

Raising the Bar: A Jumpstart to Using SOAP Web Services in RPG



Charles Guarino

Central Park Data Systems, Inc.

Copyright Central Park Data Systems, Inc.

1

About The Presenter

With an IT career spanning over 30 years, Charles Guarino has been a consultant for most of them. Since 1995 he has been founder and President of Central Park Data Systems, Inc., a New York area based IBM midrange consulting company. He has published several technical and strategic articles for the IT community and continues in that role as a frequent contributor of programming and developer tips to System i Network and Power IT Pro. He is a proud member of COMMON's Speaker Excellence Hall of Fame and also Long Island Software and Technology Network's Twenty Top Techies of 2009. Charles currently serves as a COMMON Application Development Subject Matter Expert and President and monthly Q&A host of LISUG, a Long Island IBM System i User's Group www.lisug.org. Charles can be reached at cguarino@centralparkdata.com. LinkedIn - <http://www.linkedin.com/in/guarinocharles>

In This Session

- Web services extend your applications by providing functionality you should not or could not write yourself. We will demonstrate what a web service is, where to find them and how to consume them.
- We will explore how supplying just a zip code to an application can return meaningful and real time information that is well beyond your company's four walls.
- All of the source code for these programs can be downloaded and compiled on your own systems and be put to use immediately. They are production ready!

Copyright Central Park Data Systems, Inc.

3

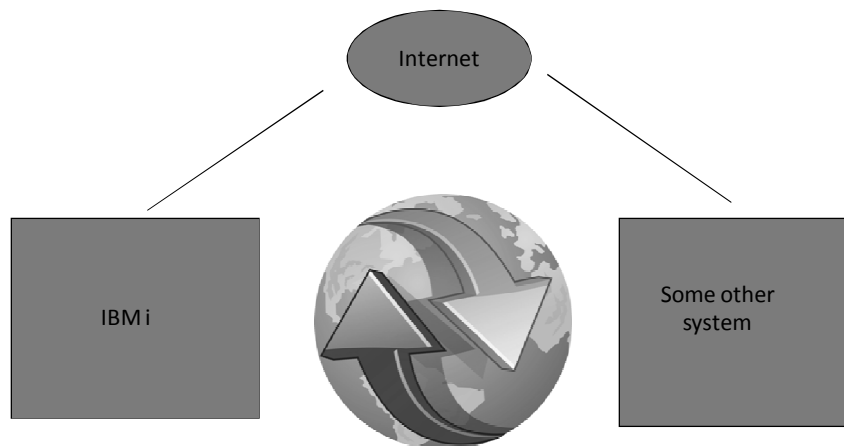
Definitions

- Web Service
- SOAP – Simple Object Access Protocol
- SOAP Methods or Operations
- WSDL – Web Services Definition Language
- URI - Uniform Resource Identifier
- Data Types – such as `xsd_string` and `xsd_int`
- XSD – XML Schema Definition Language

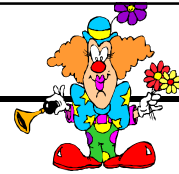
Definitions

- AXIS – The SOAP engine for HTTP servers running Apache
- QSHELL – an environment that runs Unix commands interactively or from a script
- Stub Code
- Stub create, operation and destroy procedures

Simplistic diagram of a web service request



More Web Services Fun Facts



- A STATELESS connection
- Can be sent in the clear or securely
- Can be public or private between companies or individuals
- Hardware, programming language and platform agnostic
- Can save you lots of \$ - pre-written and tested and documented services
- Can be FREE, Restricted Access, Pay-For-Each-Use or Subscription Based
- Extends your applications in ways you may have never known possible, providing real time information.
- Can be many and simultaneous points of failure

Where can you find a web service to consume?



Create your own web services using your own programs

Step 1 – STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)

Step 2 – From a browser to go <http://hostname:2001/HTTPAdmin>

For more information Google “IBM Web Administration for i”

IBM Web Administration for i

Getting started - Create and learn about the servers needed to run your Web content.

- Create a New Web Services Server**
Create Web Services Server Wizard provides a convenient way to externalize existing programs running on IBM i, such as RPG or COBOL, as Web services. This allows Web service clients to interact with IBM i program based services from the Internet or Intranet using Web service based industry standard communication protocols such as SOAP.
- Create a New HTTP Server**
Create a new HTTP Server (powered by Apache) to run your HTTP Web content. This wizard will create everything you need to get started with simple Web serving.
- Create a New Application Server**
Create a new application server to run dynamic Web applications. Create either an IBM Integrated Web application server for i or a WebSphere Application Server.
- Create a New WebSphere Portal**
Create a new application server to run powerful and compelling business partner, customer, and employee information portals. This includes configuring an HTTP server, creating a new WebSphere Application Server, and configuring database and LDAP as necessary.

www.xmethods.net – a popular site for services



[Home](#) · [Tools](#) · [Implementations](#) · [Manage](#) · [Register](#) · [Tutorials](#) · [About](#)

Welcome to XMethods.

Emerging web services standards such as SOAP, WSDL and UDDI will enable system-to-system integration that is easier than ever before. This site lists publicly available web services.

Recent Listings [[View the FULL LIST](#)]

Publisher	Style	Service Name	Description	Implementation
magid	DOC	saudi shipping	calculate saudi shipping prices	
VOORSPRONG	DOC	Worldcup 2010 Football Championships	Access information such as games, cities, results, players etc. about the 2010 tournament	Visual Dataflex
rpamplona	DOC	ACHWorks-SOAP Ver 4 (TSS - Rico Pamplona)	Web Services for ACH Processing and Payments	MS .NET
nsreevatsan	RPC	Chennai Emergency	Emergency Service Provider	Apache SOAP
RestFulServices	DOC	Weather Forecast Service	Provide weather forecast for a given place name	
RestFulServices	DOC	HCPCS Service	The Healthcare Common Procedure Coding System (HCPCS) is a set of health care procedure codes based on the American Medical Association's Current Procedural Terminology (CPT). Commonly pronounced Hick-Picks.	

Searched for Free Soap Web Services

Google free soap web service

Search About 53,000,000 results (0.22 seconds)

Everything Images Maps Videos News Shopping More

Ad - Why this ad?

Soap-Free Cleanser | psico.com
www.psico.com/
Try **Free & Clear soap-free** liquid cleanser for dry, sensitive skin.

Directory of public SOAP Web Services
www.service-repository.com/
List of public available **SOAP Web Services**. ... service-repository.com is a non- UDDI registry of public **Web Services**. ... Feel **free** to register yourself a service.
SOAP client - PeopleAsk - Google Questions Service - Hosts - Schemas

python - Public **free web services** for testing **soap client** - Stack ...
stackoverflow.com/...public-free-web-services-for-testing-soap-clien...
3 answers - Nov 22, 2008
Are there any publicly available **SOAP** 1.2/WSDL 2.0 compliant **free** ... There is a bunch on here: <http://www.webservices.net/WS/wacatlist.aspx> ...

Do you know of any **free SOAP** webservices (for tutorial purpose ...
stackoverflow.com/...do-you-know-of-any-free-soap-webservices-fo...
Mar 21, 2011 - Only for tutorial purpose, I would like to try to use a **free SOAP web services** available. Which ones come to your mind ? The business implied ...

www.service-repository.com

Membrane Directory of Public SOAP Web Services

Home Services Tools Schemas Statistics Help Login

service-repository.com is a non-UDDI registry of public Web Services. You can browse Web Services and call their operations with the build-in SOAP client. The availability of the services is checked by continuous service monitoring. Feel free to register yourself a service.

