123 with the POST request body.
- PUT /defects/123: Update defect 123 with the PUT request body.
- DELETE /defects/123: delete defect 123

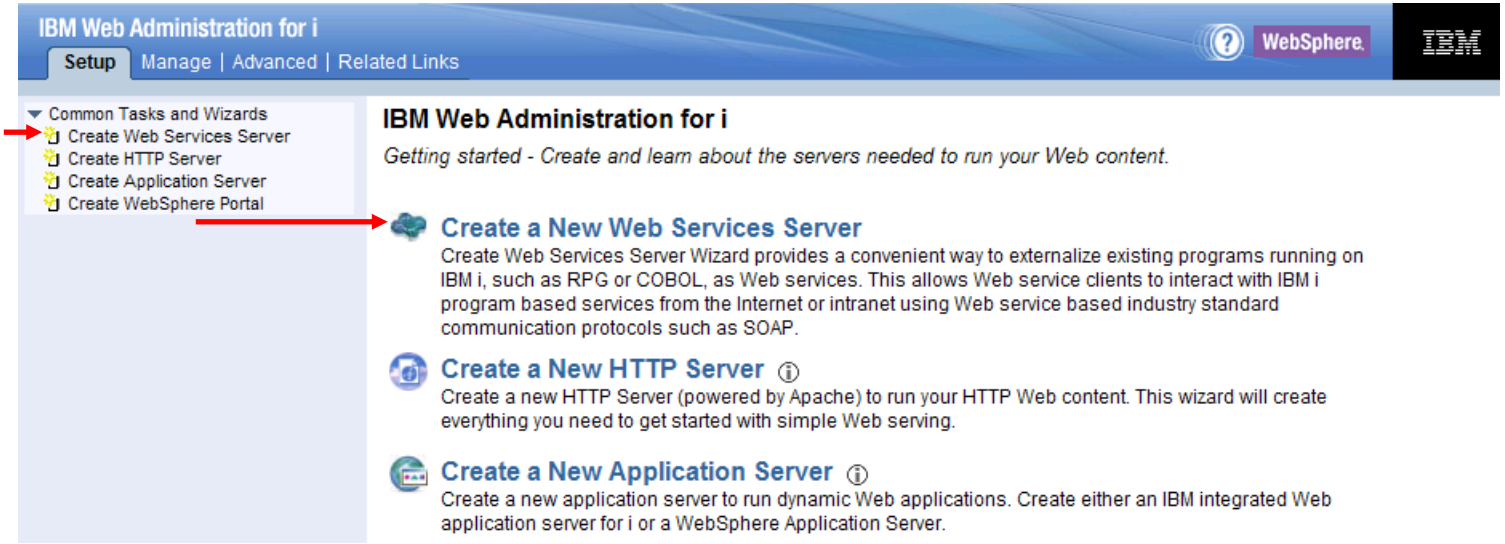
Getting Started

Planning

- Specify media types (e.g. XML, JSON, etc.) the procedure will accept
- Specify media types the procedure will return
- Optionally specify what values to inject in procedure input parameters
  - Path segment (e.g. `/accounts/{id}`)
  - Matrix parameters (e.g. `/cars;color=blue`)
  - Query parameters (e.g. `/cars?color=blue`)
  - Form data
  - HTTP headers
  - HTTP Cookies
- Optionally designate response code and HTTP header output parameters

Create web services server

Access Web Admin http://*hostname*:2001/HTTPAdmin



Click on the Create New Web Services Server link

Web Integration Permissions

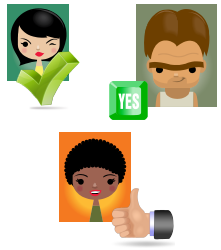
In the past, any user wanting to use Web Admin they were required to have *ALLOBJ and *IOSYSCFG special authority!

System Security policy just does not allow this!

‘Permissions’ Support

- Now a ***USER** granted ‘permission’ can use the GUI
- Group profiles are now supported

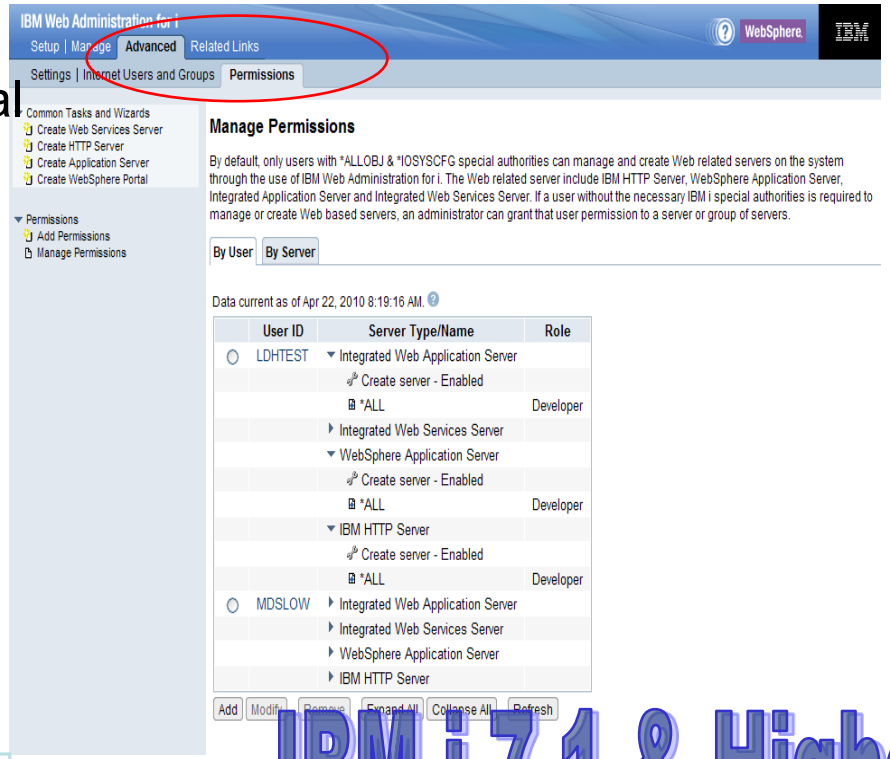
NEW



Web Integration Permissions

- **Developers can use Web Admin**
  - No longer need *ALLOBJ special authority
  - Administrators can grant users 'Permission'
  - Empowering the User
  - Group Profile support

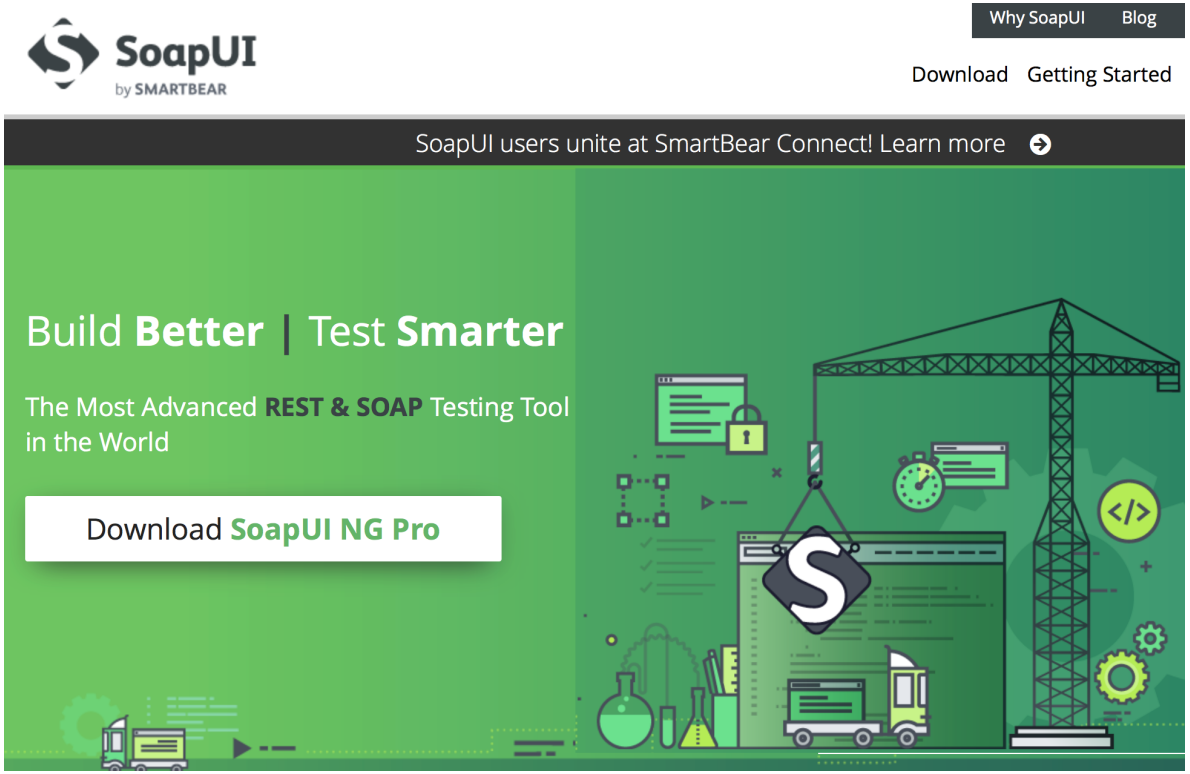
- **Two Permissions Available**
  - Operator – Start & Stop servers
  - Developer – All functions



IBM i 7.1 & Higher

Integrated GUI interface now available to Developers and Operators without compromising your system security

How do you test things ?



Free Download

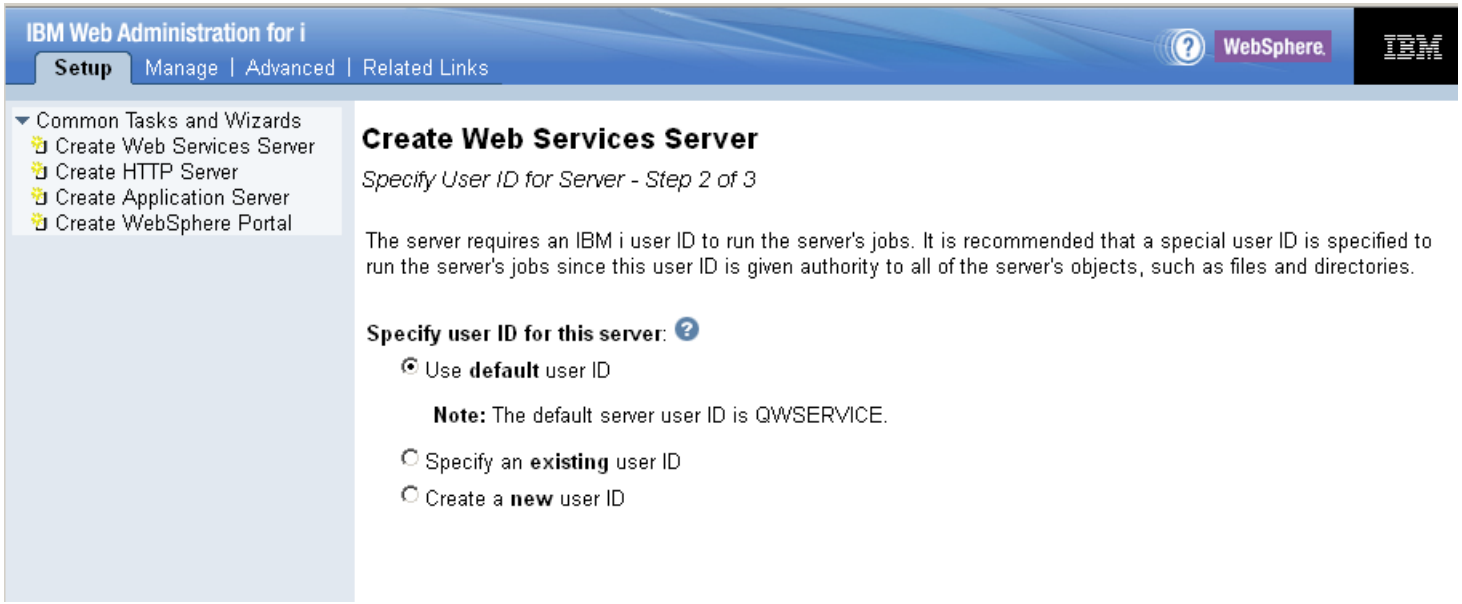
https://www.soapui.org

Create web services server (cont.)

Step 1: Specify server name.

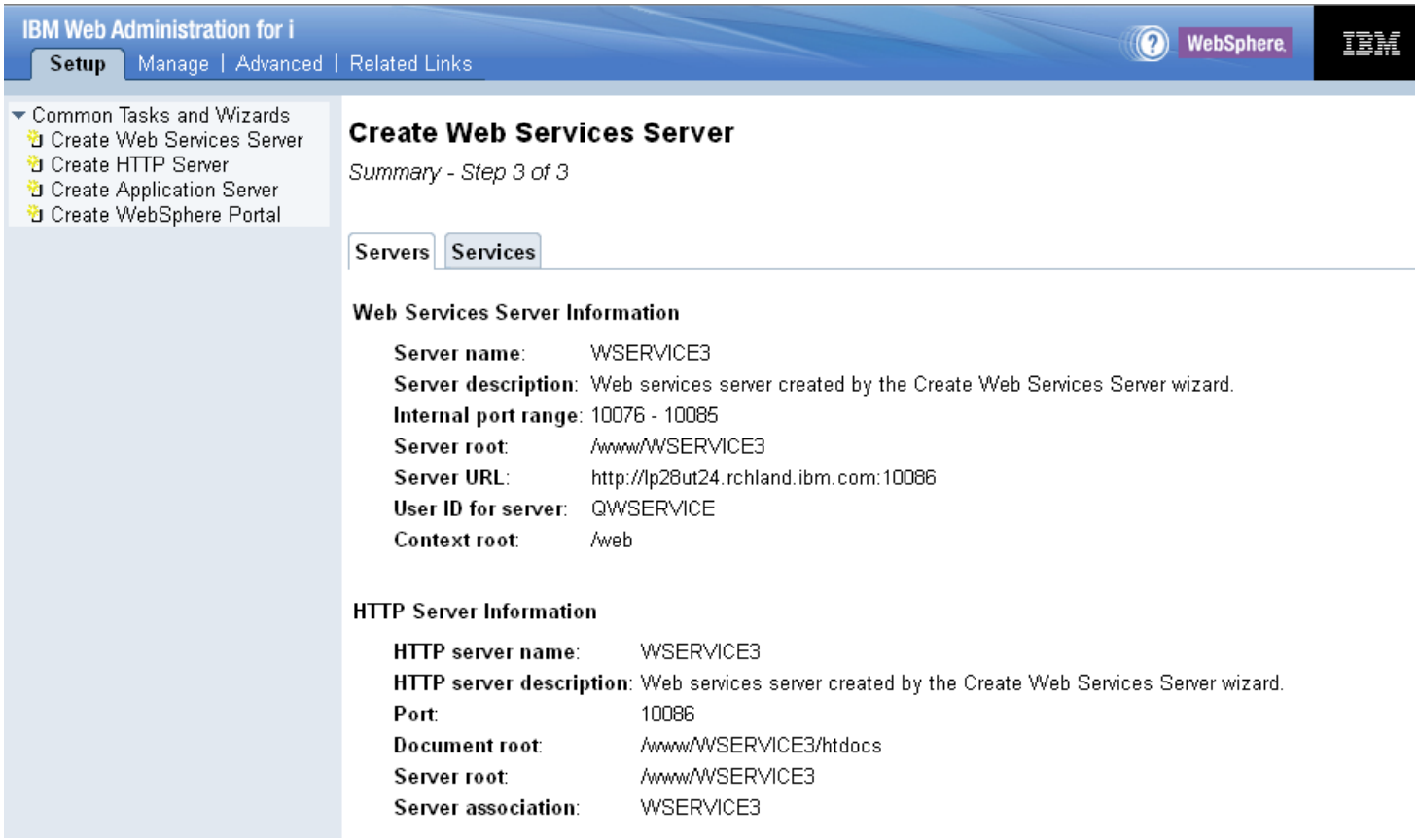
Create web services server (cont.)

