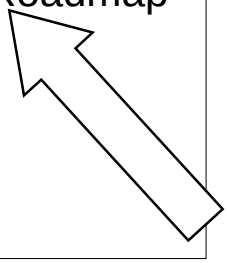
IBM Software Group

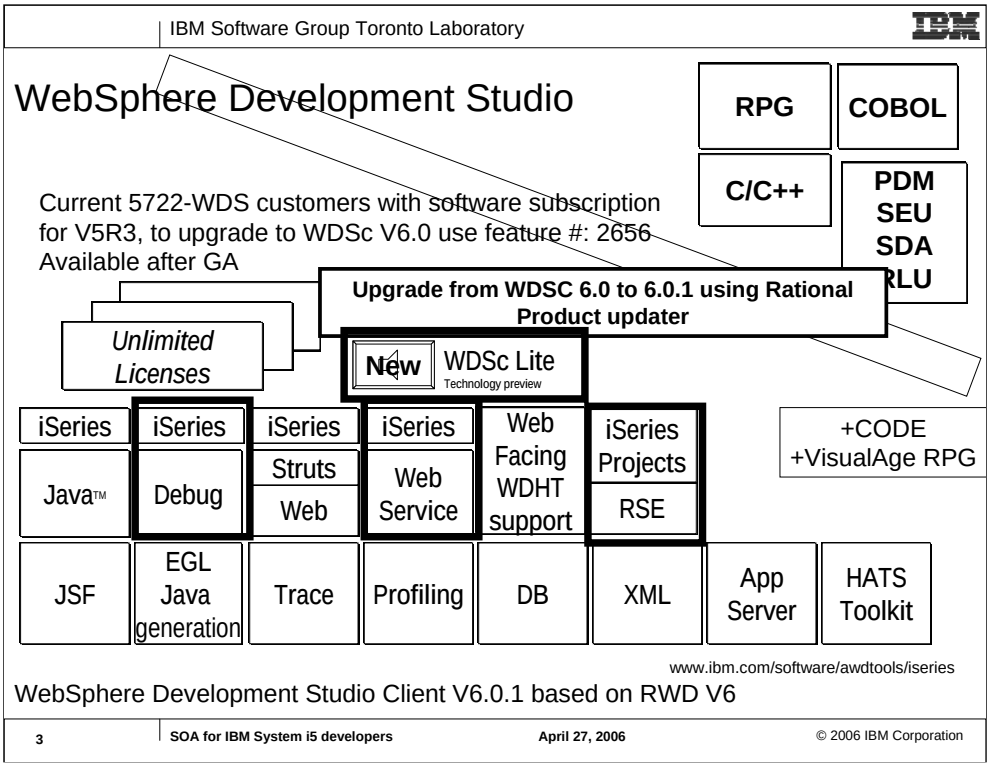
WebSphere Development Studio Client for iSeries
How to participate in SOA with RPG or COBOL
Using WebServices with RPG
Making RPG programs a WebService

Claus Weiss weiss@ca.ibm.com
George Farr farr@ca.ibm.com

AGENDA

- Overview WDSc 6.0.1 and the AD Roadmap
- Service Oriented Architecture SOA
- WebServices what are they ?
  - **Provide a RPG/COBOL program as a WebService**
  - **Call a WebService from RPG/COBOL**