Name	Description	Rating	Availability	Action
GlobalWeather		★★★★	96.51%	Call It
TopMovie		★★★★	99.89%	Call It
PeopleAsk - Google Questions Service	Get a fresh blend of questions frequently asked in Google's search box.	★★★★	99%	Call It
JDispatcherService	MAFFT (Multiple Alignment using Fast Fourier Transform) is a high speed multiple sequence alignment p...	★★★★	99.87%	Call It
Calculator		★★★★	99.41%	Call It
CentileWSservice	in input, takes a list of numbers separated by ; and a percentile in output, return the value corres...	★★★★	99.95%	Call It
BLZService	BLZService	★★★★	99.99%	Call It
HAui2Service		★★★★	99.92%	Call It
latex		★★★★	97.8%	Call It
IPXtime	Our system uses a multiple sources for resolution. Including a host database, trace route informatio...	★★★★	99.98%	Call It

Registry Source
Get the software of service-repository.com to manage you

Download now!

Web Services Tools
Dynamic SOAP Client
WSDL Diff
Schema Diff

Tags

Events

Date	Description
12-02-04 23:04:56	wsmotos.re
12-01-31 04:55:12	CDC-Confir Dados Cuda up
12-01-31 04:41:53	CDC-Confir Dados Cuda down

CDYNE Weather

http://wiki.cdyne.com/?title=CDYNE_Weather

DEVELOPER WIKI
CDYNE CORPORATION

navigation

- Home - CDYNE APIs
- www.cdyne.com

search

Go Search

toolbox

- What links here
- Related changes
- Special pages
- Printable version
- Permanent link

Click here for LIVE SUPPORT ONLINE

TRIAL KEY FREE

PAGE discussion view source history

CDYNE Weather

Contents [hide]

- Description
- 1.1 Notes
- Developer/Tester Information
- PHP Example
- Example Source Code

Description

CDYNE Weather is a free SOAP Web Service that provides you with up to date weather information in the United States. This information is derived from the National Oceanic and Atmospheric Administration's (NOAA) National Weather service into a clean and easy to parse XML format. The first method offers the city's weather by zip code, and the second shows the weather station's 7 day forecast by zip code. This free Weather Web Service also comes standard with hyperlinks that show images (GIF) of the day's weather conditions, so you can show off your application with visual representations. This service offers two methods:



Weather Visuals Download

Notes

CDYNE considers this service a form of advertising its commercial APIs and is hosted on the same redundant servers. CDYNE will continue to update this service as a part of its offering. If for some reason you cannot receive data on a particular weather station or from city information, please inquire.

RECENT CHANGE: We had a recent change to our free services. Please make sure you change the API URL from ws.cdyne.com to wsf.cdyne.com as seen below.

Developer/Tester Information

- WSDL

WSDL for CDYNE

<http://wsf.cdyne.com/WeatherWS/Weather.asmx?wsdl>

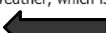
```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://ws.cdyne.com/WeatherWS/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://ws.cdyne.com/WeatherWS/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:s="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <s:schema targetNamespace="http://ws.cdyne.com/WeatherWS/" elementFormDefault="qualified">
      <s:element name="GetWeatherInformation">
        <s:complexType/>
      </s:element>
      <s:element name="GetWeatherInformationResponse">
        <s:complexType>
          <s:sequence>
            <s:element type="tns:ArrayOfWeatherDescription" name="GetWeatherInformationResult" maxOccurs="1"
              minOccurs="0"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ArrayOfWeatherDescription">
        <s:sequence>
          <s:element type="tns:WeatherDescription" name="WeatherDescription" maxOccurs="unbounded" minOccurs="0"/>
        </s:sequence>
      </s:complexType>
      <s:complexType name="WeatherDescription">
        <s:sequence>
          <s:element type="s:short" name="WeatherID" maxOccurs="1" minOccurs="1"/>
          <s:element type="s:string" name="Description" maxOccurs="1" minOccurs="0"/>
          <s:element type="s:string" name="PictureURL" maxOccurs="1" minOccurs="0"/>
        </s:sequence>
      </s:complexType>
      <s:element name="GetCityForecastByZIP">

```

This service provides three operations

Weather

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [GetCityForecastByZIP](#) 
Allows you to get your City Forecast Over the Next 7 Days, which is updated hourly. U.S. Only
- [GetCityWeatherByZIP](#) 
Allows you to get your City's Weather, which is updated hourly. U.S. Only
- [GetWeatherInformation](#) 
Gets Information for each WeatherID

<http://wsf.cdyne.com/WeatherWS/Weather.asmx>

An input form is provided for easy testing.

<http://wsf.cdyne.com/WeatherWS/Weather.asmx?op=GetCityForecastByZIP>

Weather

Click [here](#) for a complete list of operations.

GetCityForecastByZIP

Allows you to get your City Forecast Over the Next 7 Days, which is updated hourly. U.S. Only

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
ZIP:	10038

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /WeatherWS/Weather.asmx HTTP/1.1
Host: wsf.cdyne.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://wsf.cdyne.com/WeatherWS/GetCityForecastByZIP"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetCityForecastByZIP xmlns="http://wsf.cdyne.com/WeatherWS/">
      <ZIP>string</ZIP>
    </GetCityForecastByZIP>
  </soap:Body>
</soap:Envelope>
```

CDYNE returns the weather forecast in XML