Step 2: User Profile for web container.



Create web services server (cont.)

Step 3: Create the server



Create web services server (cont.)

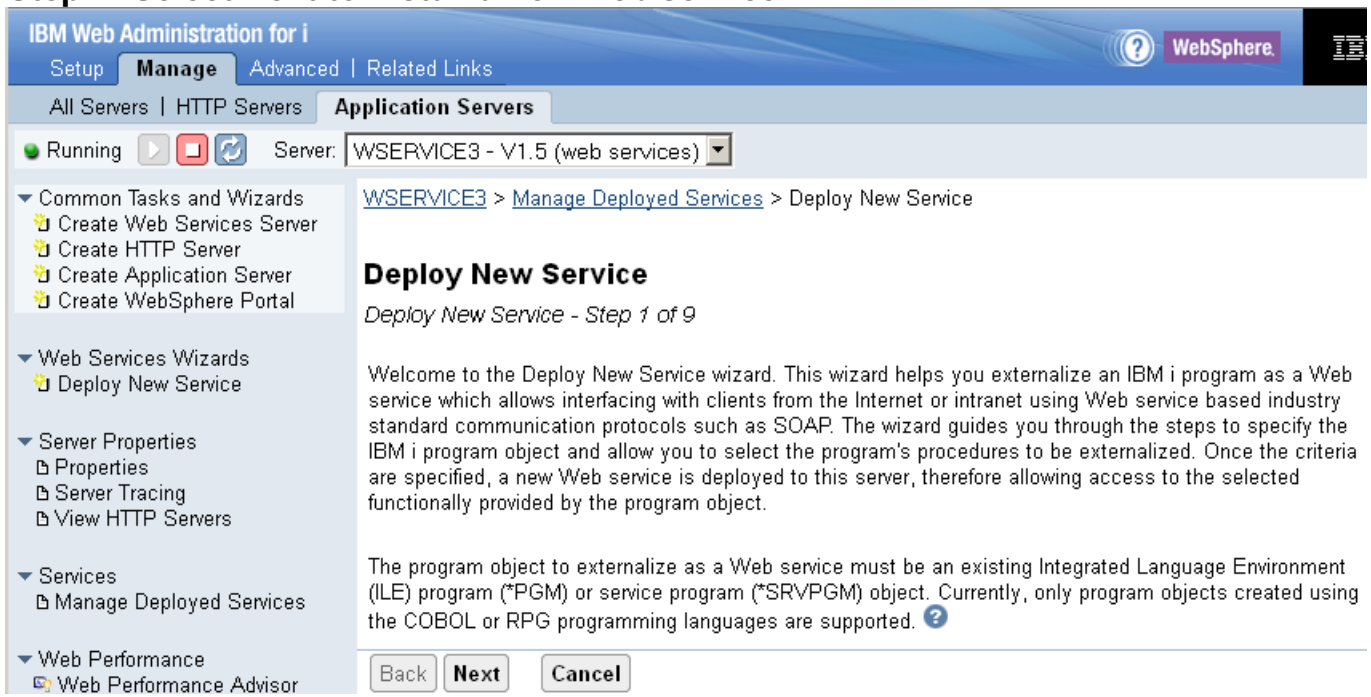
Once created, the server is started and deployed sample service started

Install web service

Select "Deploy New Service" to install a new web service

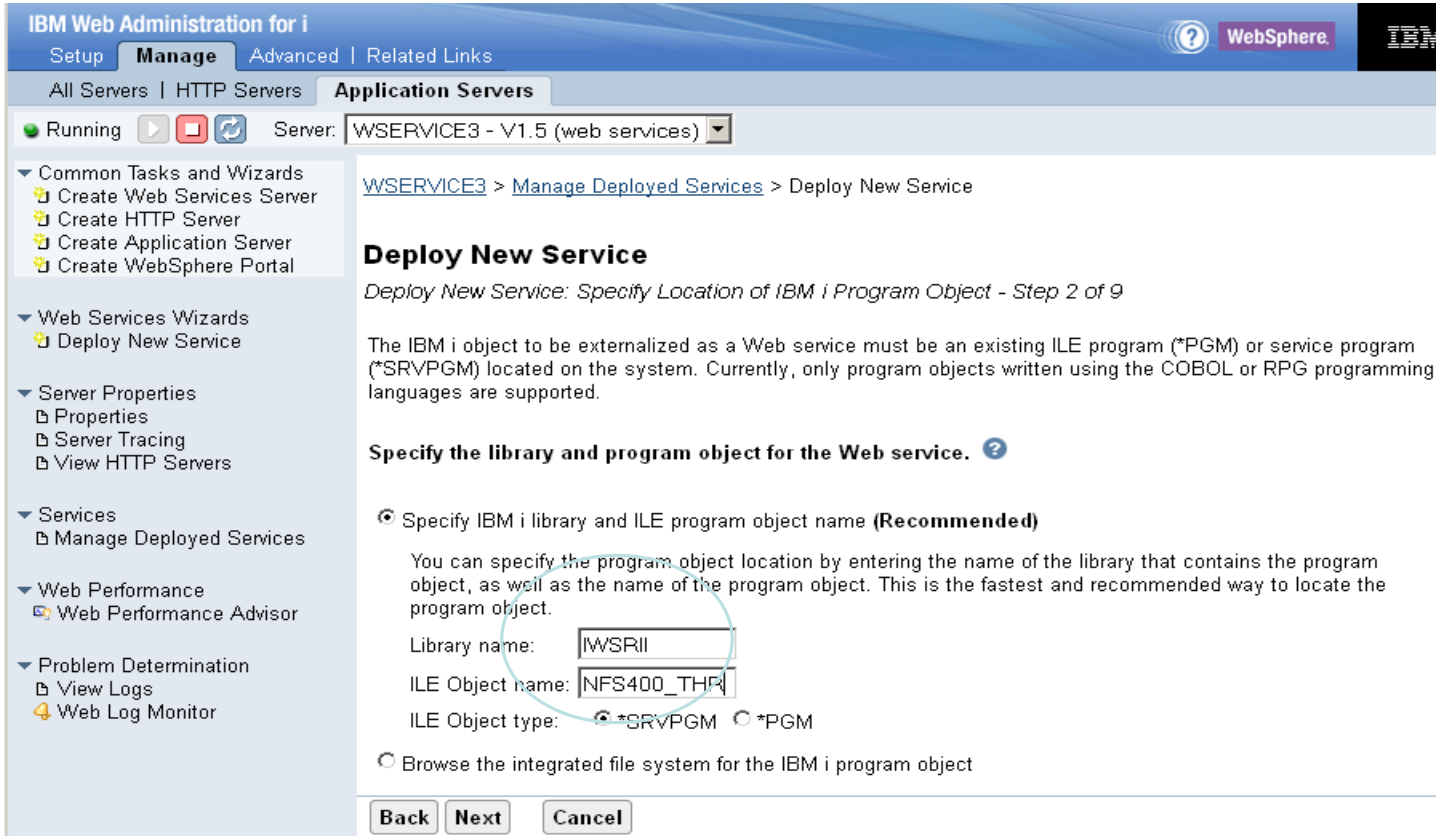
Install web service (cont.)

Step 1: Select Next to install a new web service



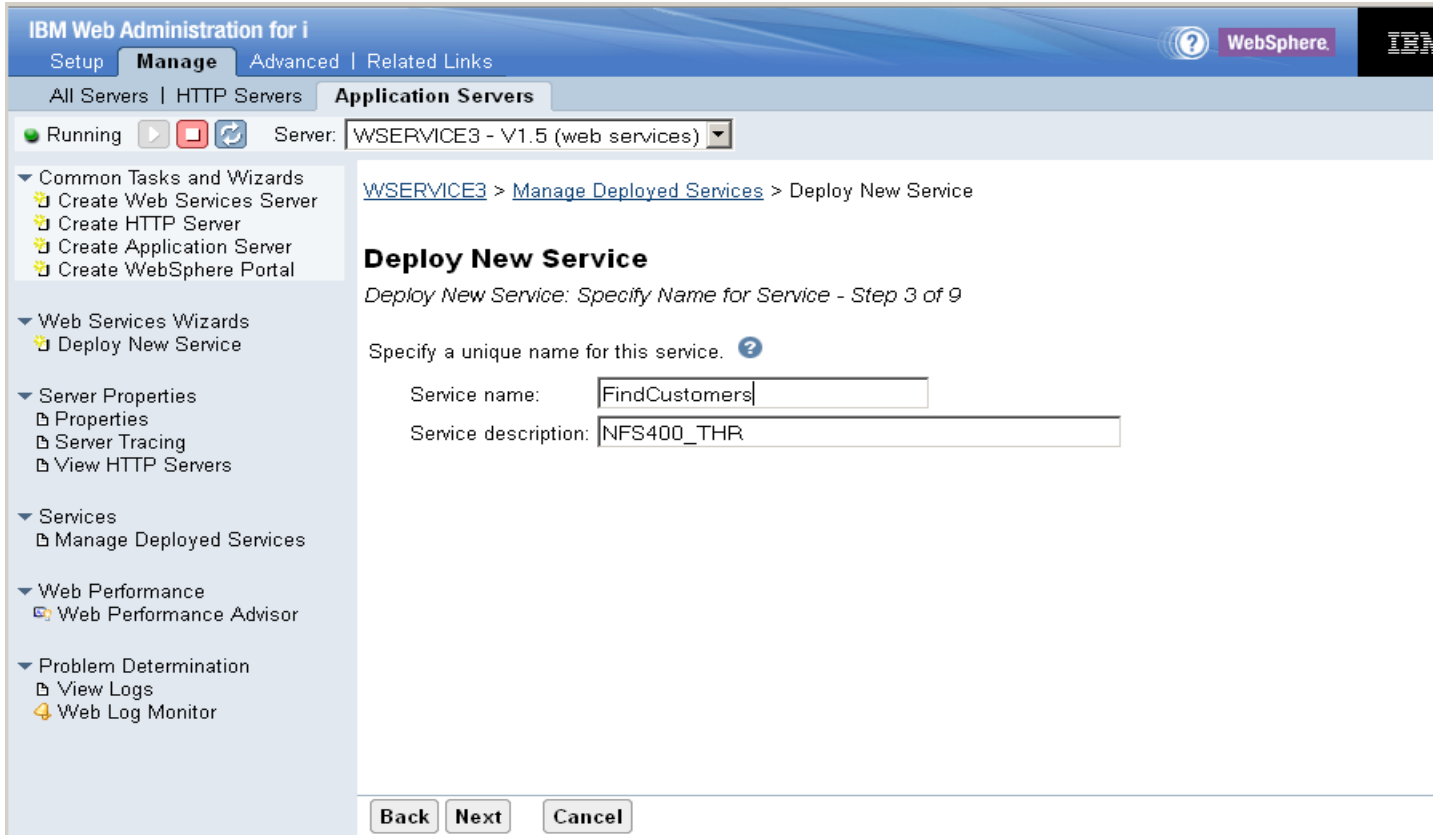
Install web service (cont.)

Step 2: What program or service program contains the web service?