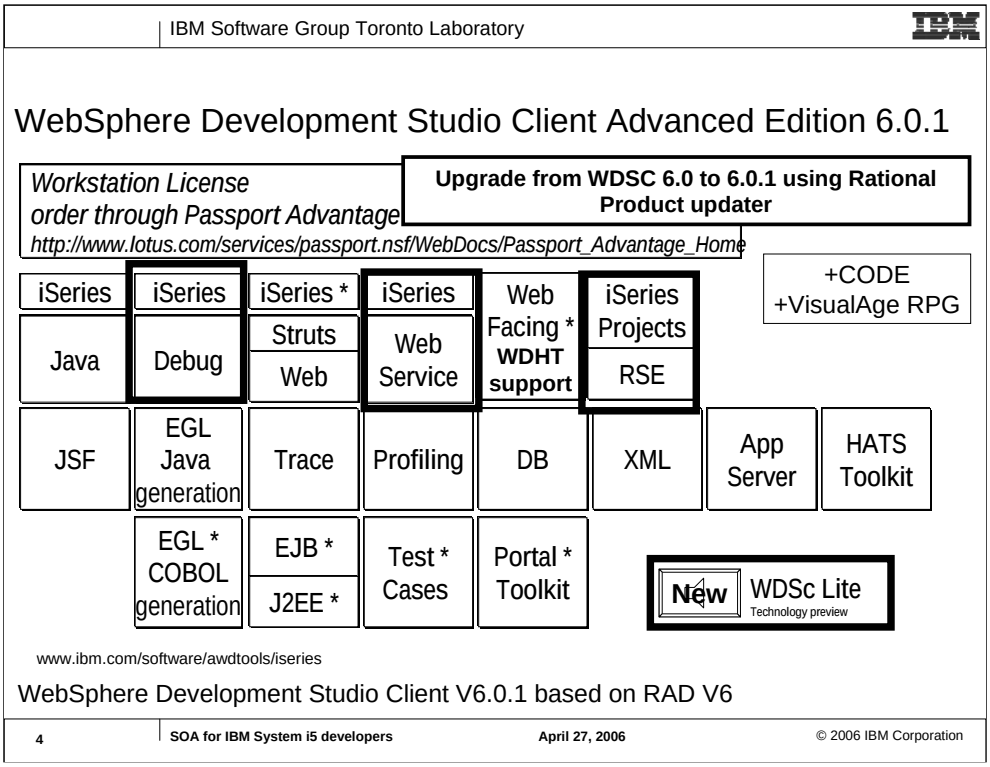


There is now only one application development product sold by IBM, for iSeries, as of V4R5. This is WebSphere Development Studio (Development Studio), which includes all four host compilers, all traditional tools (ADTS = PDM+SEU+SDA+RLU+DFU+AFP+CGU), and unlimited licenses of the workstation-based toolset named WebSphere Development Studio Client (formerly WebSphere Development Tools).

If you are an existing customer who has a subscription, you can upgrade to Development Studio free of charge. Without a Software Subscription, there is an upgrade fee. New licenses of Development Studio are priced very competitive compared to the combined prices of all constituent products. As of V5R1, there is no way to purchase the compilers or tools individually. So if you have RPG at V5R1 or higher, you must have Development Studio and hence are entitled to Development Studio Client.

For consultants who do not have an iSeries of their own, but still wish to have the client tools, Development Studio Client is also made available as a passport advantage product so it can be purchased "off the shelf" from IBM Direct.

Development Studio has been a huge success, with over 80,000 licenses sold. Just as every development machine used to have PDM and SEU, every development machine will now have all the modern Application Development tools from IBM. This ubiquity is especially important for business partners who build and sell software. These Business Partners are now free to build software using any of the technologies or tools in Development Studio, and can assume their customers will have the tools required to tailor everything from RPG to Java and Web user interfaces. This effectively raises the lowest common denominator to a level unparalleled by any other operating system.



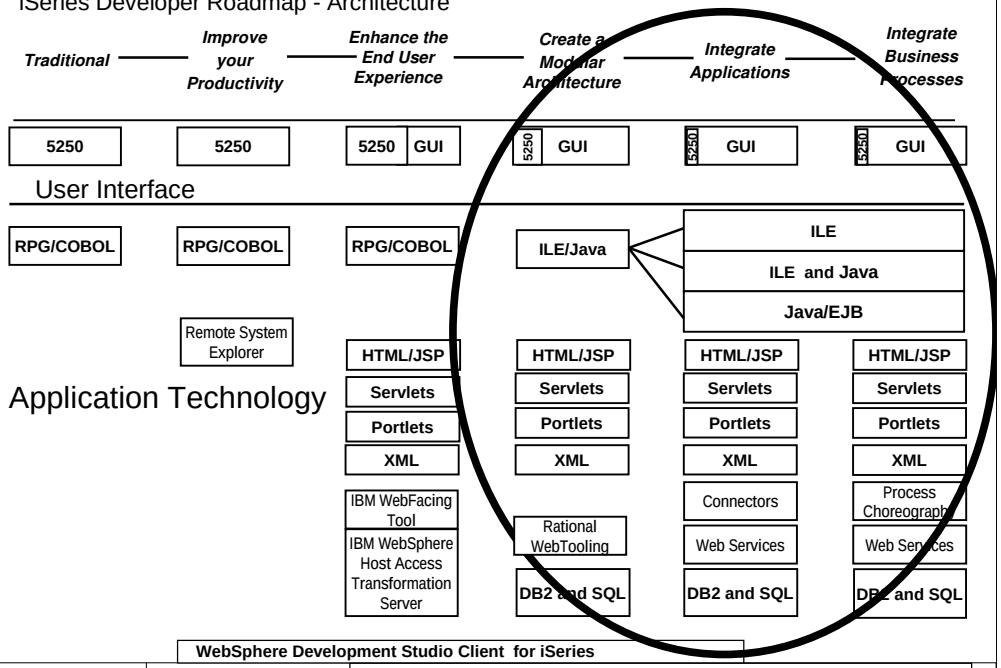
There is now only one application development product sold by IBM, for iSeries, as of V4R5. This is WebSphere Development Studio (Development Studio), which includes all four host compilers, all traditional tools (ADTS = PDM+SEU+SDA+RLU+DFU+AFP+CGU), and unlimited licenses of the workstation-based toolset named WebSphere Development Studio Client (formerly WebSphere Development Tools).