```
<?xml version="1.0" encoding="UTF-8"?>
- <ForecastReturn xmlns="http://ws.cdyne.com/WeatherWS/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Success>true</Success>
  <ResponseText>City Found</ResponseText>
  <State>NY</State>
  <City>New York</City>
  <WeatherStationCity>White Plains</WeatherStationCity>
  - <ForecastResult>
    - <Forecast>
      <Date>2012-03-24T00:00:00</Date>
      <WeatherID>2</WeatherID>
      <Description>Partly Cloudy</Description>
      - <Temperatures>
        <MorningLow>47</MorningLow>
        <DaytimeHigh>60</DaytimeHigh>
      </Temperatures>
      - <ProbabilityOfPrecipitation>
        <Nighttime>00</Nighttime>
        <Daytime>40</Daytime>
      </ProbabilityOfPrecipitation>
    </Forecast>
    - <Forecast>
      <Date>2012-03-25T00:00:00</Date>
      <WeatherID>6</WeatherID>
      <Description>Showers</Description>
      - <Temperatures>
        <MorningLow>44</MorningLow>
        <DaytimeHigh>61</DaytimeHigh>
      </Temperatures>
      - <ProbabilityOfPrecipitation>
        <Nighttime>70</Nighttime>
        <Daytime>60</Daytime>
      </ProbabilityOfPrecipitation>
    </Forecast>
  </ForecastResult>
</ForecastReturn>
```

Header

Detail

www.soapui.org

soapUI

Home Support Pro Store Downloads Open Source Search

10 REASONS TO GO PRO

1 The Editors

The **Outline Editor** helps to visualize the XML, and the **Form Editor** uses the XML to create a form for you to enter information. The Editors – one of the 10 Reasons to Go PRO.

FREE!

Download Trial Free 14 day evaluation

Read more 10 Reasons to Go PRO

About soapUI

- What is soapUI?
- Features
- What's new in 4.5
- soapUI Pro
- Gallery
- Product Comparison
- System Requirements
- Release History
- Roadmap
- soapUI FAQ

Latest News In the News

soapUI 4.5 and loadUI 2.0 BETA 2 out!

Alright, we've worked on some issues that came to notice during the Beta 1 period, and now it's time for the second iteration: BETA 2. Some of ...

Mar 11 12 [Read more...](#)

First steps

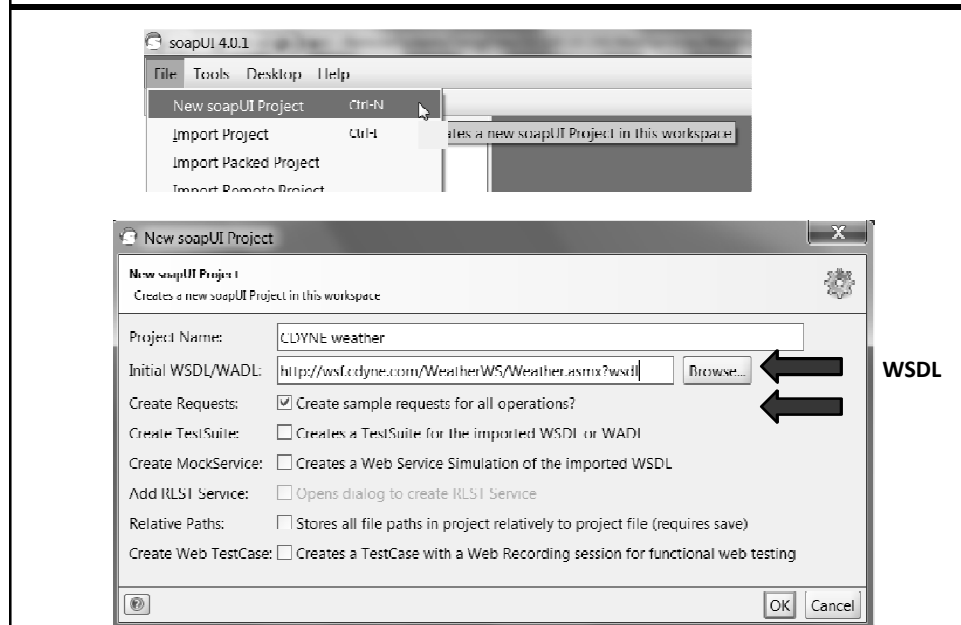
Where do I start?

- » What is soapUI?
- » soapUI Features
- » Read the soapUI Pro Comparison
- » Read the Getting Started with soapUI Guide
- » Try the Web Service sample soapUI Project

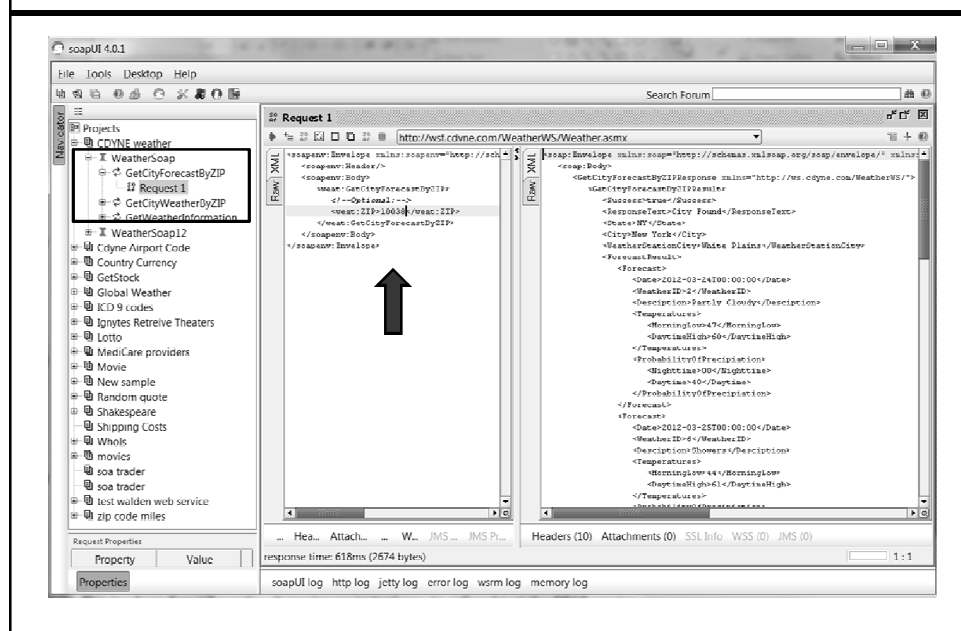
Follow us:

Copyright Central Park Data Systems, Inc.

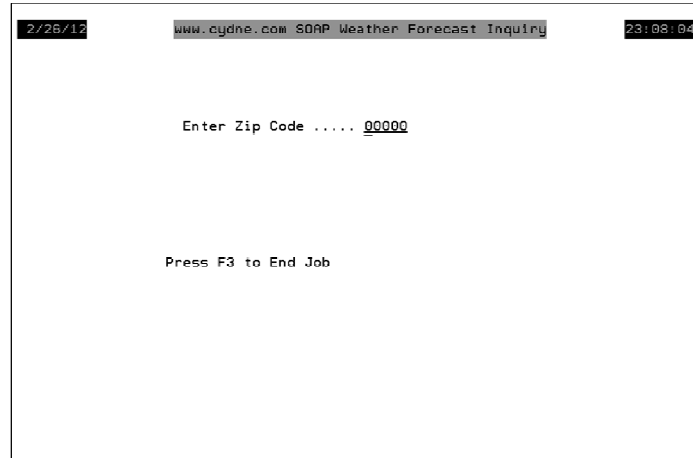
Each service is considered a Project



Same Web Service using SoapUI graphical interface



Here is what we want to accomplish:



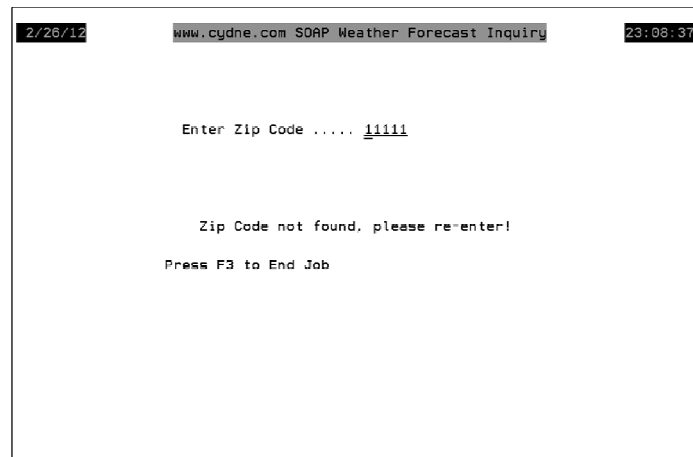
A terminal window titled "www.cydne.com SOAP Weather Forecast Inquiry" with a date of "2/26/12" and a time of "23:08:04". The prompt "Enter Zip Code" is followed by the input "00000". Below this, the instruction "Press F3 to End Job" is displayed.

```
2/26/12 www.cydne.com SOAP Weather Forecast Inquiry 23:08:04  
  
Enter Zip Code . . . . . 00000  
  
Press F3 to End Job
```

Copyright Central Park Data Systems, Inc.

21

When an invalid zip code is keyed, display second error message:



A terminal window titled "www.cydne.com SOAP Weather Forecast Inquiry" with a date of "2/26/12" and a time of "23:08:37". The prompt "Enter Zip Code" is followed by the input "11111". Below this, the error message "Zip Code not found, please re-enter!" is displayed, followed by the instruction "Press F3 to End Job".

```
2/26/12 www.cydne.com SOAP Weather Forecast Inquiry 23:08:37  
  
Enter Zip Code . . . . . 11111  
  
Zip Code not found, please re-enter!  
  