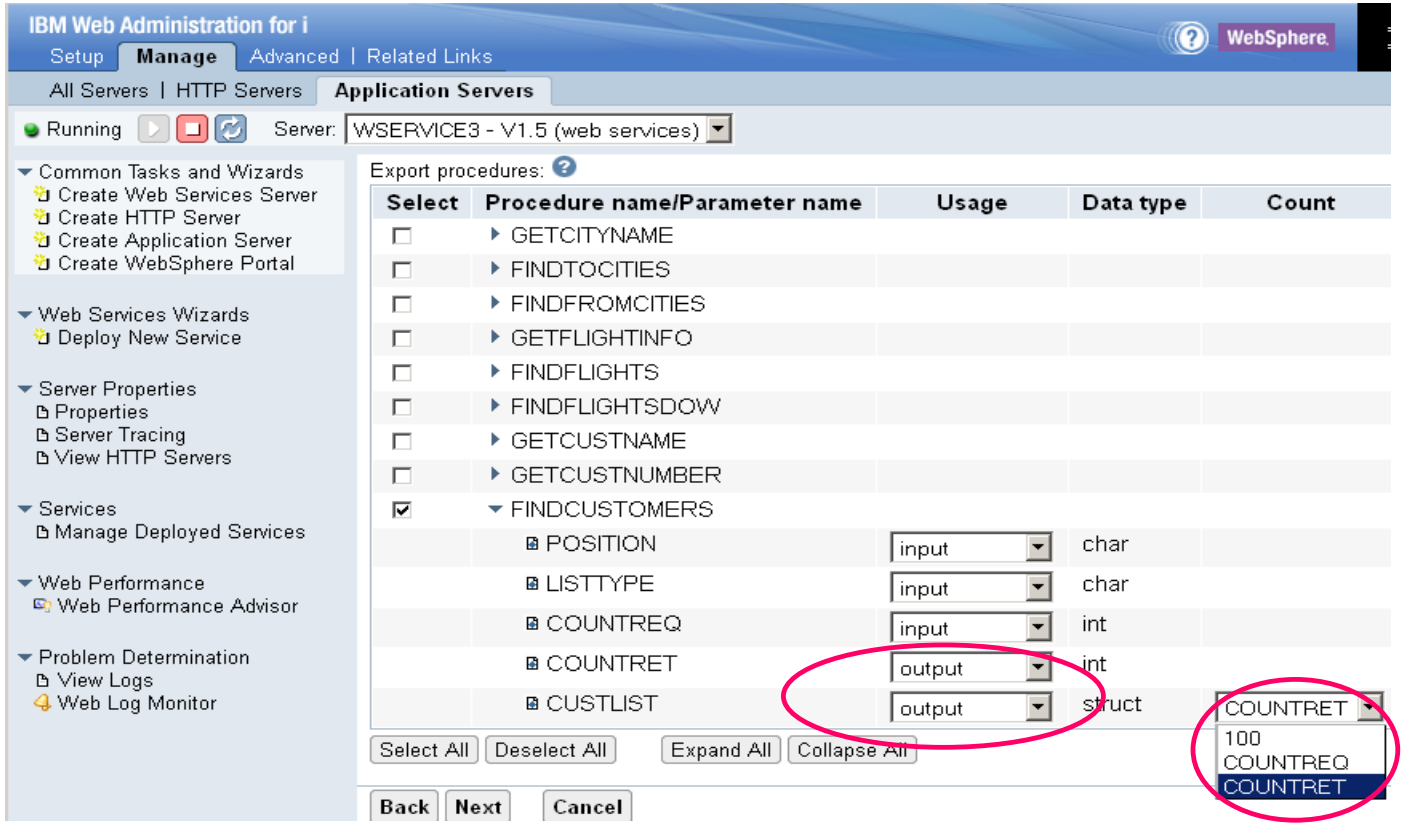
Install web service (cont.)

Step 3: What should we call this new web service?



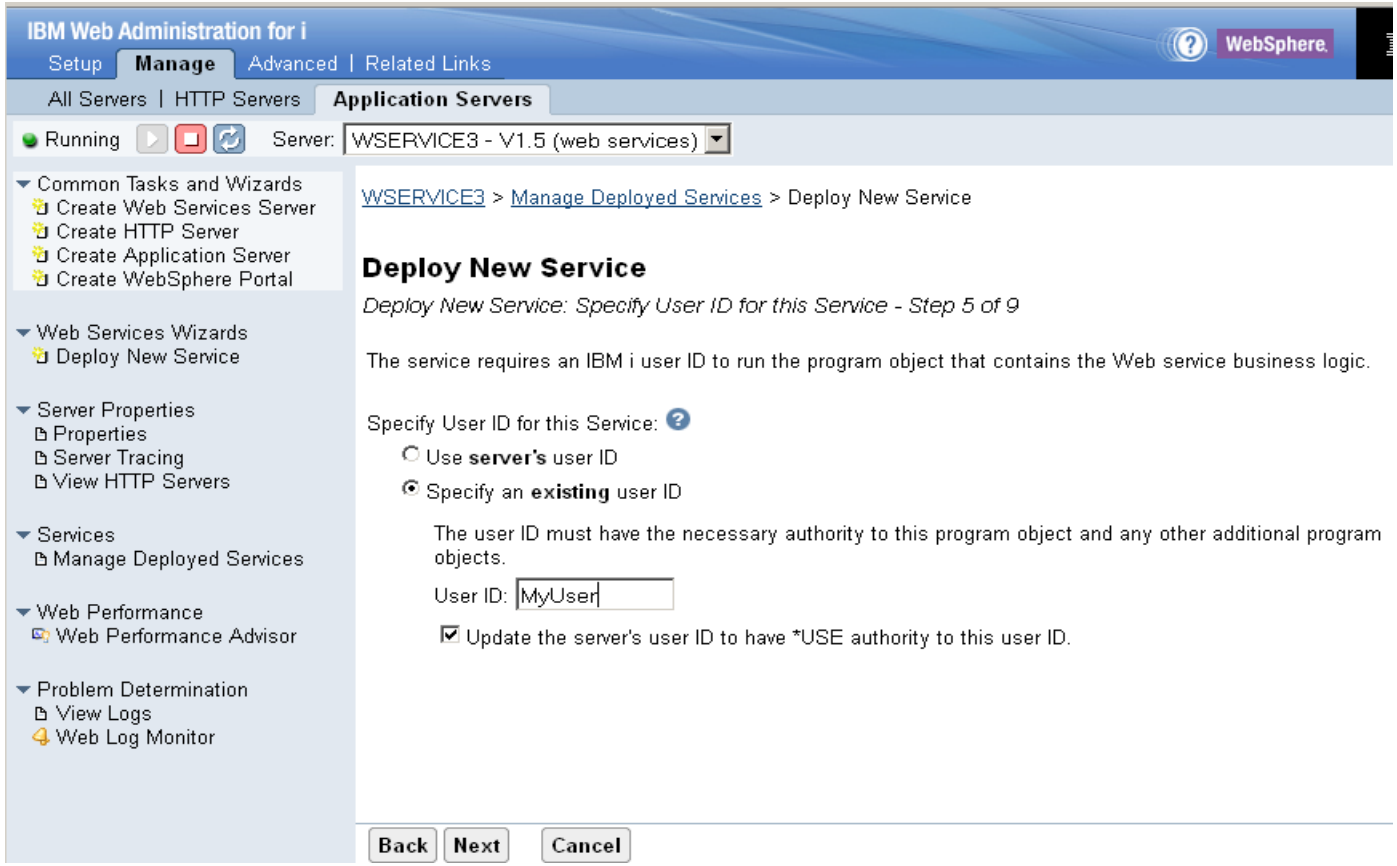
Install web service (cont.)

Step 4: What procedures should be externalized as web service operations?



Install web service (cont.)

Step 5: Specify user profile for the web service

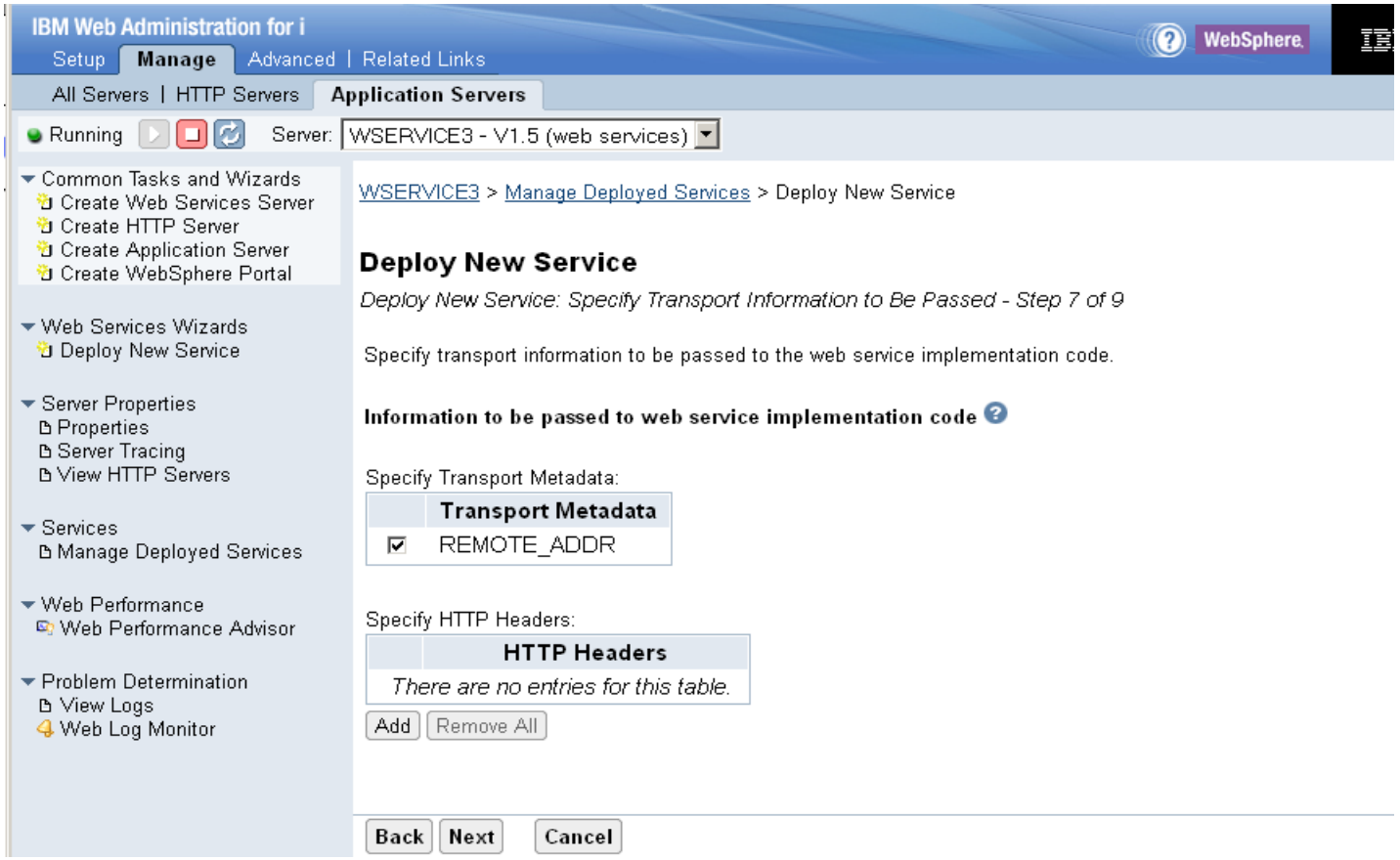


Install web service (cont.)

Step 6: Specify library list for the web service

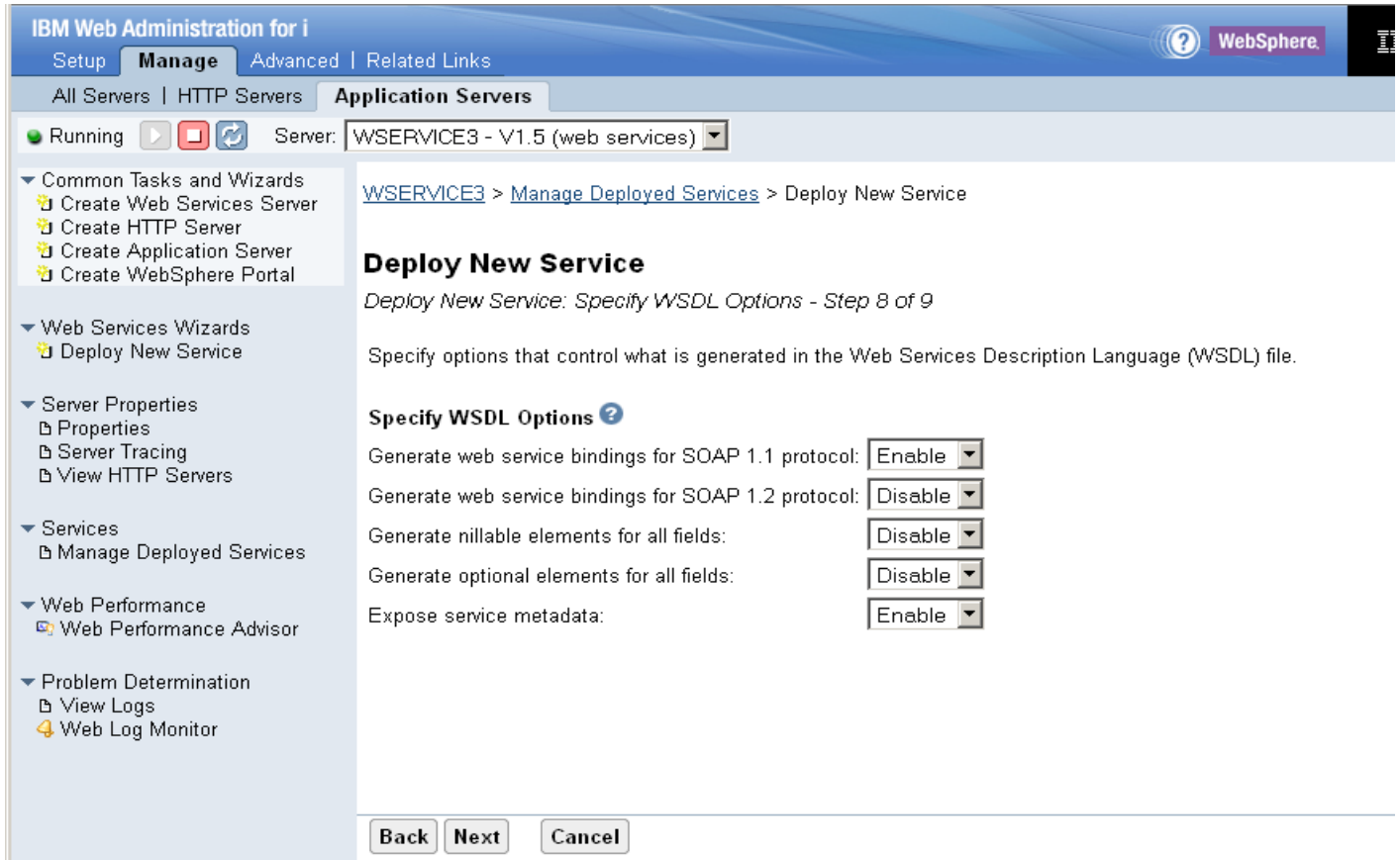
Install web service (cont.)