If you are an existing customer who has a subscription, you can upgrade to Development Studio free of charge. Without a Software Subscription, there is an upgrade fee. New licenses of Development Studio are priced very competitive compared to the combined prices of all constituent products. As of V5R1, there is no way to purchase the compilers or tools individually. So if you have RPG at V5R1 or higher, you must have Development Studio and hence are entitled to Development Studio Client.

For consultants who do not have an iSeries of their own, but still wish to have the client tools, Development Studio Client is also made available as a passport advantage product so it can be purchased "off the shelf" from IBM Direct.

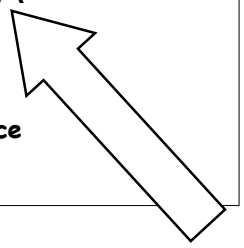
Development Studio has been a huge success, with over 80,000 licenses sold. Just as every development machine used to have PDM and SEU, every development machine will now have all the modern Application Development tools from IBM. This ubiquity is especially important for business partners who build and sell software. These Business Partners are now free to build software using any of the technologies or tools in Development Studio, and can assume their customers will have the tools required to tailor everything from RPG to Java and Web user interfaces. This effectively raises the lowest common denominator to a level unparalleled by any other operating system.

iSeries Developer Roadmap - Architecture



AGENDA

- Overview WDSc 6.0.1 and the AD Roadmap
- Service Oriented Architecture SOA
- WebServices what are they ?
  - **Provide a RPG/COBOL program as a WebService**
  - **Call a WebService from RPG/COBOL**



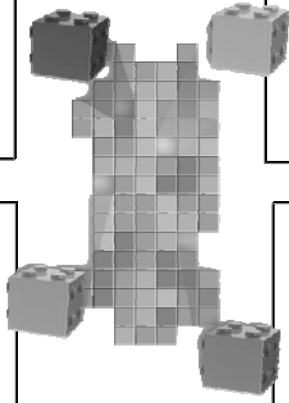
So simply put, what is it? Service Orientation? SOA?

… a service?

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

… service orientation?

A way of integrating your **business as linked services** and the outcomes that they bring



… service oriented architecture (SOA)?

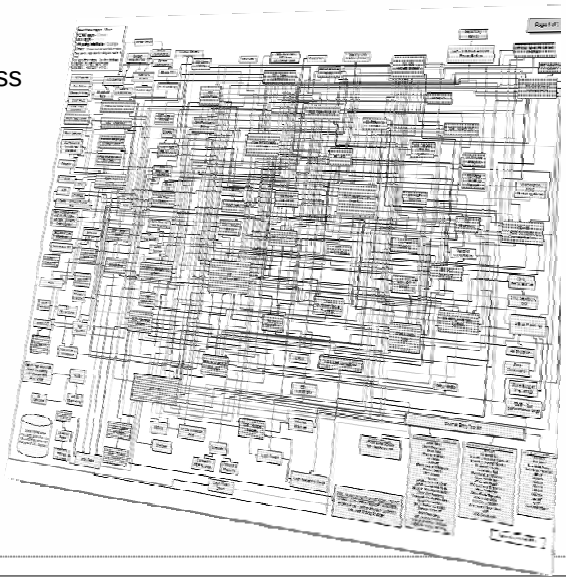
An IT **architectural style** that supports service orientation

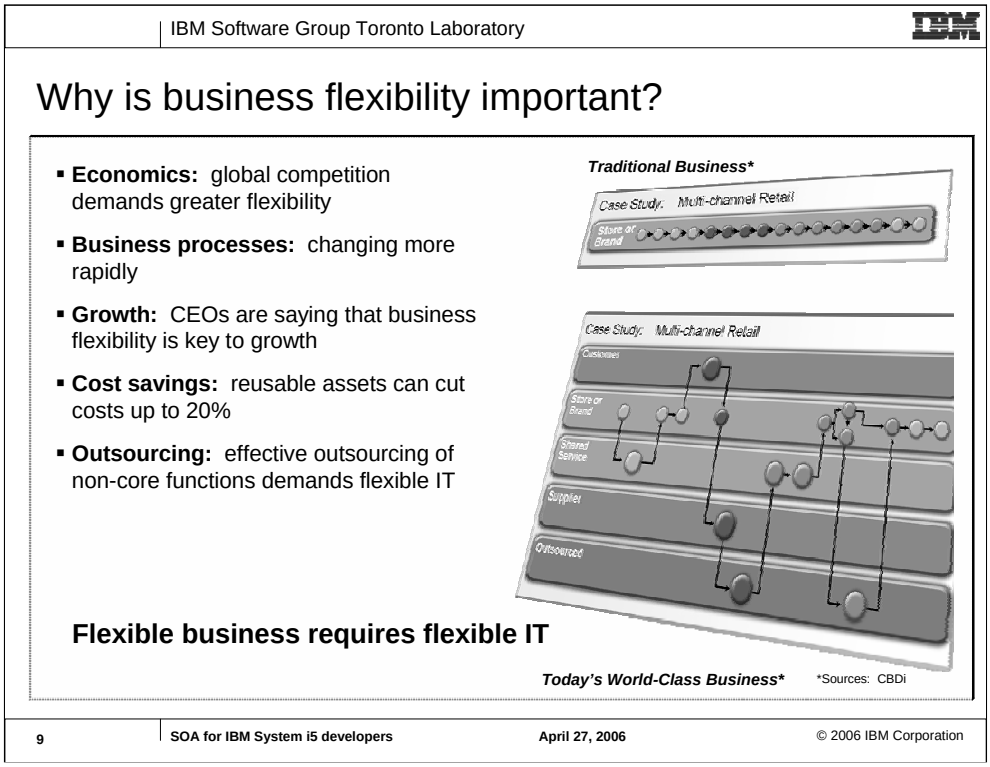
… a composite application?