Press F3 to End Job
```

Copyright Central Park Data Systems, Inc.

22

If an error occurs when service is called display third message:

2/26/12 www.cdyne.com SOAP Weather Forecast Inquiry 23:11:37

Enter Zip Code 10038

An error occurred when calling CDYNE service, please try again later.

Press F3 to End Job

Copyright Central Park Data Systems, Inc.

23

Subfile displayed with fully parsed data for zip code 10038:

2/26/12 www.cdyne.com SOAP Weather Forecast Inquiry 23:12:17

Zip Code ... 10038

City New York

State NY

Weather Station City ... White Plains

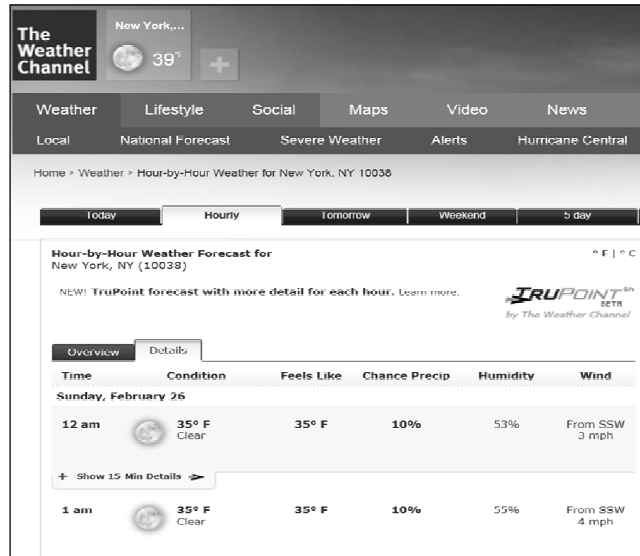
Press F1 for hourly forecast
Press F12 to Return

Date	Description	Temperature		Prob of Precip	
		Low	High	Morn	Night
2012-02-26	Sunny	27	53	10	00
2012-02-27	Sunny	33	44	00	10
2012-02-28		31	35	70	10
2012-02-29		34	39	50	70
2012-03-01	Partly Cloudy	28	41	10	20
2012-03-02	Showers	35	48	50	20
2012-03-03	Partly Cloudy	35	46	20	50

Copyright Central Park Data Systems, Inc.

24

If F1 is pressed launch Internet Explorer and display weather.com



Let's go through the steps to see how we achieved this!

- Download the IBM Web Services Client for ILE Programming Guide
- Select a SOAP web service and identify the WSDL URI
- Choose a folder where you want to create and store your stub code
- Decide which library you want the tool to create your service program
- Start QSHLL and run the command
- Create a binding directory and add the service program as an entry
- Identify the incoming and outbound parameters for the service using the "RPGLEINC" member
- Create your RPG program to prepare the service to run ("create"), consume the service and then free up ("destroy") the resource upon completion.

Before we start... IBM Licensed Program Requirements

Prerequisites

Below are the prerequisites that are required to be installed on the system in order to use the Integrated Web services for IBM I support.

Release	Product prerequisites
7.1	5770SS1 Option 3 - Extended Base Directory Support 5770SS1 Option 12 - Host Servers 5770SS1 Option 30 - Qshell 5770SS1 Option 33 - PASE 5770SS1 Option 34 - Digital Certificate Manager 4 (needed only for Web services client) 5761JV1 Option 11 - IBM Developer Kit for Java (Java SE 6 32 bit) 5770DG1 - IBM HTTP Server for I
6.1	5761SS1 Option 3 - Extended Base Directory Support 5761SS1 Option 12 - Host Servers 5761SS1 Option 30 - Qshell 5761SS1 Option 33 - PASE 5761SS1 Option 34 - Digital Certificate Manager 4 (needed only for Web services client) 5761JV1 Option 8 - IBM Developer Kit for Java (J2SE 5.0 32 bit) 5761DG1 - IBM HTTP Server for I
6.4	5722SS1 Option 3 - Extended Base Directory Support 5722SS1 Option 12 - Host Servers 5722SS1 Option 30 - Qshell 5722SS1 Option 33 - PASE 5722SS1 Option 34 - Digital Certificate Manager 4 (needed only for Web services client) 5722JV1 Option 8 - IBM Developer Kit for Java (J2SE 5.0 32 bit) 5722DG1 - IBM HTTP Server for I

➡ ALSO MAKE SURE YOU HAVE SS1 "SYSTEM OPENNESS INCLUDES" INSTALLED

Google "IBM Web Services Client For ILE Programming Guide"



Part 1. Web services fundamentals
Chapter 1. Introduction to Web services
Chapter 2. XML primer
Chapter 3. SOAP primer
Chapter 4. WSDL primer
Part 2. Web services client for ILE concepts
Chapter 5. Web services client overview
Chapter 6. Command line tools
Chapter 7. Configuration files
Chapter 8. The Web services client for ILE installation details
Part 3. Using C++ stubs
Chapter 9. WSDL and XML to C++

Select the web service you want to use:
http://wiki.cdyne.com/?title=CDYNE_Weather

The WSDL URI

<http://wsf.cdyne.com/WeatherWS/Weather.asmx?wsdl>

```
<?xml version="1.0" encoding="UTF-8"?>
- <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://ws.cdyne.com/WeatherWS/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://ws.cdyne.com/WeatherWS/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:s="http://www.w3.org/2001/XMLSchema">
  - <wsdl:types>
    - <s:schema targetNamespace="http://ws.cdyne.com/WeatherWS/" elementFormDefault="qualified">
      - <s:element name="GetWeatherInformation">
        <s:complexType/>
      </s:element>
      - <s:element name="GetWeatherInformationResponse">
        <s:complexType>
          <s:sequence>
            <s:element name="GetWeatherInformationResult" type="tns:ArrayOfWeatherDescription" maxOccurs="1"
              minOccurs="0"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      - <s:complexType name="ArrayOfWeatherDescription">
        <s:sequence>
          <s:element name="WeatherDescription" type="tns:WeatherDescription" maxOccurs="unbounded" minOccurs="0"/>
        </s:sequence>
      </s:complexType>
      - <s:complexType name="WeatherDescription">
        <s:sequence>
          <s:element name="WeatherID" type="s:short" maxOccurs="1" minOccurs="1"/>
          <s:element name="Description" type="s:string" maxOccurs="1" minOccurs="0"/>
          <s:element name="PictureURL" type="s:string" maxOccurs="1" minOccurs="0"/>
        </s:sequence>
      </s:complexType>
      - <s:element name="GetCityForecastByZIP">
        <s:complexType>
          <s:sequence>
            <s:element name="ZIP" type="s:string" maxOccurs="1" minOccurs="0"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>
```

Where will you store the source code?

Source code will automatically be placed in the IFS

By default it is placed in your HOME directory

You can override this location at creation time

We will not use the default for this example

Our service program will be placed in library
WEBSERVICE

WSDL2SH.SH (runs in QSHELL)

```
The wsdl2ws.sh command line tool

Synopsis

wsdl2ws.sh [arguments] WSDL-URI

Description

The wsdl2ws.sh command enables you to generate Web service client stubs (sometimes referred to
service interface stubs or Web service client proxies) from a WSDL file. The wsdl2ws.sh command uses
the WSDL that is passed to it, and any associated XSD files, to create the client stub code.

Required arguments

WSDL-URI
    Specifies the location of the input WSDL file using a Universal Resource Identifier (URI). You can
    also use a regular file path if the WSDL file is on the local file system.

Optional arguments

-h, -help
    Displays a help message and exits.

-l <language>
    Target language - default is c++. C or RPG stubs can also be generated.

-o <directory>
    Sets the root directory for emitted files. Default is the current working directory. If a directory in
    the specified path <directory> does not exist, the directory will be created.