Step 7: Specify what request information should be passed to web service



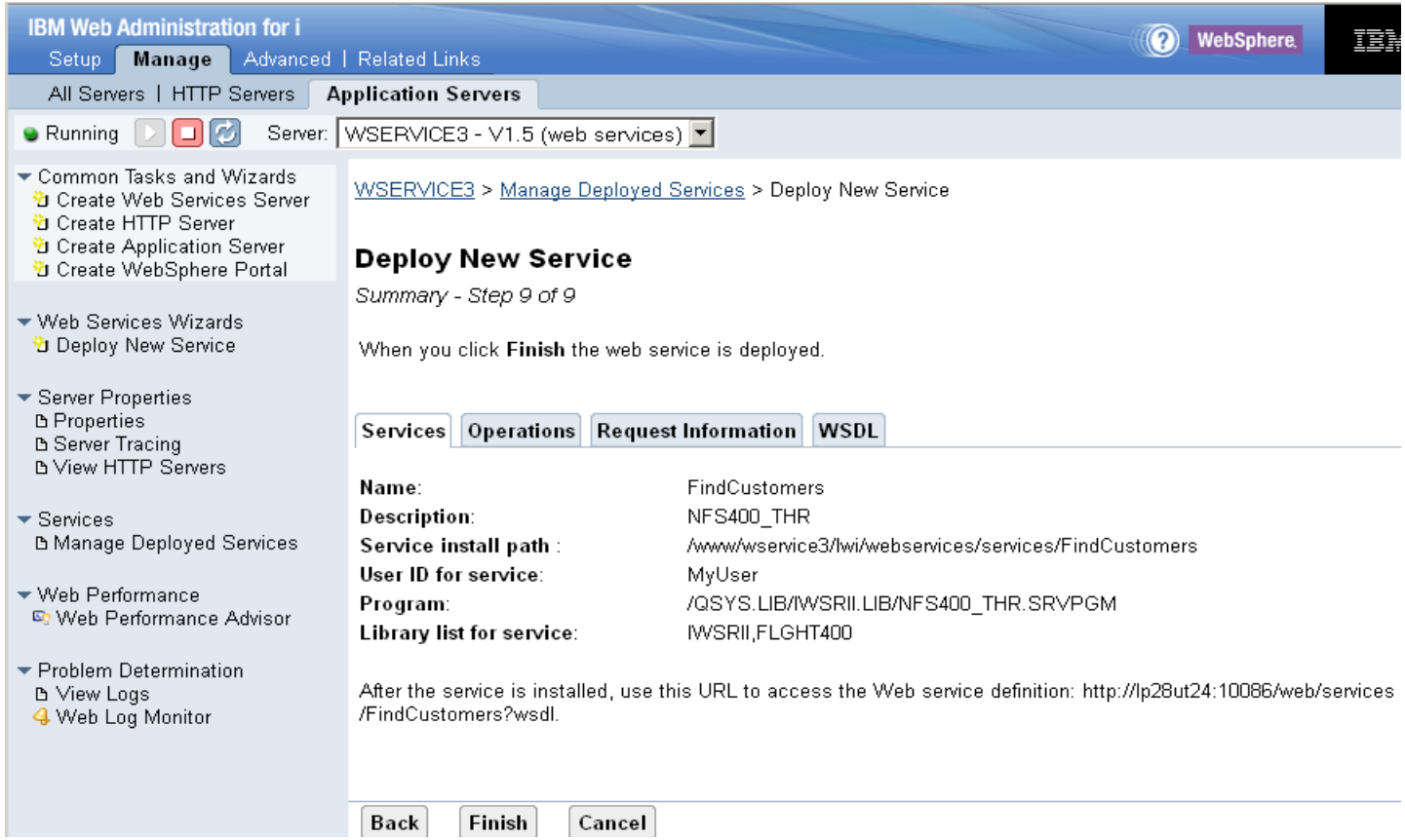
Install web service (cont.)

Step 8: Specify WSDL options for web service



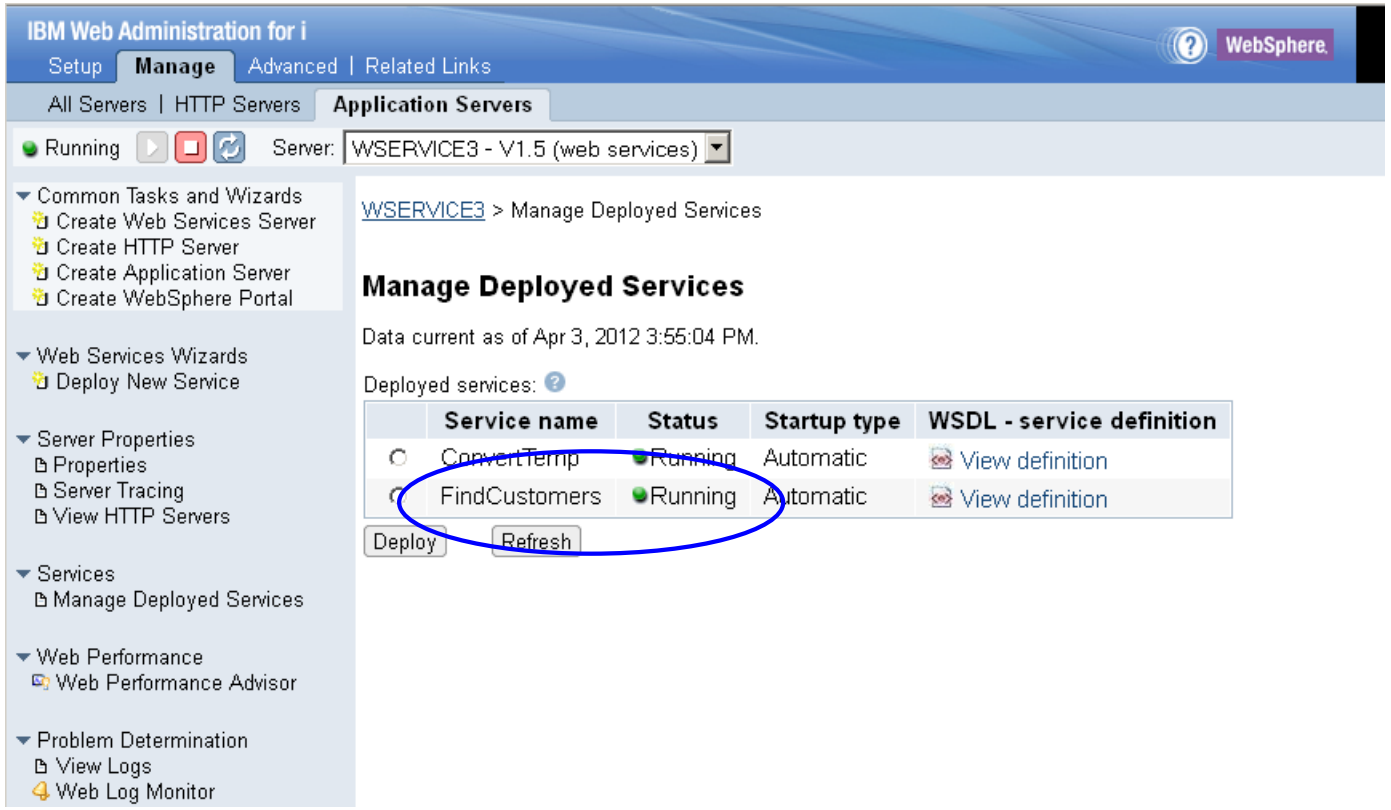
Install web service (cont.)

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



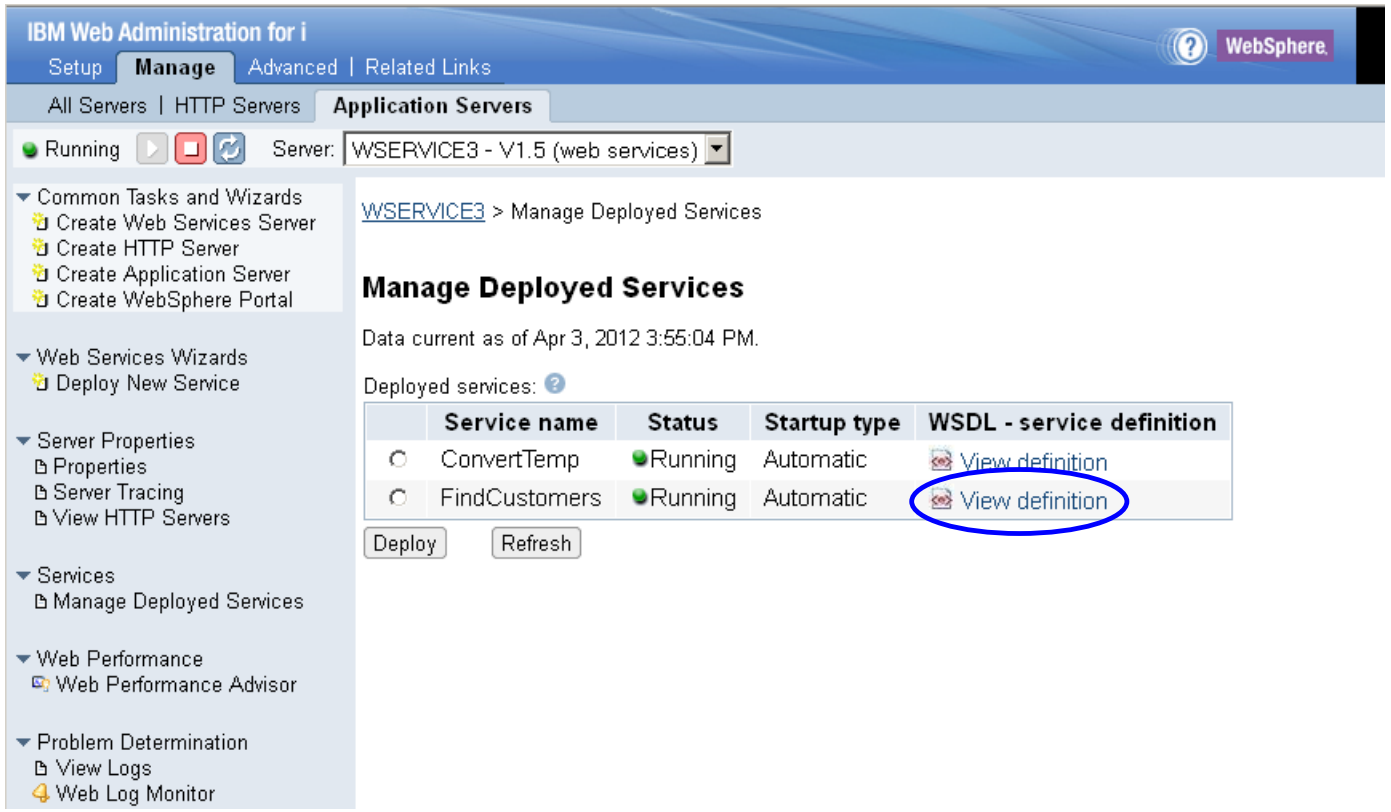
Install web service (cont.)

After a few seconds, service is installed and started



Manage the web service

You can view WSDL as long as server is active



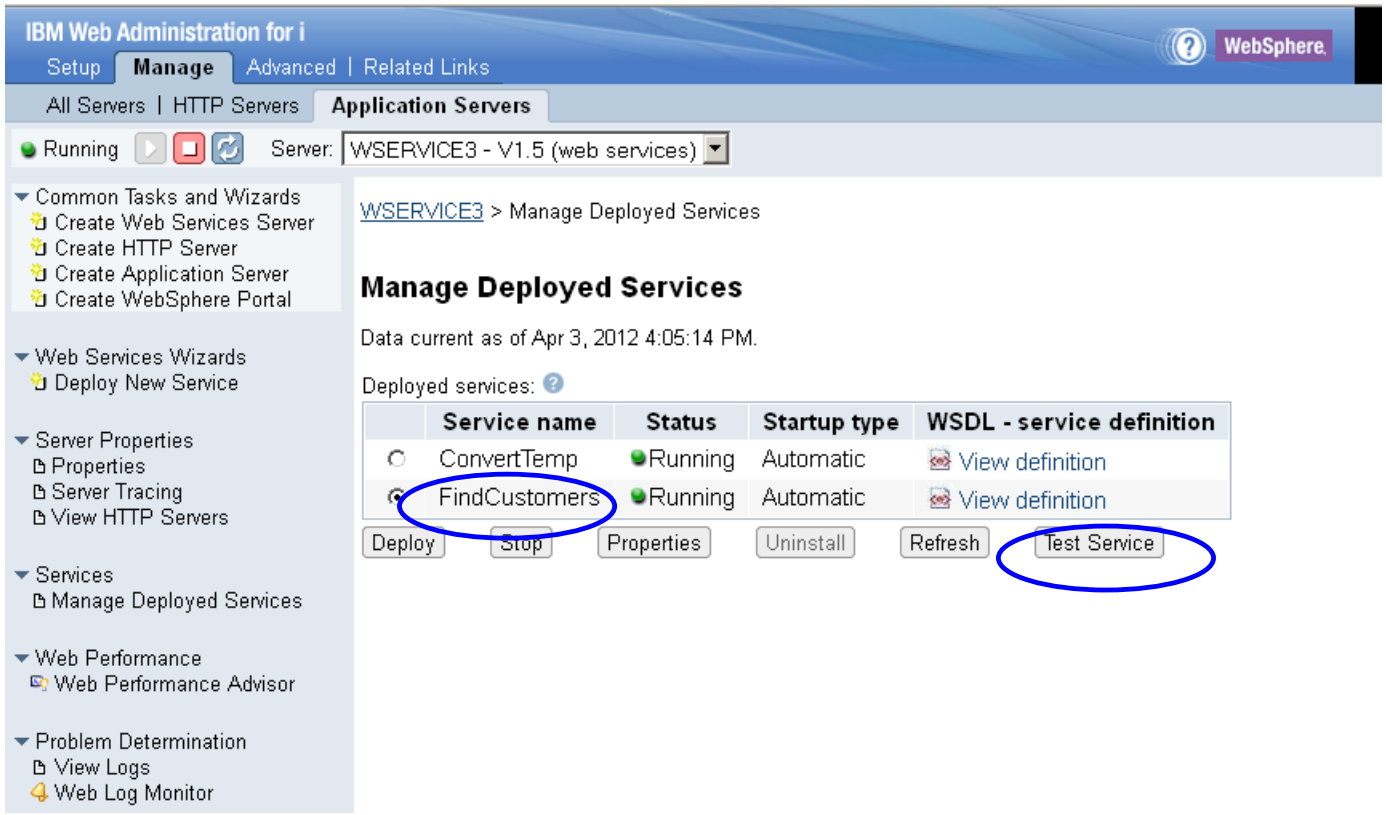
Manage the web service (cont.)