A set of **related & integrated** services that support a business process built on an SOA

What are the barriers to business flexibility?

- No standards on business process – need best practices
- The inability to reuse
- Architectural policy
- Point application buys
- Infrastructure built over time, no roadmap





Why is SOA so important? Several reasons:

Economics. Globalization has made the world marketplace more competitive. So in order to be more competitive, companies need to be more flexible.

Rapidly changing environments. As the world changes ever faster, company business processes need to change simply to keep up.

Growth. Flexibility is at the top of the agenda of most CEOs simply in order to keep their companies growing.

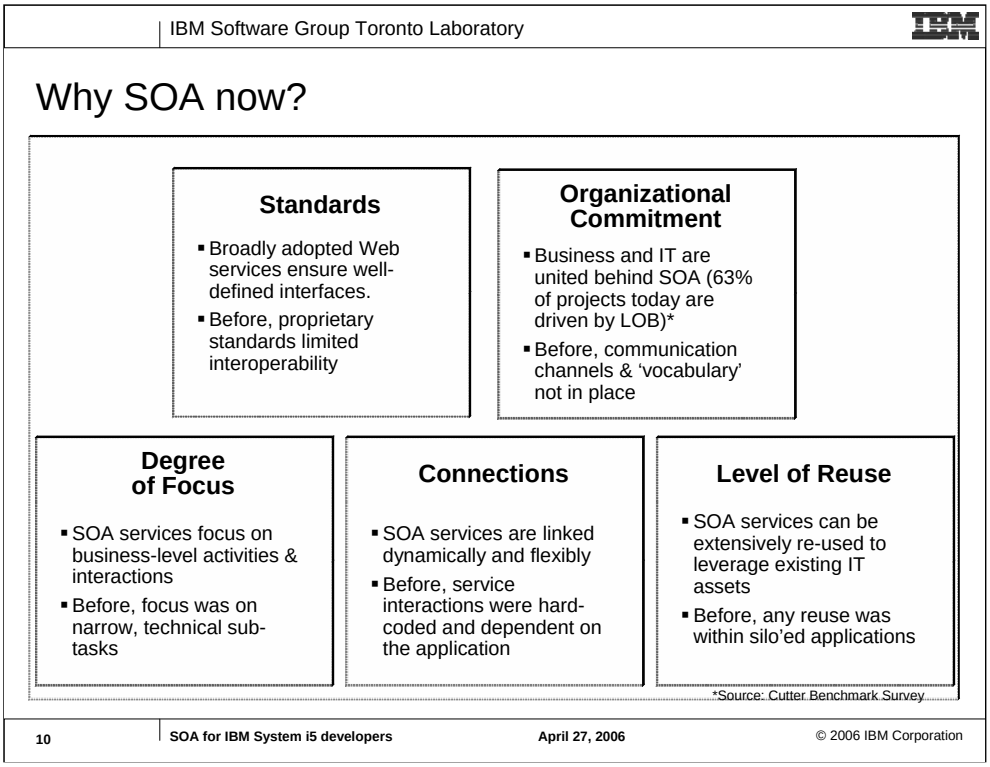
Cost savings. Reusable assets can cut cost by up to 20%.

Outsourcing

More and more companies are looking for more effective ways to outsource non-core business services. SOA makes it much easier to interconnect business services across different organizations.

Flexible business requires flexible IT

According to experts, SOA can help you reuse software assets and thus save time and money in IT costs. Jeffrey Poulin anfd Brent Carlson in Computerworld February 2, 2004



You've heard about things like re-use before so what is different about SOA than past approaches to IT?

•To begin with, SOA takes better advantage of standards than anything that has come before it. SOA uses broadly adopted Web services standards that ensure well-defined interfaces. In the past, standards exists but they didn't have the extensive buy-in that they do today. These amounted to proprietary standards limited interoperability

•Second, we see a much greater degree of organizational commitment to SOA. While in the past IT and business lacked a common way to communicate about designing architectures, we see SOA acting as a unifying force between business and IT.