-m <max-string-size>
    Maximum size to use when defining RPG character fields to hold string data. Minimum value is
    16. Maximum value for IBM i 5.4 is 650001. Maximum value for IBM i 6.1 and subsequent
    releases is 80000002. Default is 128. This option is valid only if -lrpg is specified.
```

The parameters of WSDL2SH continued...

-mb *<max binary size>*
Maximum size to use when defining RPG character fields to hold binary data (XSD hexBinary and base64Binary types). Minimum value is 48. Maximum value for IBM i 5.4 is 65000³. Maximum value for IBM i 6.1 and subsequent releases is 16000000³. This option is valid only if **-lrpg** is specified.

-ma *<max array size>*
Maximum size to use when defining RPG arrays. Minimum value is 1. Maximum value for IBM i 5.4 is 32000³. Maximum value for IBM i 6.1 and subsequent releases is 80000000³. Default is 20. This option is valid only if **-lrpg** is specified.

-s *<service program>*
Path to service program to be built using the generated code. For example, **-s/QSYS.LIB/MYLIB.LIB/MYWS.SRVPGM**. The path can also point to a library, in which case the name of the service program will be the prefix (specified by the **-p** option) appended with 'WS.SRVPGM'. If the option is not specified, a service program is not built.

-d
Generate service program with debug views. If specified, lowest level of optimization is used. Default is to generate fully optimized code with no debug views.

-p *<prefix>*
1-3 character prefix. The prefix will be used in the names of module objects (*MODULE). It will also be used in the service program name if a name is not passed on the **-s** option.

-v
Be verbose - will show exception stack trace when exceptions occur.

-t *<timeout>*
Specifies how long the command waits, in seconds, for the WSDL-URI to respond before giving up. The default is 0 (no timeout).

-b *<binding-name>*
Binding name that will be used to generate stub.

-w *<wrapped | unwrapped>*
Generate wrapper style or not - default is wrapped. This affects the definition of the Web service operation that is generated - whether the Web service operation will have one input structure as a parameter (unwrapped) or whether the individual fields in the structure can be passed as individual parameters (wrapper-style). In order for an operation to be eligible for wrapper-style, the following criteria must be met:

- There is at most one single part in input and output messages.
- Each part definition must reference an element.
- The input element must have the same name as the operation.
- The input and output elements have no attributes.

Examples

1. The following generates RPG stub code in directory /Stub/rpg using URI to WSDL file and compiles the stub code into a service program:
`wsl2ws.sh -lrpg -o/Stub/rpg -s/qsys.lib/ws.lib/wsrpg.srvpgm
http://ip0out16-10021/web/services/ConvertTemp.wsdl`
2. The following generates C stub code in directory /Stub/c using path to WSDL file and compiles the stub code into a service program:
`wsl2ws.sh -lc -o/Stub/c -s/qsys.lib/ws.lib/wsc.srvpgm
/Stub/ConvertTemp.wsdl`

WSDL2SH.SH {arguments} WSDL-URI

Parameters you will likely be using:

- l Target language
- o IFS directory
- s Service program and library
- d Create service program with debug views
- ma Maximum array size (default is 20)
- p Prefix for modules – VERY IMPORTANT
if you use only one library for development

WSDL2RPG !!!

IBM. - English - Sign in (or register)

developerWorks. Technical topics Evaluation software Community Events Search developerWorks

developerWorks > Technical topics > IBM > Technical library >

Invoking web services using RPG proxy

Setting up and generating web service RPG proxy

Nadir Amra (amra@us.ibm.com), IBM software engineer, IBM

Date: 08 Jun 2011
Level: Introductory
Also available in: Chinese

Summary: This article demonstrates how you can invoke a web service from an RPG application using a new feature of the integrated web services client for ILE - the ability to generate an RPG proxy (or stub).

Tags for this article: rpg, web_services

Tag this! Update My dW Interests (Log in | What's this?)

Activity: 2369 views
Comments: 8 (# View | Add comment - Sign in)

★ ★ ★ ★ ★ Average rating (6 votes)
 Rate this article

The wsdl2rpg.sh command line tool

The `wsdl2rpg.sh` command has the same options as `wsdl2ws.sh` with the exception of the `-l` (target language) option. Specifying `wsdl2ws.sh` with `-lrpg` is the same as using the `wsdl2rpg.sh` command.

If you Google this you will likely also find the non-IBM version written by Thomas Raddatz. This is NOT the one supported by IBM.

STRQSH

```
QSH Command Entry

$

===> /qibm/proddata/os/WebServices/V1/client/bin/wsdl2ws.sh
      http://wsf.cdyne.com/WeatherWS/Weather.asmx?wsdl
      -o/webservices/weatherws -lrpg -d
      -s/qsys.lib/web service.lib/weathrsvc.srvpgm

F3=Exit F6=Print F9=Retrieve F12=Disconnect
F13=Clear F17=Top F18=Bottom F21=CL command entry
```



In addition, in order to simplify paths when referring to files or commands in the Web Services Client for ILE install directory, `/QIBM/ProdData/OS/WebServices/V1/client`, we will use `<install_dir>` as the initial path in path names to represent the install directory.

A line by line analysis

```

===> /qibm/proddata/os/WebServices/V1/client/bin/wsd12ws.sh
      http://wsf.cdyne.com/WeatherWS/Weather.asmx?wsdl
      -o/webservices/weatherws -lrpg -d
      -s/qsys.lib/web service.lib/weathrsvc.srvpgm
F3=Exit  F6=Print F9=Retrieve F12=Disconnect
F13=Clear F17=Top  F18=Bottom F21=CL command entry

```

```

/qibm/proddata/os/WebServices/V1/client/bin/wsd12ws.sh

```

Execute command “wsd12ws.sh” from the install directory of /qibm/proddata/os/WebServices/V1/client/bin

```

===> /qibm/proddata/os/WebServices/V1/client/bin/wsd12ws.sh
      http://wsf.cdyne.com/WeatherWS/Weather.asmx?wsdl
      -o/webservices/weatherws -lrpg -d
      -s/qsys.lib/web service.lib/weathrsvc.srvpgm
F3=Exit  F6=Print F9=Retrieve F12=Disconnect
F13=Clear F17=Top  F18=Bottom F21=CL command entry

```

```

http://wsf.cdyne.com/WeatherWS/Weather.asmx?wsdl

```

The URI where the WSDL is located

```

===> /qibm/proddata/os/WebServices/V1/client/bin/wsl2ws.sh
      http://wsf.cdyne.com/WeatherWS/Weather.asmx?wsdl
      -o/webservices/weatherws -lrpg -d
      -s/qsys.lib/web service.lib/weathrsvc.srvpgm
F3=Exit  F6=Print F9=Retrieve F12=Disconnect
F13=Clear F17=Top  F18=Bottom F21=CL command entry

```

-o/webservices/weatherws -lrpg -d

Place the tool's output in directory /webservices/weatherws
 Create RPG language source members
 Create service program weathrsvc with debug views

```

===> /qibm/proddata/os/WebServices/V1/client/bin/wsl2ws.sh
      http://wsf.cdyne.com/WeatherWS/Weather.asmx?wsdl
      -o/webservices/weatherws -lrpg -d
      -s/qsys.lib/web service.lib/weathrsvc.srvpgm
F3=Exit  F6=Print F9=Retrieve F12=Disconnect
F13=Clear F17=Top  F18=Bottom F21=CL command entry

```

-s/qsys.lib/web service.lib/weathrsvc.srvpgm

Create the service program WEATHRSVC
 in library WEBSERVICE

SUCCESS !!!

QSH Command Entry

```
> /qibm/proddata/os/WebServices/V1/client/bin/wsd12ws.sh
    http://wsf.cdyne.com/WeatherWS/Weather.asmx?wsdl
    -o/webservices/weatherws -lrpg -d
    -s/qsys.lib/web service.lib/weathrsvc.srvpgm

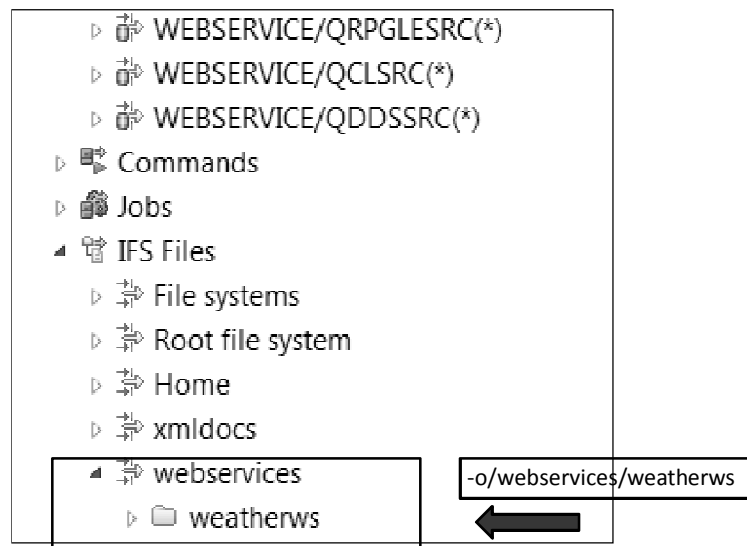
Code generation completed. Generated files in directory
'/webservices/weatherws'.