View the WSDL file (partial listing below)

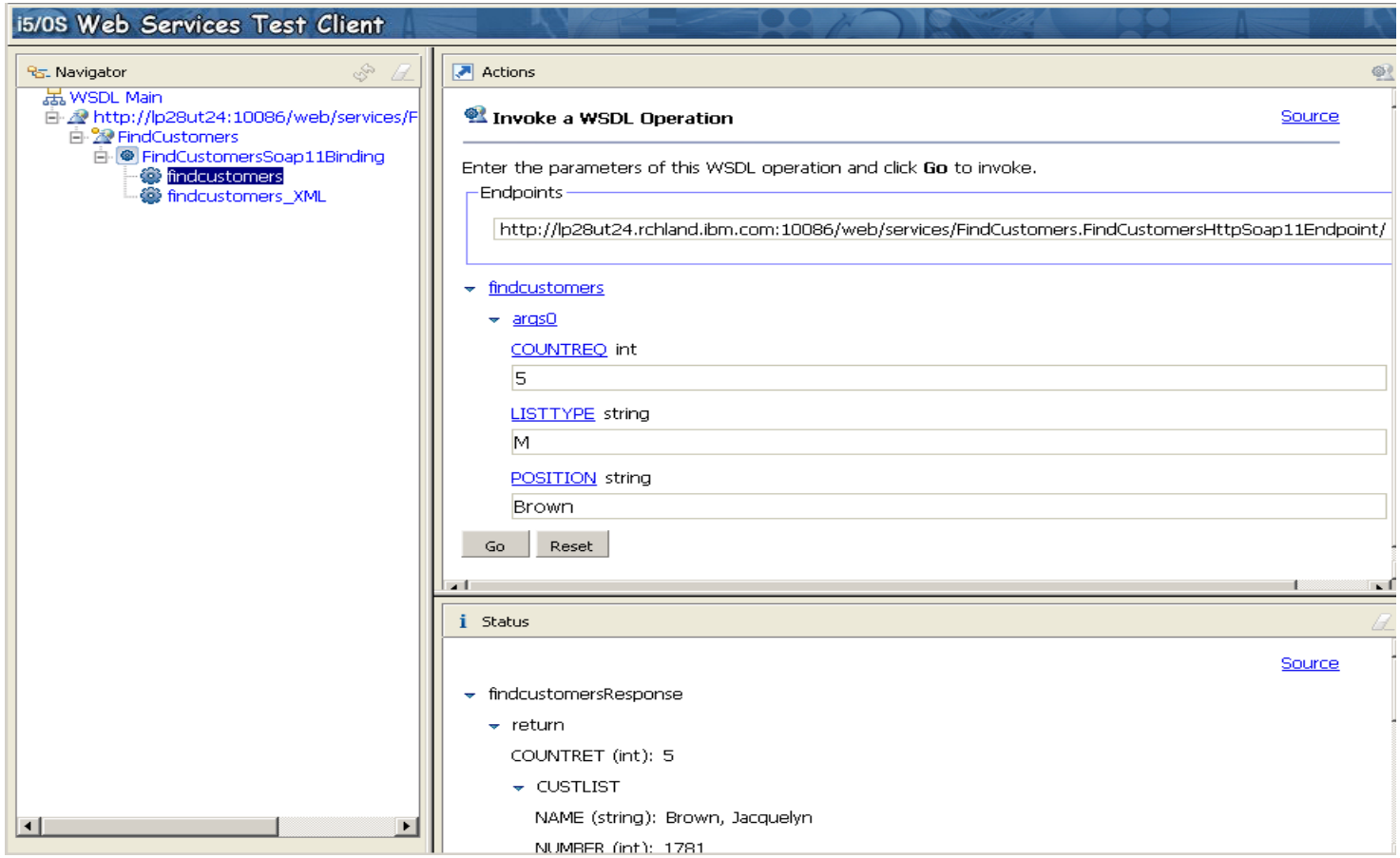
```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
    xmlns:ns="http://findcustomers.wsbeans.iseries/xsd" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
    xmlns:tns="http://findcustomers.wsbeans.iseries" xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:ns1="http://org.apache.axis2/xsd"
    xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" targetNamespace="http://findcustomers.wsbeans.iseries">
  <wsdl:documentation>FindCustomers</wsdl:documentation>
  <wsdl:types>
    <xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
               targetNamespace="http://findcustomers.wsbeans.iseries/xsd">
      <xs:element name="findcustomers_XML">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="args0" type="ns:FINDCUSTOMERSInput"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="FINDCUSTOMERSInput">
        <xs:sequence>
          <xs:element name="COUNTREQ" type="xs:int"/>
          <xs:element name="LISTTYPE" type="xs:string"/>
          <xs:element name="POSITION" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
      <xs:element name="findcustomers_XMLResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="return" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="findcustomers">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="args0" type="ns:FINDCUSTOMERSInput"/>
```

Manage the web service (cont.)

You can test the web service (but not over SSL and only for SOAP 1.1)

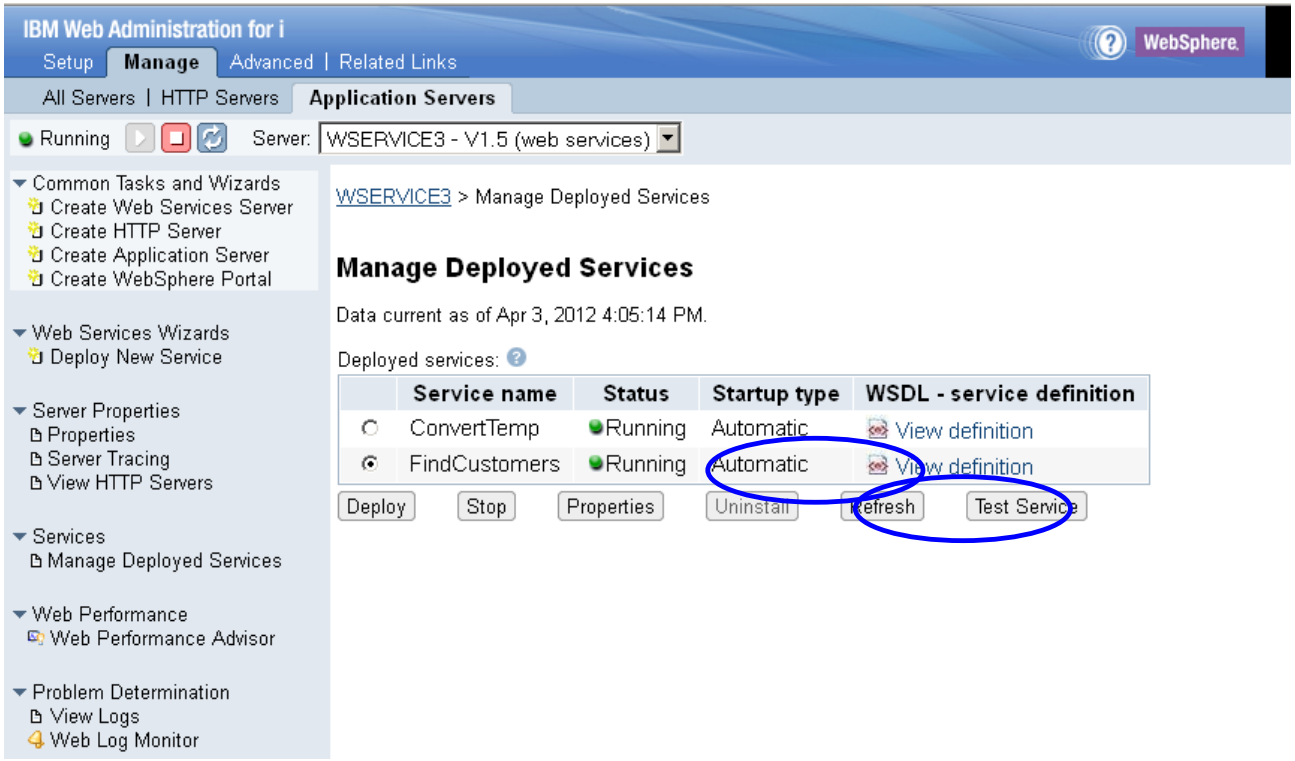


Manage the web service (cont.)



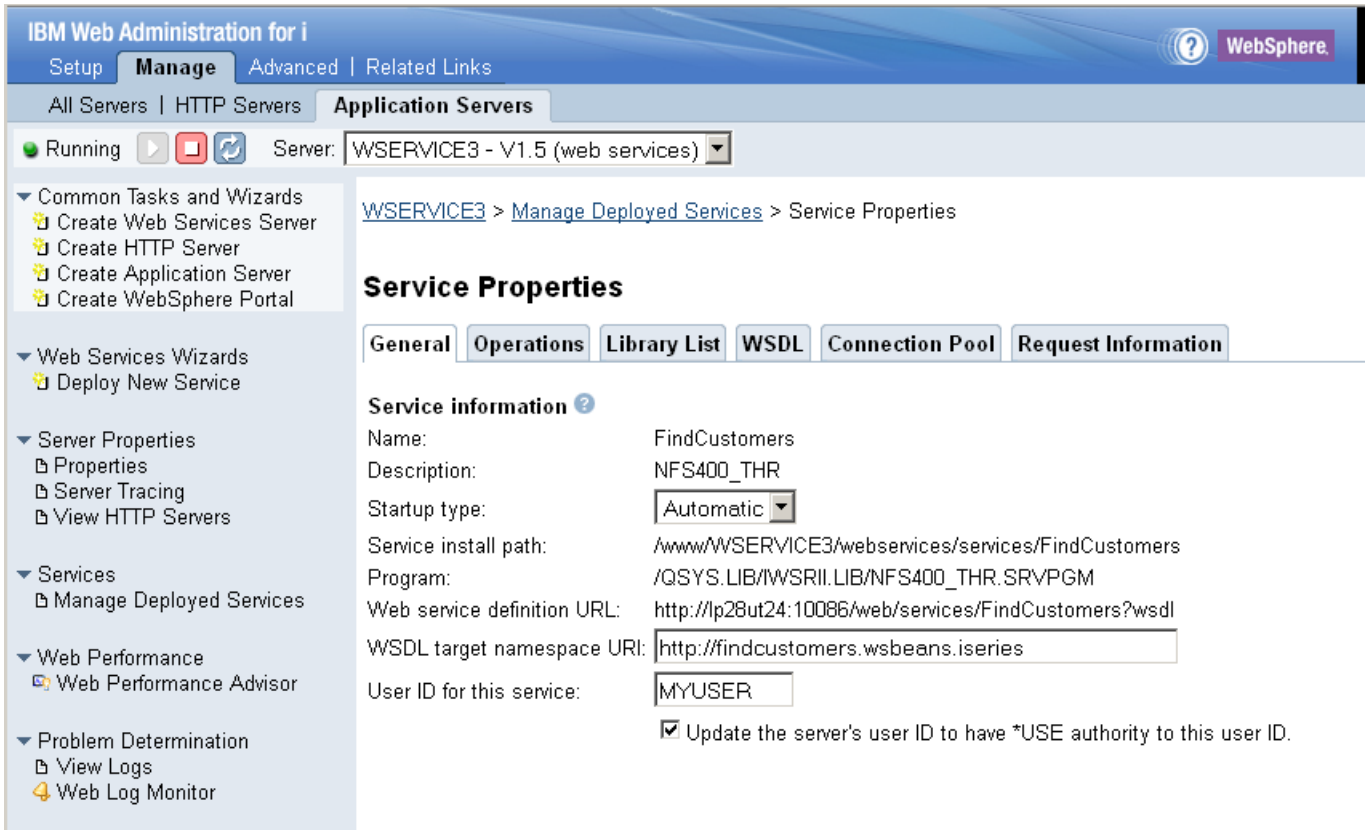
Manage the web service (cont.)

You can view and modify web service properties



Manage the web service (cont.)

Web service properties – General tab



About integrated web services server REST support

- Supported in IBM i 7.1, 7.2, & 7.3
  - On version 2.6 of integrated web services server
  - Server will handle both SOAP and REST services
- Uses JAX-RS
  - Java API for RESTful Web Services
- Two ways to deploy a REST service
  - IBM Web Administration GUI updated
    - Deploying a REST service will require more user input than when deploying a SOAP service
  - QShell script installWebService.sh updated to support REST

Best practices for REST services

- Use HTTP methods as CRUD (create/read/update/delete) operations: POST (create), GET (read), PUT (update), DELETE (delete)
- URI design matters
  - Use nouns, not verbs (/accounts/{id} not /getaccount?id=nn}
  - Predictable
  - Learn from popular APIs (Google, Facebook, Twitter, etc.)
- Keep them stateless (independent)
- Don’t send data that is not needed
- Think about cacheability
  - To improve network efficiency, scalability and user-perceived performance of you API
- Think about pagination, querying, sorting

New things to set when deploying a REST web service

- Specify the URI path to the resource (e.g. /accounts)
- For each procedure (resource method)
  a) Specify the HTTP method the procedure will handle
  b) Optionally specify URI segment path for the procedure
  c) Specify media types (e.g. XML, JSON, etc.) the procedure will accept
  d) Specify media types the procedure will return
  e) Optionally specify what values to inject in procedure input parameters
     Path segment (e.g. `/accounts/{id}`)
     Matrix parameters (e.g. `/cars;color=blue`)
     Query parameters (e.g. `/cars?color=blue`)
     Form data
     HTTP headers
     HTTP Cookies
  f) Optionally designate response code and HTTP header output parameters

Procedure and program parameter rules

- No injection to input parameters will be allowed if:
  - There is more than one input parameter that is a structure
  - There is an input parameter that is an array
  - The data type of an input parameter is something other than byte, integer, char, float, packed, or zoned
- If you want to accept JSON or XML as an input parameter, then specify an input parameter that is a structure
  - A resource request method (i.e. procedure) can accept JSON, XML, or both, assuming you indicate what media types the procedure accepts
  - A resource request method can return both types of media types, based on what the client sends on the Accept request header. For example, following example indicates that client only accepts XML responses:

```
Accept: application/xml
```