•<Note: Do not spend as much time with the next 3 sections>

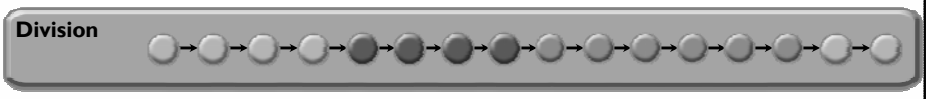
•Third, the degree of focus is different. You may have heard people call this "level of granularity" or "degree of abstraction". In any case, there is more of a one-to-one relationship between steps in a business process and the services that support these steps.

•SOA services are linked dynamically and flexibly while past service interactions were hard-coded and dependent on the applications where they resided.

•And finally, looking at the level of reuse, SOA services can be re-used extensively regardless of whether they're newly created services or existing IT assets that have been converted to services.

Traditional Business Process

Case Study: Order to Cash Process



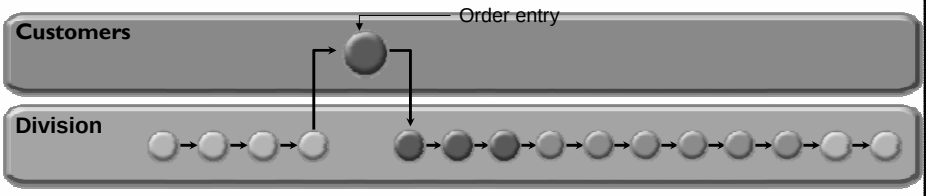
- Business process is embedded in three separate applications
- Business functions are tightly coupled within applications
- Business functions have unique and proprietary interfaces, restricting re-use
- Manual steps introduce functional gaps in the process
- Process cannot be easily measured and managed
- Changes to the process are difficult to implement
- New processes which are designed this way require long development cycles

Result: Business cannot operate on demand

On Demand Flexibility: Customer Self Service

Case Study: Order to Cash Process

Change: Allow customers and partners to order and manage accounts on-line



•Customers now order online using a web browser and the Internet

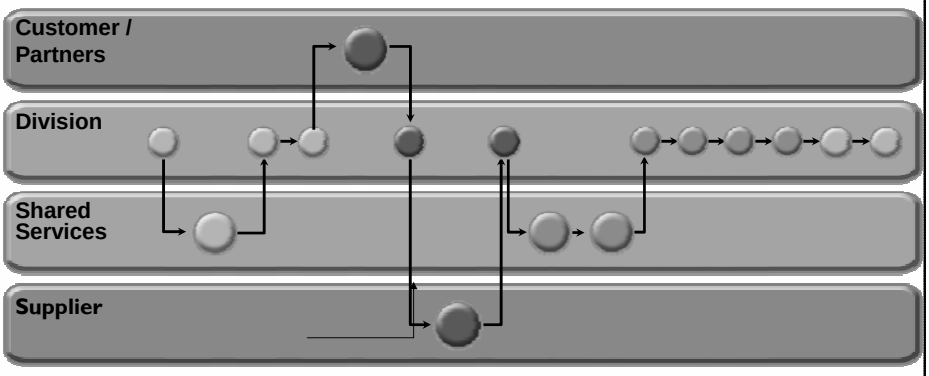
•Business partners can order using a web service call from their own process

•Customers are better served

On Demand Flexibility: Vendor Managed Inventory

Case Study: Order to Cash Process

Change: Minimize or eliminate inventory management function



Inventory Management

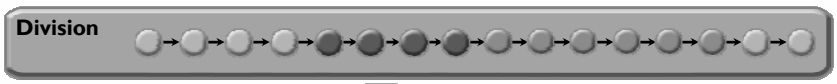
•Minimize or eliminate inventory management function

•Costs are reduced because less inventory is needed

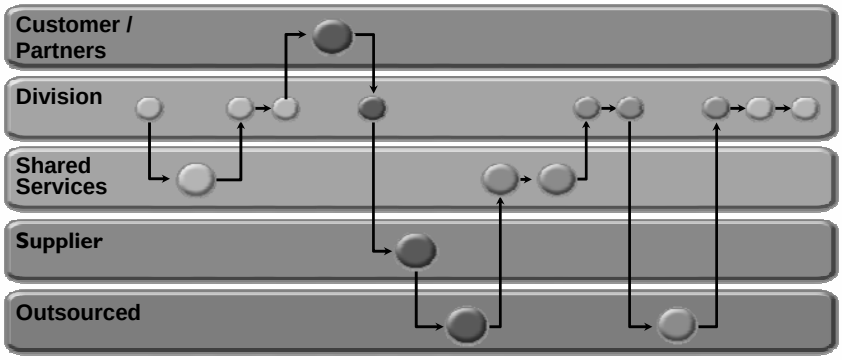
•Inventory servicing is better because of supplier integration with the process

Implementing a Service Oriented Architecture?