Attempting to create service program...

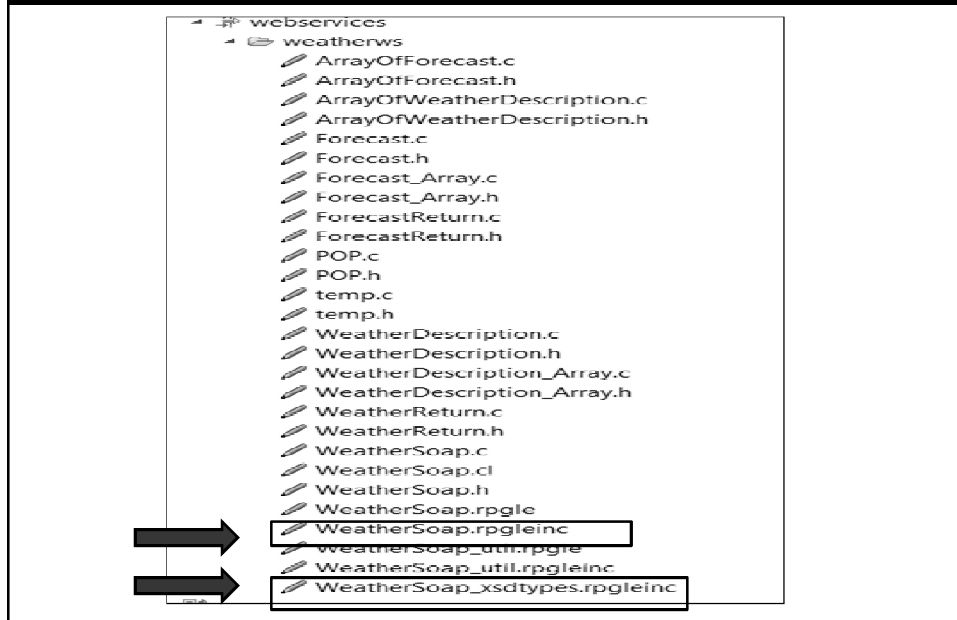
Service program created. Service program is
'/QSYS.LIB/WEBSERVICE.LIB/WEATHRSVC.SRVPGM'.
$

===> _____
```

What just happened?



Source members created in /webservices/weatherws



Service program WEATHRSVC was created DSPSRVPGM WEATHRSVC

-s/qsys.lib/webservice.lib/weathrsvc.srvpgm



Display Service Program Information		Display 5 of 10
Service program	WEATHRSVC	
Library	WEBSERVICE	
Owner	CGUARINO	
Service program attribute		
Detail	*PROCEXP	
Procedure Exports:		
Procedure Name	ARGOPT	
destroy_WeatherSoap_stub	*NO	
get_WeatherSoap_stub	*NO	
get_WeatherSoap_Status	*NO	
set_WeatherSoap_ExceptionHandler	*NO	
stub_create_WeatherSoap@	*NO	
stub_destroy_WeatherSoap@	*NO	
Axis_Create_temp	*NO	
Axis_Create_ArrayOfForecast	*NO	
Axis_Create_ArrayOfWeatherDescription	*NO	
Axis_Create_Forecast	*NO	
		More...
F3=Exit F12=Cancel F17=Top F18=Bottom		

Place the newly created service program into a binding directory

Create Binding Directory (CRTBNDDIR)

Binding directory: WEBSERVICE Name
 Library: WEBSERVICE Name
 Authority: *LIBCIRI/AVI Name
 Text 'description': 'Contains Service Programs Created By' Character value

☐ Advanced ☐ All Parameters ☐ Keywords

CRTBNDDIR BNDDIR(WEBSERVICE/WEBSERVICE) TEXT('Contains Service Programs Created By WSDI 2RPG')

Add Binding Directory Entry (ADDBNDDIRE)

Binding directory: webservice Name
 Library: webservice Name
 Object specifications:
 Object: weatherws Name, generic*
 Library: webservice Name
 Object type: *SRVPGM
 Activation: *IMMED Add

Position specifications:
 Object position: *LAST
 Object: Name
 Library: *LIBL Name
 Object type: *SRVPGM

☐ Advanced ☐ All Parameters ☐ Keywords

ADDBNDDIRE BNDDIR(WEBSERVICE/WEBSERVICE) OBJ

Identify the web service create and destroy prototypes and their parameters. These will be found in the .RPGLEINC source member.

```

* *****
* WEB SERVICE CLIENT STUB PROTOTYPES
* *****

* *****
* RPG Call : stub_create WeatherSoap
* parm in : This t this
* return : *ON = connected
*         : *OFF = connect failed
* *****
D stub_create_WeatherSoap...
D          FR              1N  extproc('stub_create_WeatherSoap@')
D this          likeds(This_t)

* *****
* RPG Call : stub_destroy WeatherSoap
* parm in : This t this
* return : *ON = destroyed
*         : *OFF = destroy failed
* *****
D stub_destroy_WeatherSoap...
D          FR              1N  extproc('stub destroy WeatherSoap@')
D this          likeds(This_t)

```

Identify the stub_op for the web service operation you want to consume.
Pay particular attention to the parameters and how they're defined.

```

* *****
* RPG call : stub_op_GetCityWeatherByZIP
* parm in : This_t this
* parm in : xsdc_string Value0
* parm out: WeatherReturn out
* return  : *ON = success
*          : *OFF = failure
* *****
D stub_op_GetCityWeatherByZIP...
D          PR          1N      extproc('GetCityWeatherByZIP@')
D this      likeds(This_t)
D Value0    likeds(xsd_string)
D out       likeds(WeatherReturn_t)

* *****
* RPG call : stub_op_GetCityForecastByZIP
* parm in : This_t this
* parm in : xsdc_string Value0
* parm out: ForecastReturn out
* return  : *ON = success
*          : *OFF = failure
* *****
D stub_op_GetCityForecastByZIP...
D          PR          1N      extproc('GetCityForecastByZIP@')
D this      likeds(This_t)
D Value0    likeds(xsd_string)
D out       likeds(ForecastReturn_t)

```

Qualified
Data
Structures

Where are those qualified data structures coming from?
IFS source members "WeatherSoap..."

_xsdtypes.rpgleinc

```

D This_t      DS          qualified based(Template)
D endpoint    2048a
D handle      *
D excOccurred 1n
D excCode     10i 0
D excString   2040a
D reserved    1024a

```

```

D xsd_string...
D          DS          qualified based(Template)
D isNil      1n
D value      a varying(4) len(120)
D reserved   1a