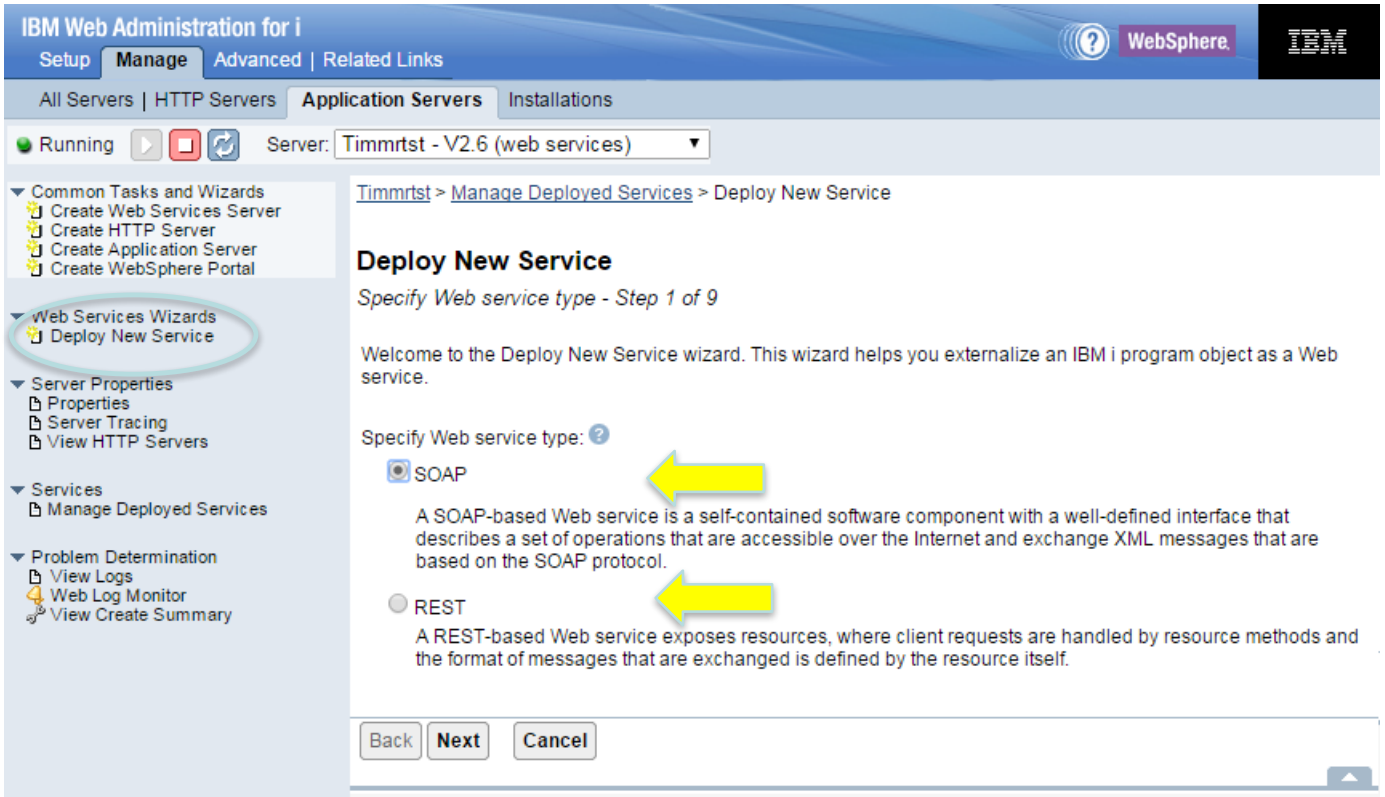
HTTP response code and headers

- A procedure output parameter with type integer can be designated as the HTTP response code parameter
  - Allows you to control what response code to return (e.g. 405 – not allowed)

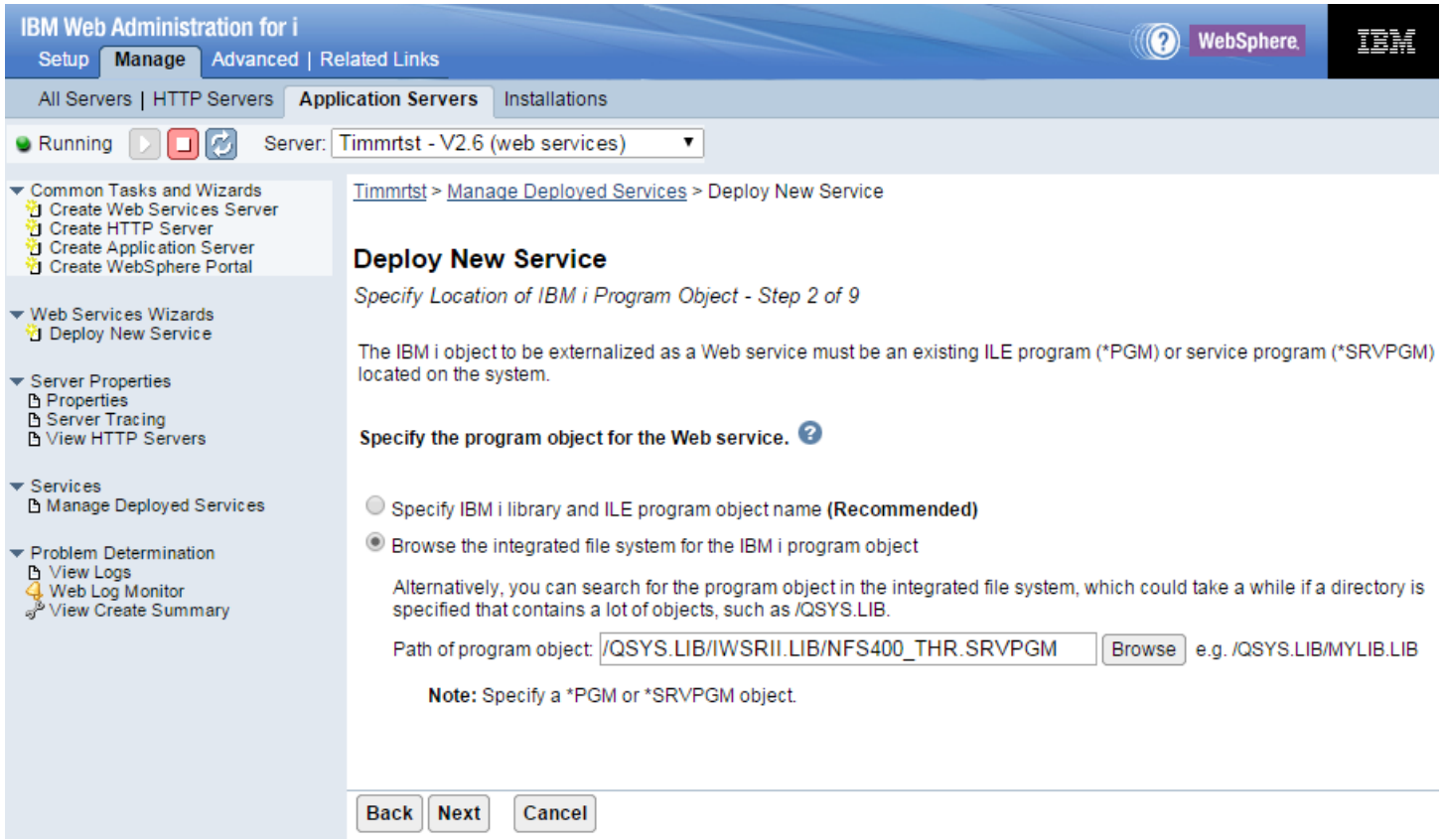
- A procedure output parameter that is an array of type char can be designated as the HTTP header parameter
  - Mainly for specifying HTTP caching headers

Demo of REST based Methodology

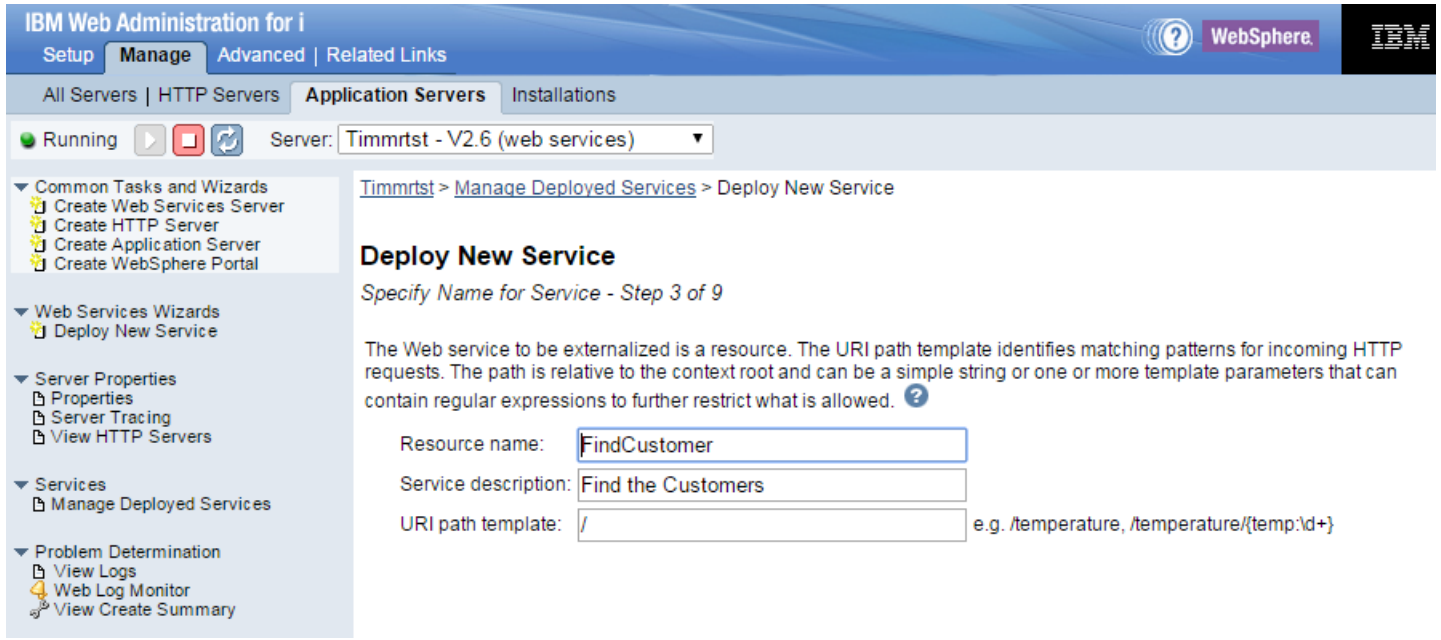
Deploy new Service – SOAP or REST



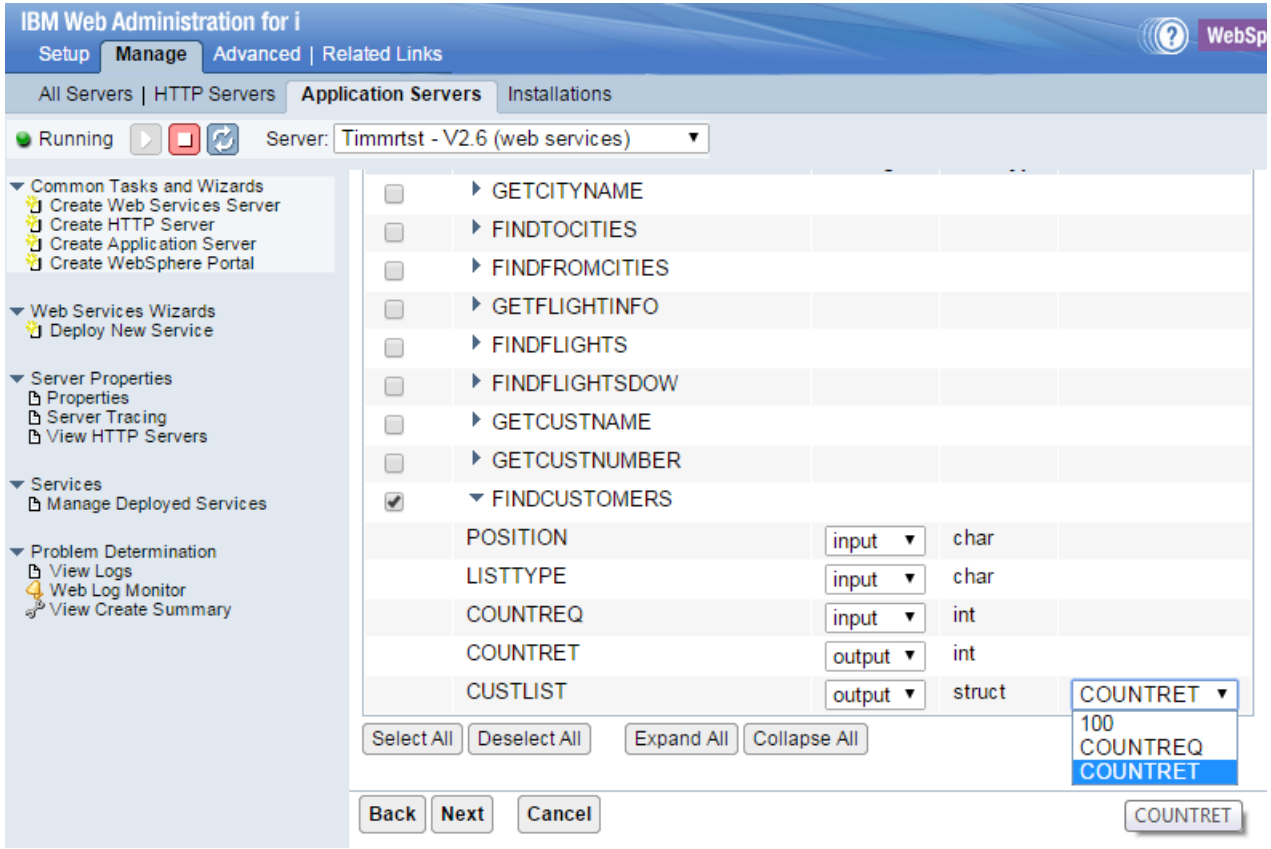
REST Service – Specify *PGM or *SVRPGM



REST – Specify Service Name



REST – Select the Export Procedures



REST – Define the Parameters

Deploy New Service

Specify Resource Method Information - Step 5 of 9

Procedures are mapped to resource methods. Each resource method needs to be defined to handle client requests by mapping an HTTP request method to a resource method.