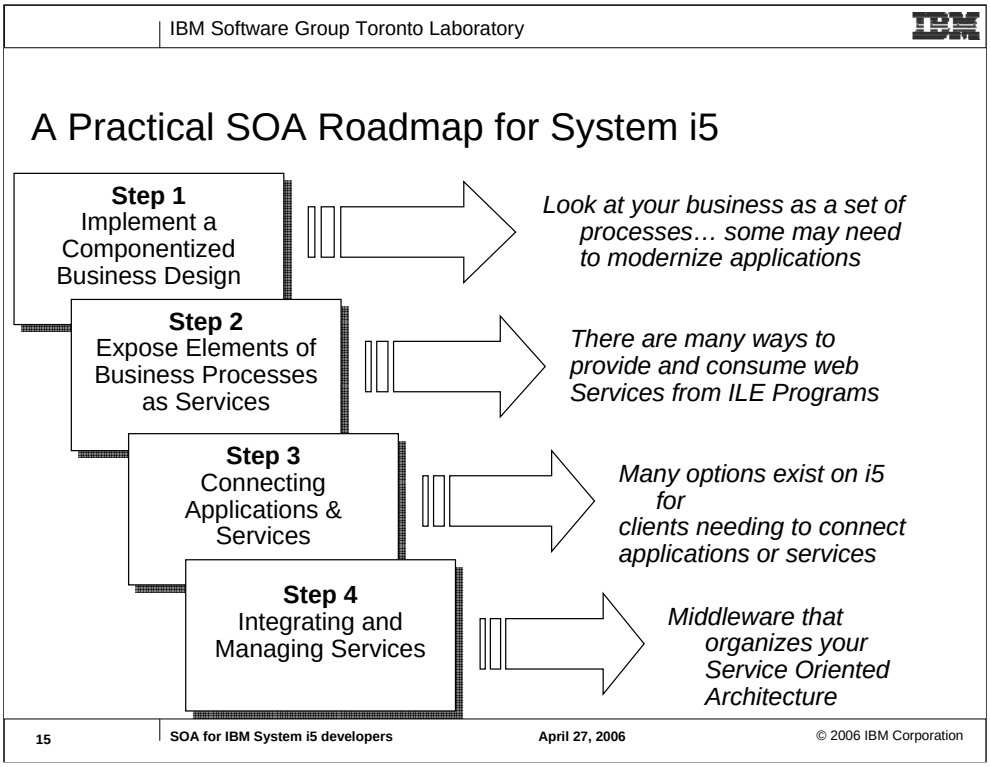
So how do you go from this environment…



To this environment?



There are 4 Steps to Implement SOA on System i5



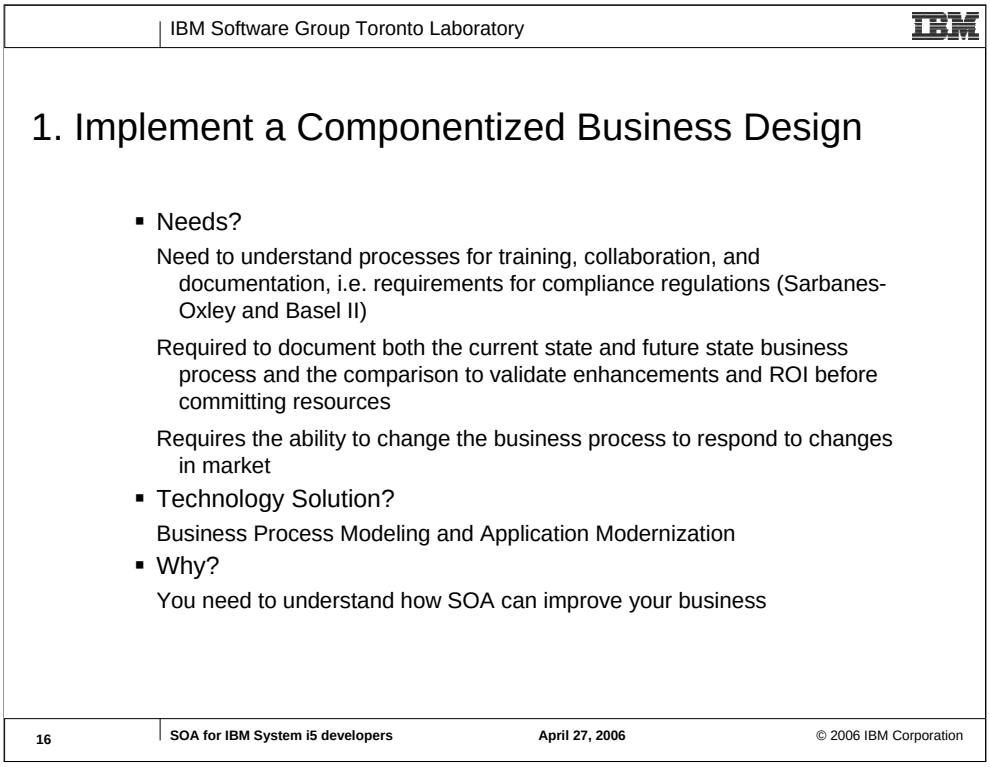
It's important to take a consultative approach with clients and have a basic understanding of what they get with iSeries:

Step 1: The key to any SOA implementation is to understand your business processes and the applications you use to achieve them. On iSeries many legacy RPG applications need to be modernized, iSeries has tools like Databorough's x-Analysis to accomplish this. Tools like this help RPG applications move easily to Model-View-Controller making easier for step 2. As with any roadmap you need alternatives… if a customer doesn't have time to do this they can use HATS to buy them time. HATS will allow them to use what they have a deploy web services without changing apps.

Step 2: The key here is to get iSeries customers to embrace web services. The more they embrace web services the more they'll need to manage them giving us the Step 3 and 4 upsell. Customers need to know that there are 6 ways to provide web services directly from there RPG apps using WebSphere Development Studio Client. iSeries customers cam also consume web services from RPG using a special Apache AXIS client shipped with i5/OS. HATS and EGL are tools within WDSc that will help them here.

Step 3: Again they need to hear the whole story. Customers can connect applications using data queues native to i5/OS. However when other heterogeneous are in the picture WebSphere MQ would be the play. IF the customer has Java experience they need to know the JMS capability of WAS. Finally larger customers will be more open to WebSphere ESB in an LPAR. We're currently working on getting WESB into plan for i5/OS.

Step 4: The audience should be familiar with this. As customers deploy and consume web services they'll need tools to mange this environment. Medium sized customers will be more inclined to use WBI Server Express Plus, Enterprise would use the SOA Foundation tools.



Needs?

We need to expose our internal stock purchase application over the Web to some of our most important business partners who will allow their employees to purchase stock from our company. We need an easy way to provide application to application integration over the Web.

Technology Solution?

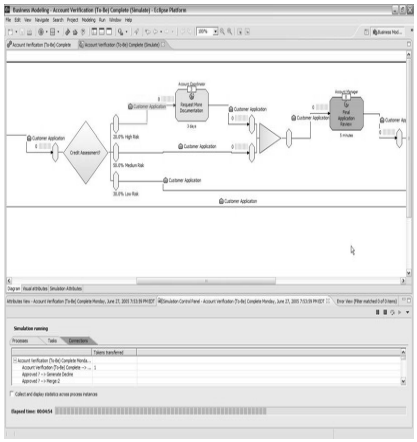
Web Services

Why?

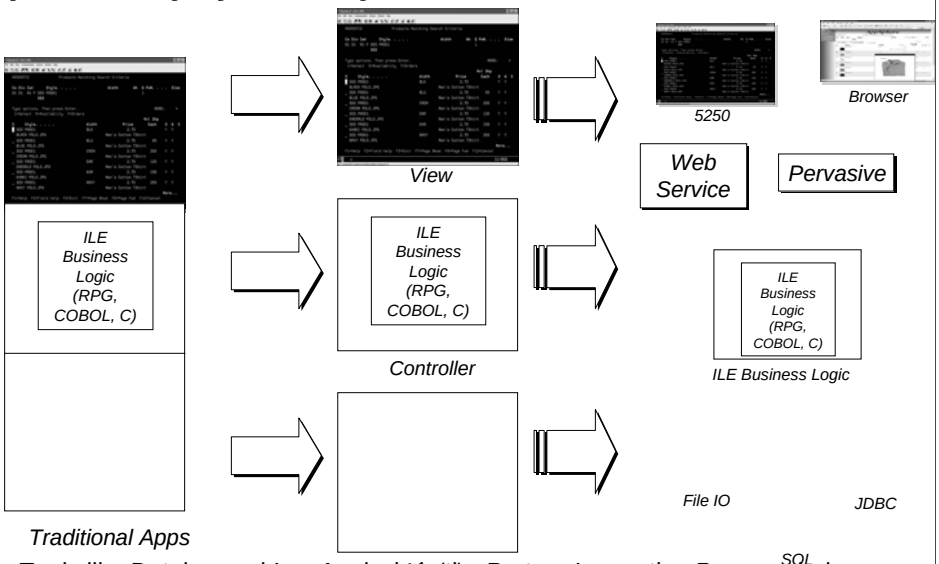
Web Services provide application-to-application integration and are the first integration technology that has been agreed upon by all major vendors including Microsoft. Another very important aspect of Web services is that they can communicate using ubiquitous network protocols that everyone has – SOAP over HTTP. Note that other protocols can be used such as SOAP over JMS but when exposing applications over the internet to the widest variety of users the most likely implementation will be SOAP over HTTP.