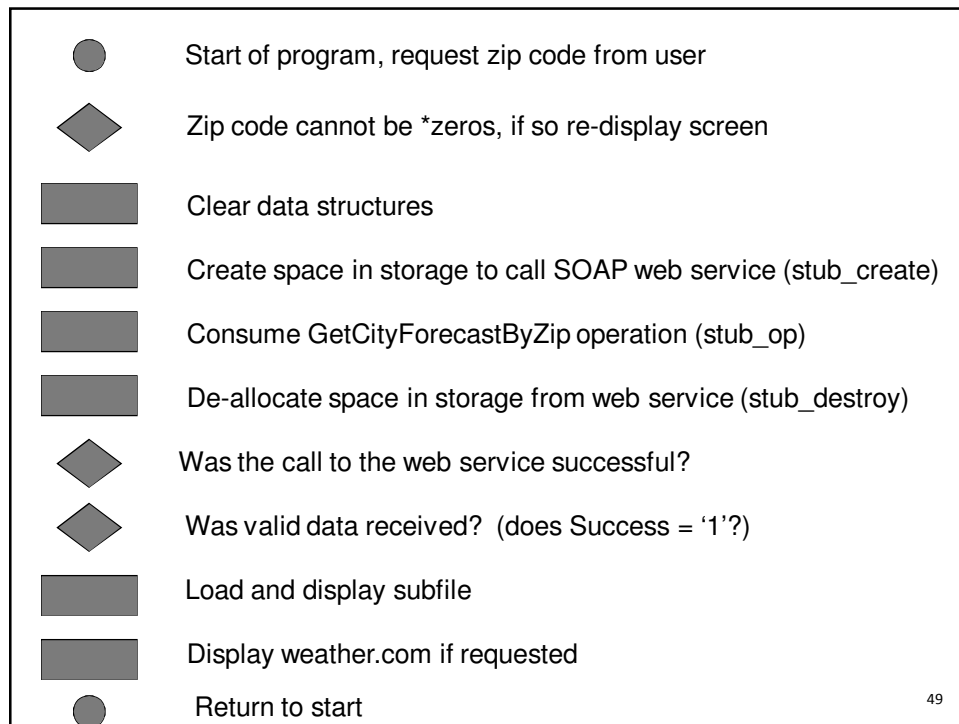
```

.rpgleinc

```

* *****
* RPG type : ForecastReturn_t
* *****
D ForecastReturn_t...
D          DS          qualified based(Template)
D isNil_ForecastReturn_t... 1n
D Success      likeds(xsd_boolean)
D ResponseText likeds(xsd_string)
D State        likeds(xsd_string)
D City         likeds(xsd_string)
D WeatherStationCity...
D          likeds(xsd_string)
D ForecastResult likeds(ArrayOfForecast_t)

```



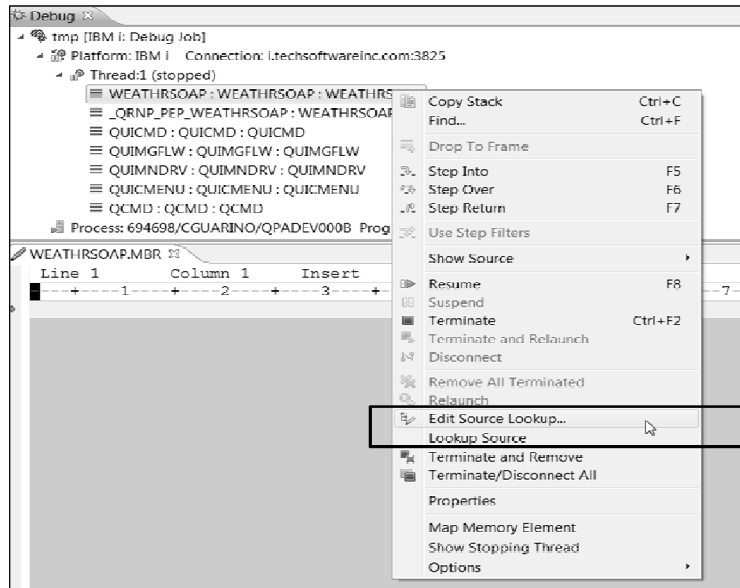
Program WEATHRSOAP

```

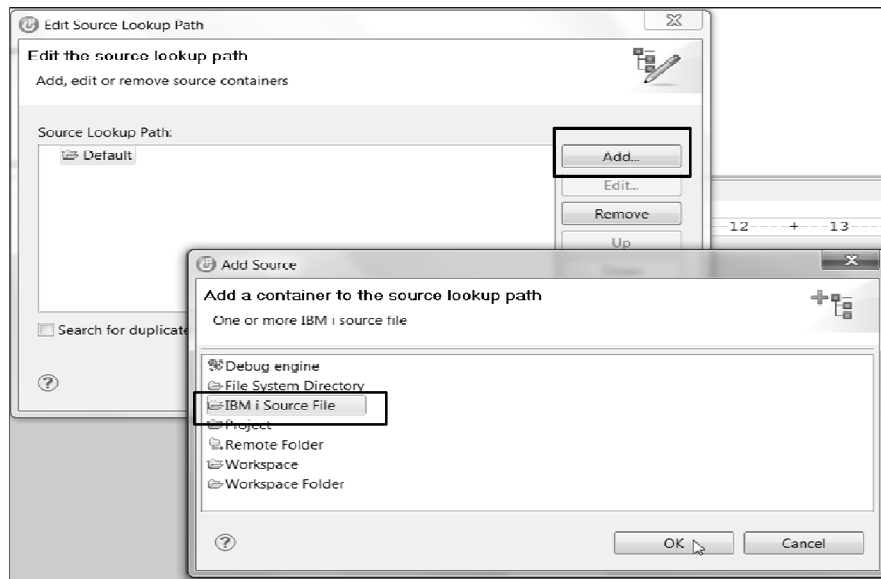
WEATHRSOAP.RPGLE
Line 60      Column 16      Replace      Browse
.....1.....2.....3.....4.....5.....6.....7.....8
002400      /free
002500      // Prompt user for zip code
002600      dou endofjob;
002700      // Was EGY requested?
002800      if endofjob;
002900      *inlr = *on;
003000      return; *on;
003100      endif;
003200      // Clear error conditons
003300      zipcodezero = *off;
003400      zipcodenotfound = *off;
003500      // Zipcode cannot be equal to all zeros
003600      if zipcode = *zeros;
003700      zipcodezero = *on;
003800      itar;
003900      endif;
004000      // Prepare to call soap service
004100      clear webServiceDs;
004200      clear ZipCodeDs;
004300      clear ForecastDs;
004400      ZipCodeDs.value = %char(zipcode);
004500      if stub_create_WeatherSoap(WebServiceDs);
004600      // Invoke the Web service operation.
004700      if stub_op_GetCityForecastByZip(WebServiceDs:ZipCodeDs:ForecastDs);
004800      endif;
004900

```

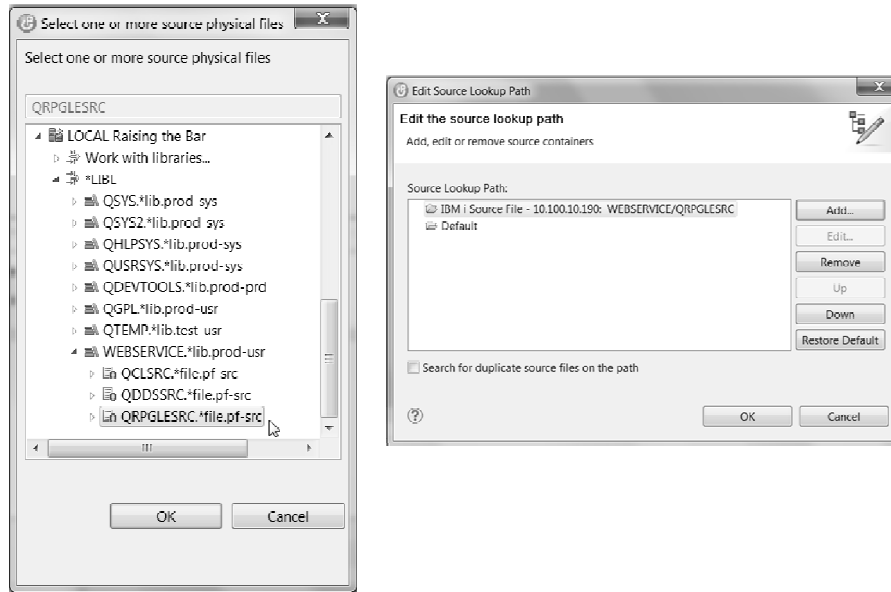
Debugging WEATHRSOAP in RDP



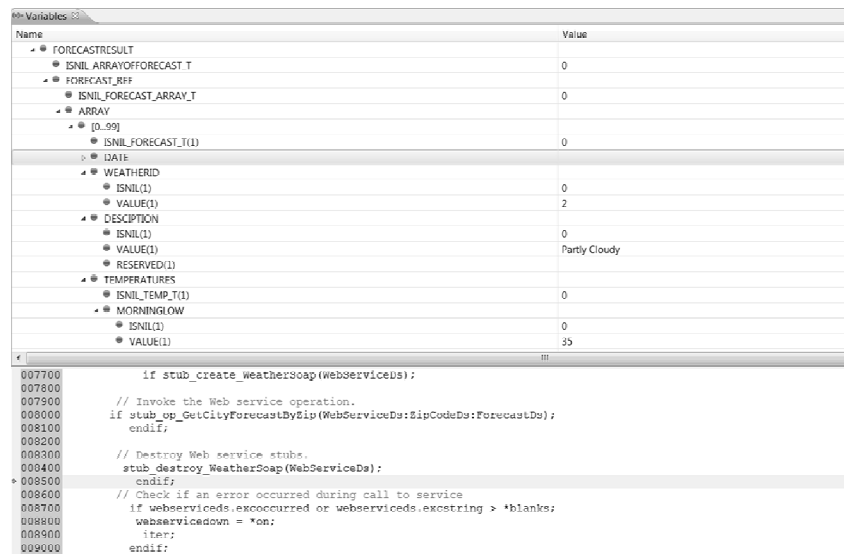
Need to tell RDP where the source code resides



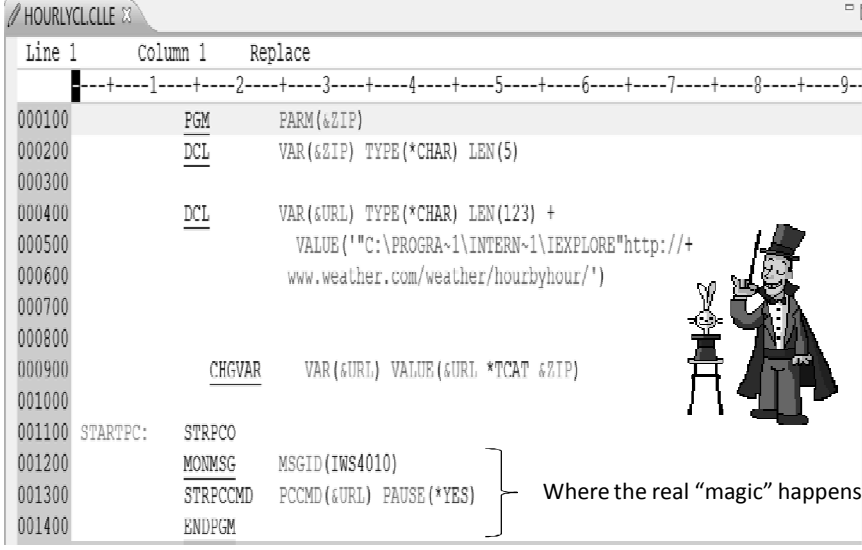
Source code is stored in WEBSERVICE/QRPGLESRC



Stepping through the code...



CLLE that launches Internet Explorer web browser



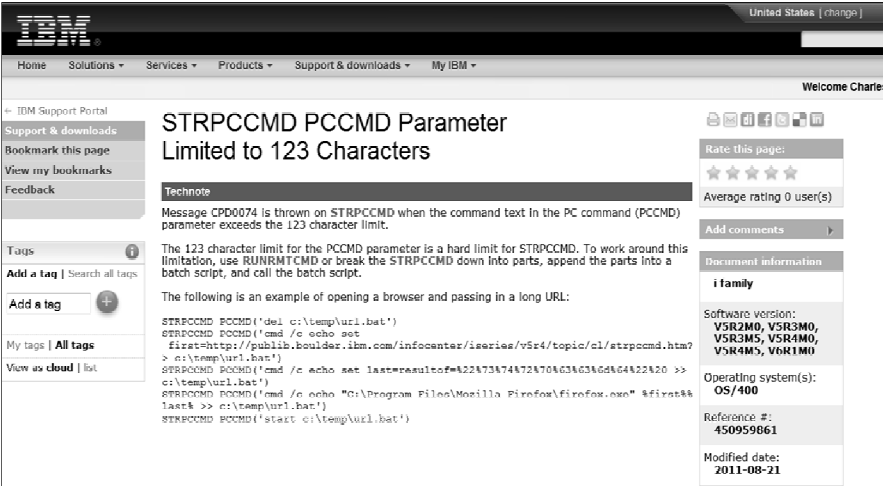
```