Specify resource method information.

Procedure name: FINDCUSTOMERS
URI path template for resource: /
HTTP request method: GET
URI path template for method: *NONE
Allowed input media types: *ALL
Returned output media types: *XML_AND_JSON
HTTP response code output parameter: *NONE
HTTP header array output parameter: *NONE
Whether to wrap input parameters:

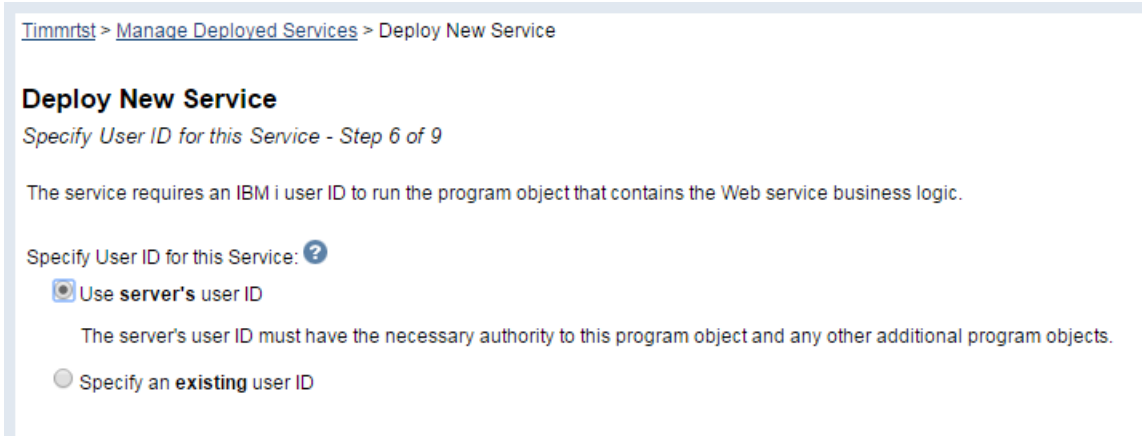
- ( ) Wrap input parameters
- (•) Do not wrap input parameters

Input parameter mappings:

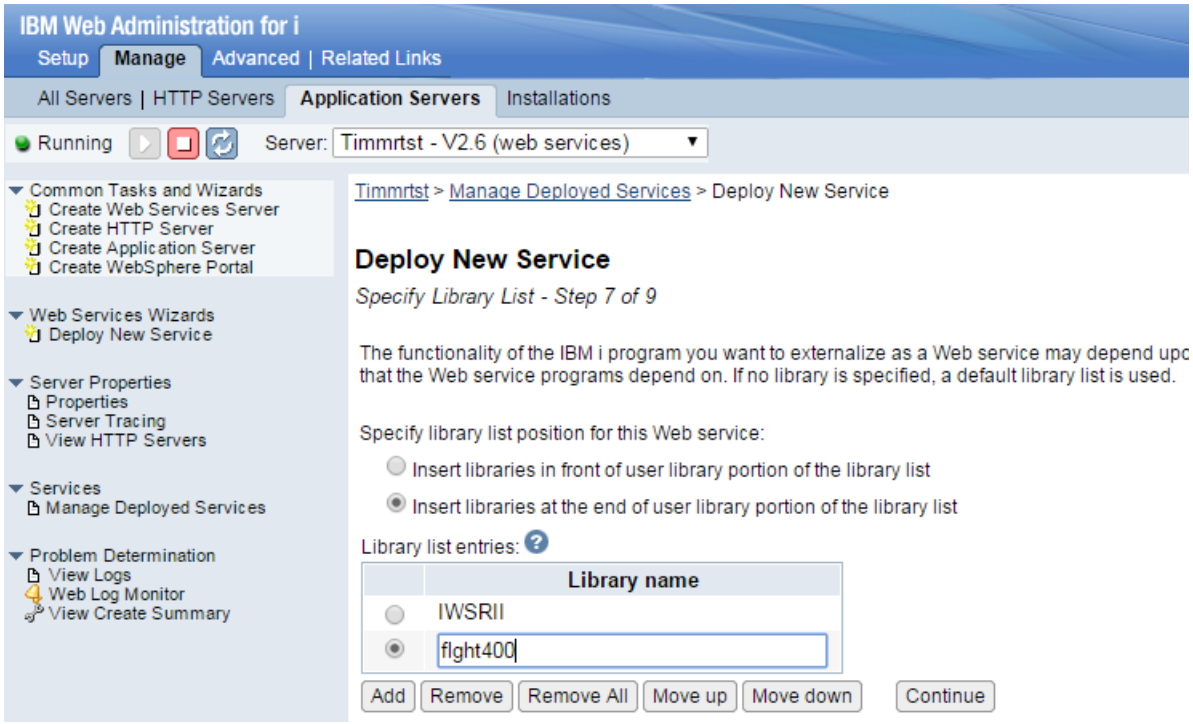
Parameter name	Data type	Input source	Identifier	Default Value
POSITION	char	*QUERY_PARAM	posistion	*NONE
LISTTYPE	char	*QUERY_PARAM	listtype	*NONE
COUNTREQ	int	*QUERY_PARAM	countreq	*NONE

Back Next Cancel

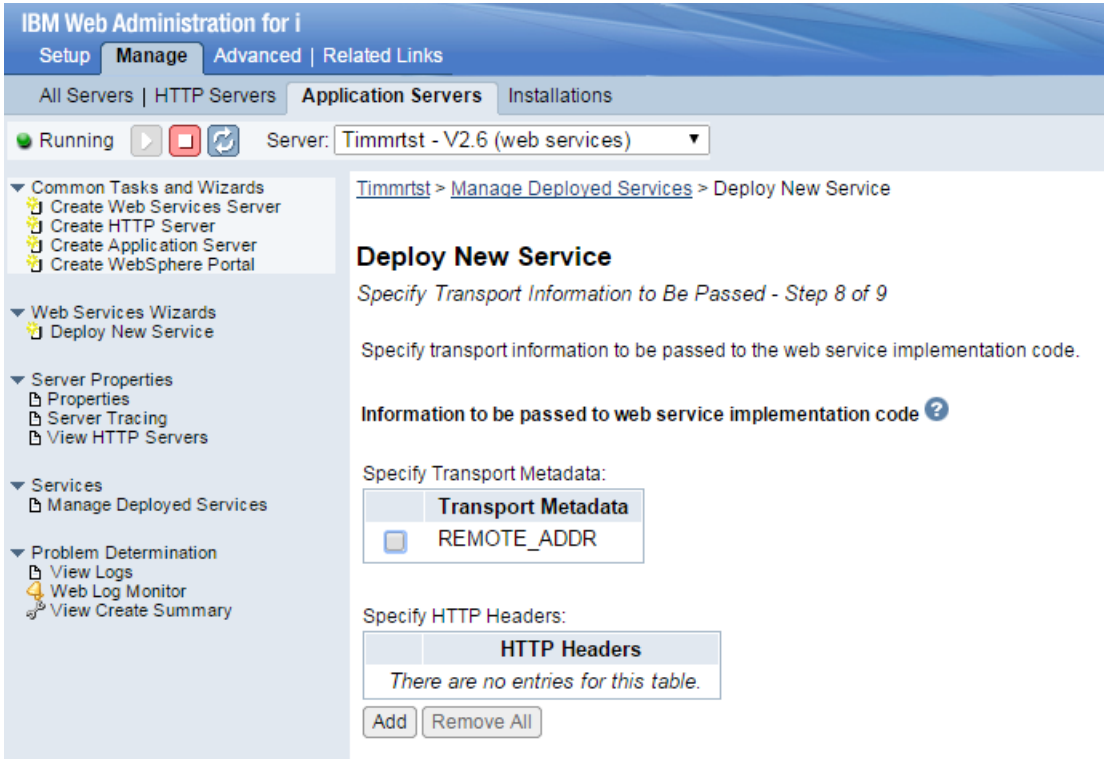
REST – Specify User Profile for the Service



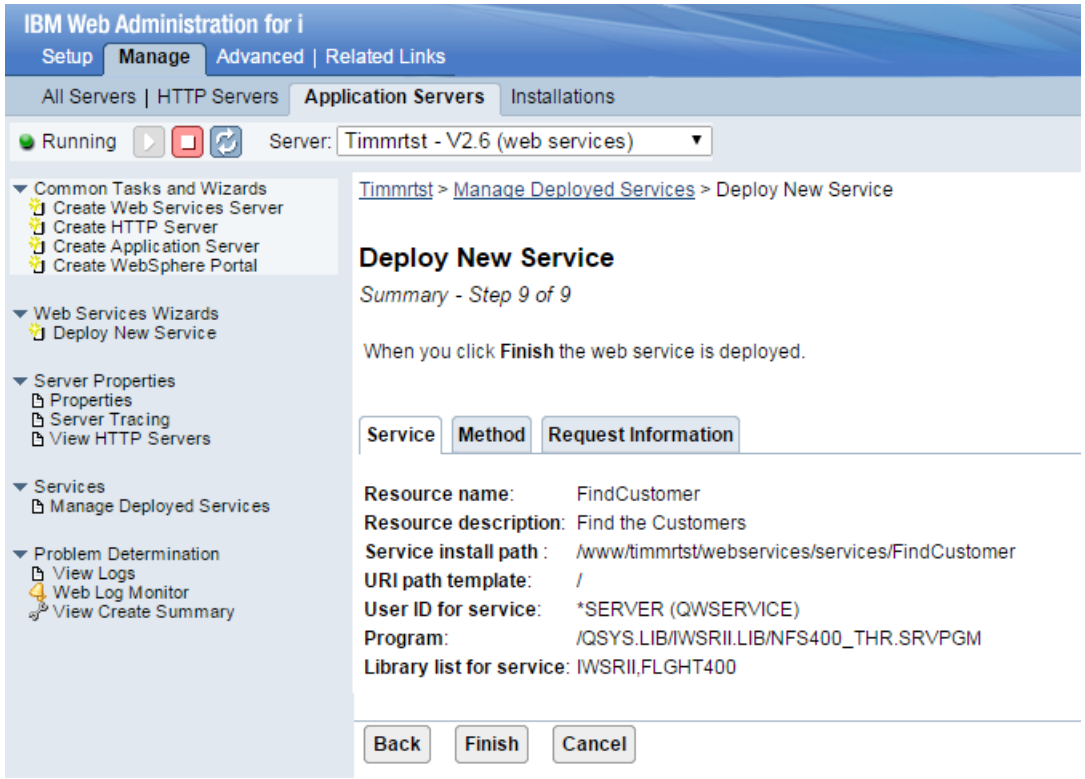
REST – Update the Library List



REST – Transport Information



REST - Finish



What have we done lately….

- 3 node support
  - HTTP on one node
  - Application Server on a node
  - Backend RPG on a node

- Use Authenticated User

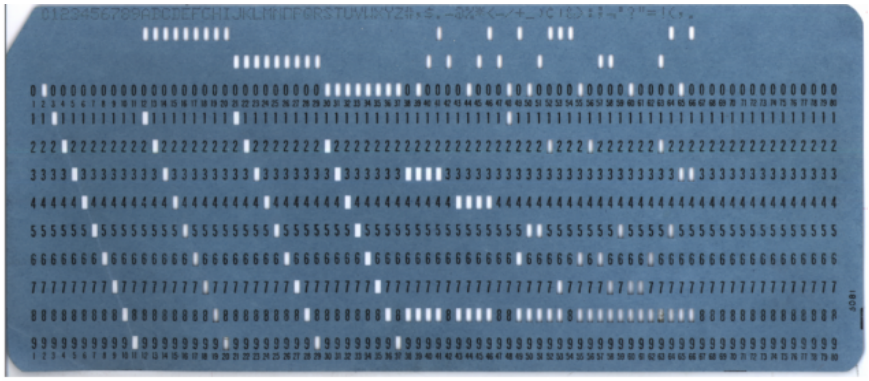
- Services re-deploy
- Connection pool pre-initialization
- Variable length fields

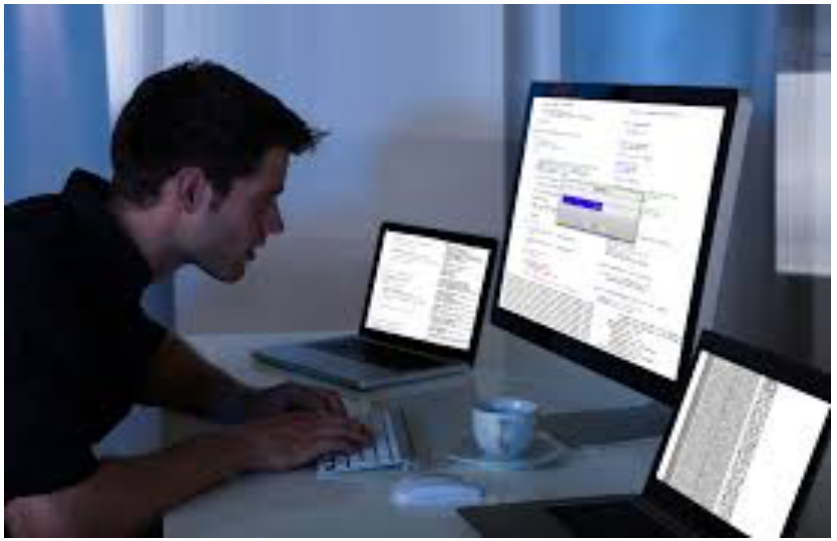
- Many other updates as requested by the community

https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/dW%20IBM%20Integrated%20Web%20Services%20for%20i


IBM
Cognitive Systems
Calling Web Services from IBM i
Tim Rowe - Business Architect for Application Development
timmr@us.ibm.com
for Business

RPG - Where to we start....