//HOURLYCLLE
Line 1      Column 1      Replace
-----1-----2-----3-----4-----5-----6-----7-----8-----9-----
000100      PGM          PARM(&ZIP)
000200      DCL          VAR(&ZIP) TYPE(*CHAR) LEN(5)
000300
000400      DCL          VAR(&URL) TYPE(*CHAR) LEN(123) +
000500                      VALUE('"C:\PROGRA~1\INTERN~1\IEEXPLORE"http://+
000600                      www.weather.com/weather/hourbyhour/')
000700
000800
000900      CHGVAR        VAR(&URL) VALUE(&URL *TCAT &ZIP)
001000
001100  STARTPC:  STRPCO
001200          MONMSG      MSGID(IWS4010)
001300          STRPCCMD    PCCMD(&URL) PAUSE(*YES)
001400          ENDPGM
  
```

Where the real "magic" happens

Copyright Central Park Data Systems, Inc. 55

In case you have a very long URL



IBM Support Portal

STRPCCMD PCCMD Parameter Limited to 123 Characters

Technote

Message CPD0074 is thrown on STRPCCMD when the command text in the PC command (PCCMD) parameter exceeds the 123 character limit.

The 123 character limit for the PCCMD parameter is a hard limit for STRPCCMD. To work around this limitation, use RUNRMTCMD or break the STRPCCMD down into parts, append the parts into a batch script, and call the batch script.

The following is an example of opening a browser and passing in a long URL:

```

STRPCCMD PCCMD('del c:\temp\url.bat')
STRPCCMD PCCMD('cmd /c echo set first=http://publib.boulder.ibm.com/infocenter/iseries/v5r4/topic/cl/strpccmd.htm?
> c:\temp\url.bat')
STRPCCMD PCCMD('cmd /c echo set lastresultof=42247347467247040346346464422420 >>
c:\temp\url.bat')
STRPCCMD PCCMD('cmd /c echo "C:\Program Files\Mozilla Firefox\Firefox.exe" %first% %
last% >> c:\temp\url.bat')
STRPCCMD PCCMD('start c:\temp\url.bat')
  
```

Software version:
V5R2M0, V5R3M0,
V5R3M5, V5R4M0,
V5R4M5, V6R1M0

Operating system(s):
OS/400

Reference #:
450959061

Modified date:
2011-08-21

Search for "STRPCCMD PCCMD"

Raising the Bar: A Jumpstart to Using SOAP Web Services in RPG



Charles Guarino
THANK YOU !!!

Copyright Central Park Data Systems, Inc.

57

Sessions by Charles Guarino

Sunday 9:30AM	REST Web Services for Everyday RPG Consumption Using REST to extend your applications beyond your four walls
Sunday 12:30PM	Raising the Bar: A Jumpstart to Using SOAP Web Services in RPG A step-by-step discussion on implementing IBM's WSDL2RPG
Monday 11:00AM	From STRPDM to RDP An introduction to RDP and source code editing
Monday 2:00PM	From STRPDM to RDP – The Journey Continues Exploring advanced features in RDP and making it your own
Monday 3:30PM	From STRSDA and STRRLU to RDP A step by step demonstration on using the RDP visual designer
Tuesday 9:30 AM	From STRDBG to the RDP Debugger A walk through on using the debugging perspective
Wednesday 9:30AM	Introduction to Processing XML in RPG A beginner's guide to processing XML in RPG
Wednesday 2:00PM	There's Gold in Them Thar Hills! Making the most of an RPG formatted dump
Wednesday 5:00PM	Program Application Recovery and Other Tricks Proven techniques to keep your cool before and after an IT crisis

Copyright Central Park Data Systems, Inc.

58