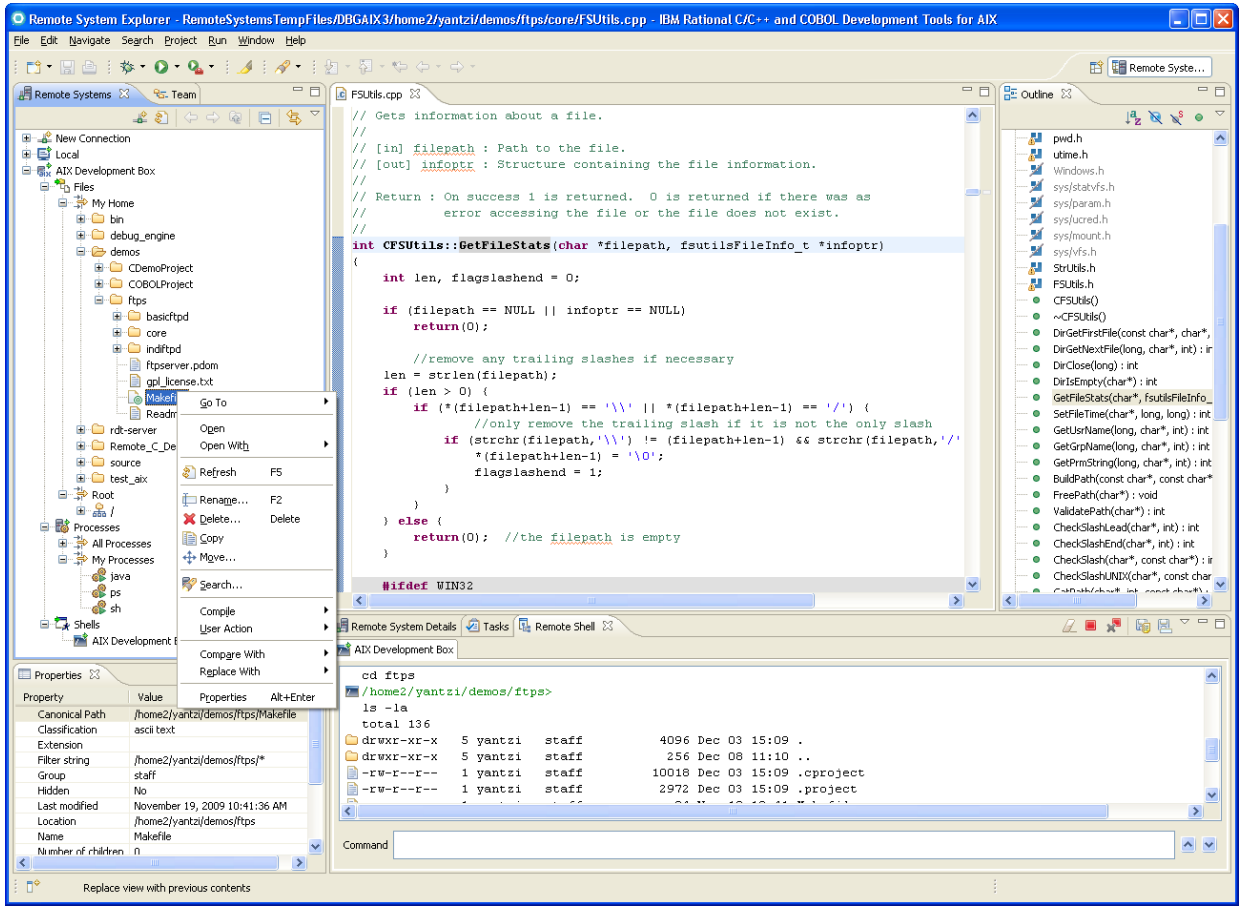
RPG - A Modern Business Language

- Interoperability
  - Java
  - XML
  - SQL
- Readability
  - Free form
  - Blank lines
  - Comments
- Functionality
  - Procedures
  - Data areas
  - Data structures
  - More data types
  - Extended file support
- Modern Tools
  - RD i, RTC, ARCAD Power Pack

```
    read  file;                 // Get next record
    dow not %eof(file);         // Keep looping with record
        if %error;
            dsply 'The read failed';
            leave;
        else;
            chain(n) name database data;
            time = hours * num_employees
                      + overtime_saved;
            pos = %scan (',' : name);
            name = %xlate(upper : lower : name);
            exsr handle_record;
            read file;
        endif;
    enddo;
begsr handle_record;
    eval(h) time = time + total_hours_array (empno);
    temp_hours = total_hours - excess_hours;
    record_transaction();
endsr;
```

Rational Developer for i

- *Modern*
- *Integrated*
- Analysis
- Debugger
- Visual
- Supports RPG, COBOL, CL, C, C++, SQL, DDM



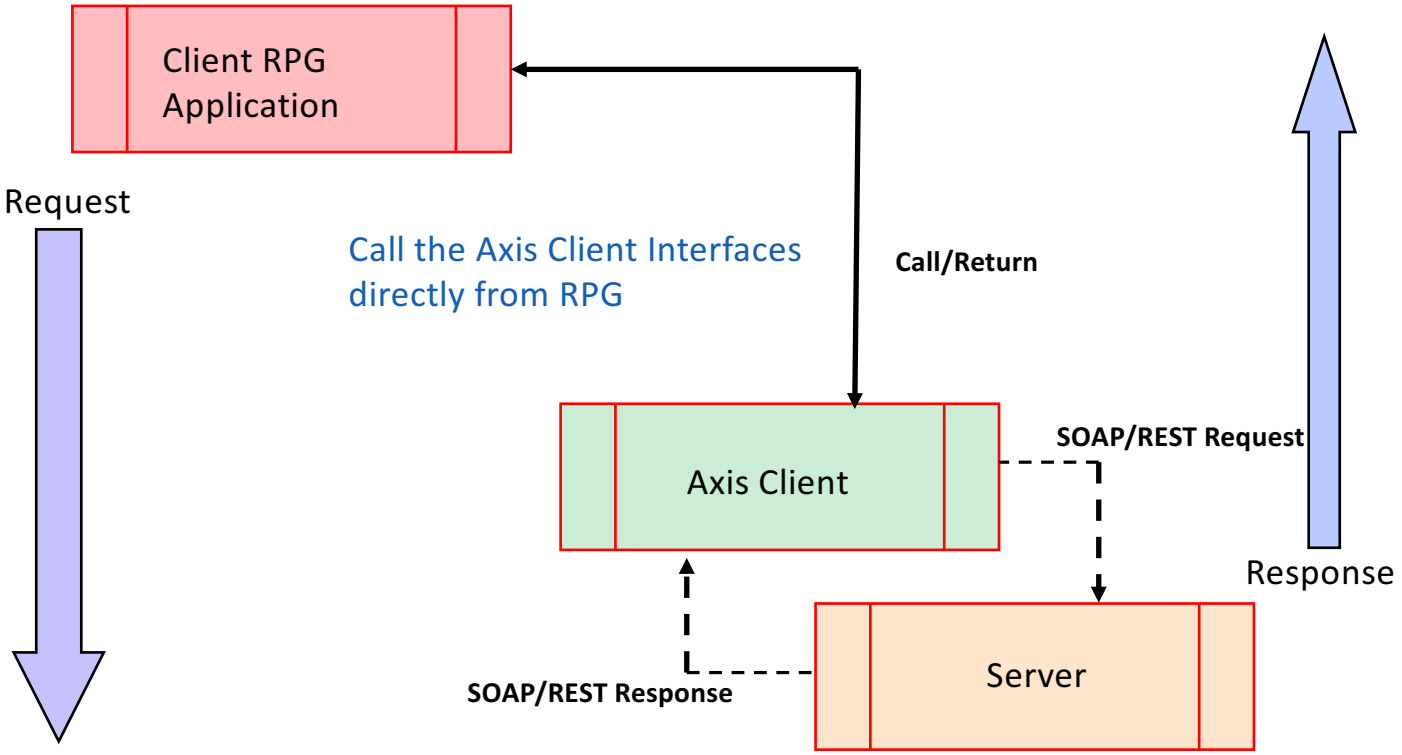
How do you connect RPG to Watson

- Many different ways….
  - SQL
  - Integrated Web Services Client
  - Call another language that speaks ‘REST’
    - Java
    - Node.js
    - Python

Integrated Web Services Client

- Overview
  - Based on Apache AXIS C++
  - Consists
    - Tool to convert WSDL to RPG/C/C++ stubs (SOAP only support)
    - SOAP Client
    - REST Client
  - Availability
    - IBM i SS1 Option 3
    - **Latest HTTP group PTF**
- Supports
  - C, C++, RPG, COBOL

Integrated Web Services Client



RPG Calling Axis Client

```
str1 = 'https://watson-api-explorer.mybluemix.net/' +
       'language-translator/api/v2/translate?model_id=' +
  fromLang + '-' + toLang + '&text=' +  str1 + X'00';


   // Create HTTP transport handle.
   tHandle = axiscTransportCreate(str1:AXISC_PROTOCOL_HTTP11);

   // Set SSL information - turn off SSLv2, SSLv3, TLSv1 and tolerate
   // certificate not being in key store
   NONE = 'NONE' + X'00';
   propBuf1 = '/QIBM/USERDATA/ICSS/CERT/SERVER/DEFAULT.KDB' + X'00';
   propBuf2 = 'true' + X'00';

   axiscTransportSetProperty(tHandle: AXISC_PROPERTY_HTTP_SSL:
          %addr(propBuf1):
          %addr(NULLSTR): %addr(NULLSTR):
          %addr(NONE)  : %addr(NONE):
          %addr(NONE)  : %addr(NULLSTR): %addr(NULLSTR):
          %addr(propBuf2));
```

RPG Calling Axis Client

```
// Indicate that the payload in response should stay in UTF-8
    propBuf2 ='false'+X'00';
    axiscTransportSetProperty(tHandle: AXISC_PROPERTY_CONVERT_PAYLOAD:
          %addr(propBuf2));

    // Flush transport so request is sent and receive response.
    rc = axiscTransportFlush(tHandle);
 if (rc = 0);
       rc = axiscTransportReceive(tHandle:
       %ADDR(response): %SIZE(response): 0);

       dow rc > 0 AND bytesRead < %SIZE(response);
    bytesRead = bytesRead + rc;
          rc = axiscTransportReceive(tHandle:
         %ADDR(response)+bytesRead:
         %SIZE(response)-bytesRead: 0);
  enddo;
endif;
```

Some Additional Light Reading

Developer Works – 3 Part Series on Rest for IBM i

- https://www.ibm.com/developerworks/ibmi/library/i-rest-web-services-server1/
- http://www.ibm.com/developerworks/ibmi/library/i-rest-web-services-server2/
- https://www.ibm.com/developerworks/ibmi/library/i-rest-web-services-server3/

Questions and Answers

Special notices

This document was developed for IBM offerings in the United States as of the date of publication. IBM may not make these offerings available in other countries, and the information is subject to change without notice. Consult your local IBM business contact for information on the IBM offerings available in your area.

Information in this document concerning non-IBM products was obtained from the suppliers of these products or other public sources. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. Send license inquires, in writing, to IBM Director of Licensing, IBM Corporation, New Castle Drive, Armonk, NY 10504-1785 USA.

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

The information contained in this document has not been submitted to any formal IBM test and is provided "AS IS" with no warranties or guarantees either expressed or implied.

All examples cited or described in this document are presented as illustrations of the manner in which some IBM products can be used and the results that may be achieved. Actual environmental costs and performance characteristics will vary depending on individual client configurations and conditions.

IBM Global Financing offerings are provided through IBM Credit Corporation in the United States and other IBM subsidiaries and divisions worldwide to qualified commercial and government clients. Rates are based on a client's credit rating, financing terms, offering type, equipment type and options, and may vary by country. Other restrictions may apply. Rates and offerings are subject to change, extension or withdrawal without notice.

IBM is not responsible for printing errors in this document that result in pricing or information inaccuracies.

All prices shown are IBM's United States suggested list prices and are subject to change without notice; reseller prices may vary.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

Any performance data contained in this document was determined in a controlled environment. Actual results may vary significantly and are dependent on many factors including system hardware configuration and software design and configuration. Some measurements quoted in this document may have been made on development-level systems. There is no guarantee these measurements will be the same on generally-available systems. Some measurements quoted in this document may have been estimated through extrapolation. Users of this document should verify the applicable data for their specific environment.

Revised September 26, 2006

Special notices (cont.)

IBM, the IBM logo, ibm.com AIX, AIX (logo), AIX 5L, AIX 6 (logo), AS/400, BladeCenter, Blue Gene, ClusterProven, DB2, ESCON, i5/OS, i5/OS (logo), IBM Business Partner (logo), IntelliStation, LoadLeveler, Lotus, Lotus Notes, Notes, Operating System/400, OS/400, PartnerLink, PartnerWorld, PowerPC, pSeries, Rational, RISC System/6000, RS/6000, THINK, Tivoli, Tivoli (logo), Tivoli Management Environment, WebSphere, xSeries, z/OS, zSeries, Active Memory, Balanced Warehouse, CacheFlow, Cool Blue, IBM Systems Director VMControl, pureScale, TurboCore, Chiphopper, Cloudscape, DB2 Universal Database, DS4000, DS6000, DS8000, EnergyScale, Enterprise Workload Manager, General Parallel File System, , GPFS, HACMP, HACMP/6000, HASM, IBM Systems Director Active Energy Manager, iSeries, Micro-Partitioning, POWER, PowerExecutive, PowerVM, PowerVM (logo), PowerHA, Power Architecture, Power Everywhere, Power Family, POWER Hypervisor, Power Systems, Power Systems (logo), Power Systems Software, Power Systems Software (logo), POWER2, POWER3, POWER4, POWER4+, POWER5, POWER5+, POWER6, POWER6+, POWER7, System i, System p, System p5, System Storage, System z, TME 10, Workload Partitions Manager and X-Architecture are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries.

A full list of U.S. trademarks owned by IBM may be found at: http://www.**ibm.com**/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
AltiVec is a trademark of Freescale Semiconductor, Inc.
AMD Opteron is a trademark of Advanced Micro Devices, Inc.
InfiniBand, InfiniBand Trade Association and the InfiniBand design marks are trademarks and/or service marks of the InfiniBand Trade Association.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.
Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries or both.
Microsoft, Windows and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries or both.
NetBench is a registered trademark of Ziff Davis Media in the United States, other countries or both.
SPECint, SPECfp, SPECjbb, SPECweb, SPECjAppServer, SPEC OMP, SPECviewperf, SPECapc, SPEChpc, SPECjvm, SPECmail, SPECimap and SPECsfs are trademarks of the Standard Performance Evaluation Corp (SPEC).
The Power Architecture and Power.org wordmarks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.
TPC-C and TPC-H are trademarks of the Transaction Performance Processing Council (TPPC).
UNIX is a registered trademark of The Open Group in the United States, other countries or both.

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

Revised December 2, 2010