1. Implement a Componentized Business Design

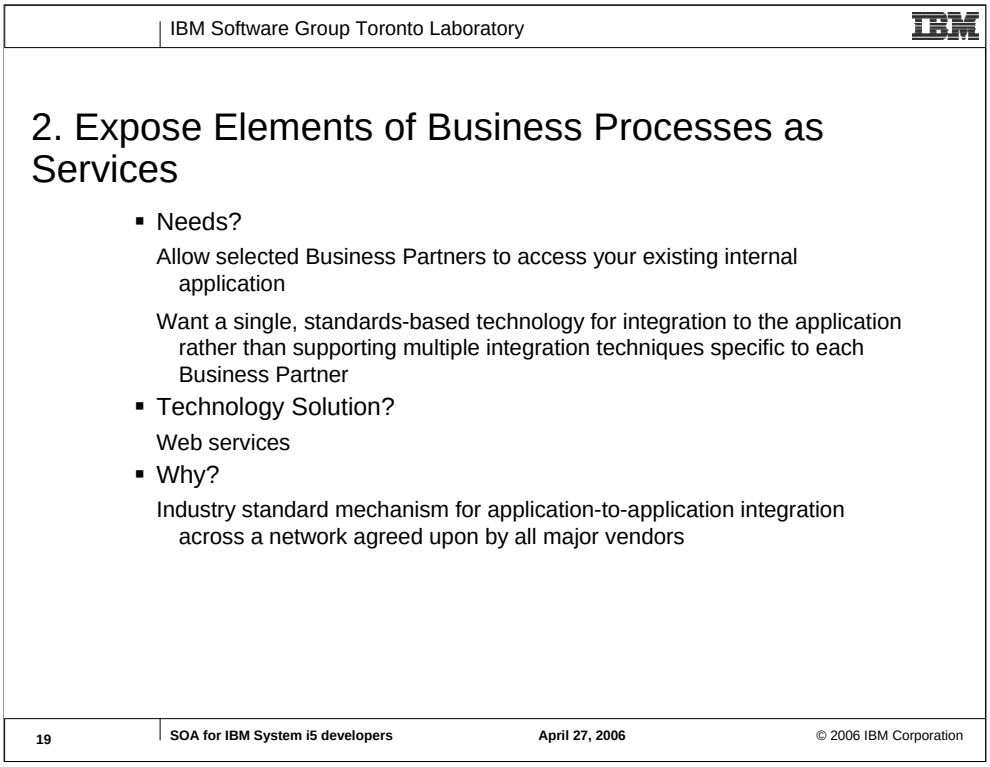
- Map the business processes in the organization
  - Identify the applications that make up these processes
  - Determine who consumes these applications (People or other Applications)
  - **WebSphere Business Modeler** can model and simulate business processes without writing code



By the way, you may have to Modernize…



- Tools like Databorough's x-Analysis in the Partner Innovation Program help



Needs?

We need to expose our internal stock purchase application over the Web to some of our most important business partners who will allow their employees to purchase stock from our company. We need an easy way to provide application to application integration over the Web.

Technology Solution?

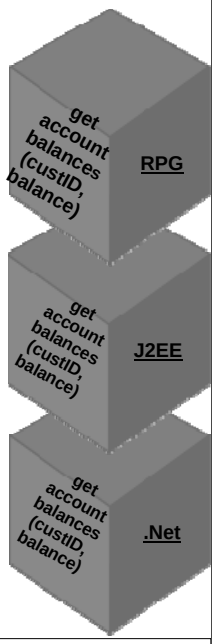
Web Services

Why?

Web Services provide application-to-application integration and are the first integration technology that has been agreed upon by all major vendors including Microsoft. Another very important aspect of Web services is that they can communicate using ubiquitous network protocols that everyone has – SOAP over HTTP. Note that other protocols can be used such as SOAP over JMS but when exposing applications over the internet to the widest variety of users the most likely implementation will be SOAP over HTTP.

What is a Web Service?

- **Web Services provide business function (get stock quote, check order status, check credit rating, etc.) that can be described, published, located, invoked over a network**
  - Built on established, ubiquitous Internet standards (SOAP/HTTP)
    - Supports other protocols providing higher QoS (SOAP/JMS, etc.)
  - Platform and implementation neutral providing true interoperability
  - Service Consumers need to know the Interfaces to web services but not the implementation details of services
- **Web Services allow businesses to:**
  - Describe application functionality (services) without regard to implementation details
  - Publish those services
  - Discover other services they wish to use
  - Connect and invoke those services



Web Services are self-contained, modular applications that can be described, published, located, and invoked over a network. Web services can perform business functions (services) such as getting a stock quote, checking the status of an order, determining a credit rating, etc.. These business services can be new applications or just wrapped around existing legacy systems to make them network-enabled.

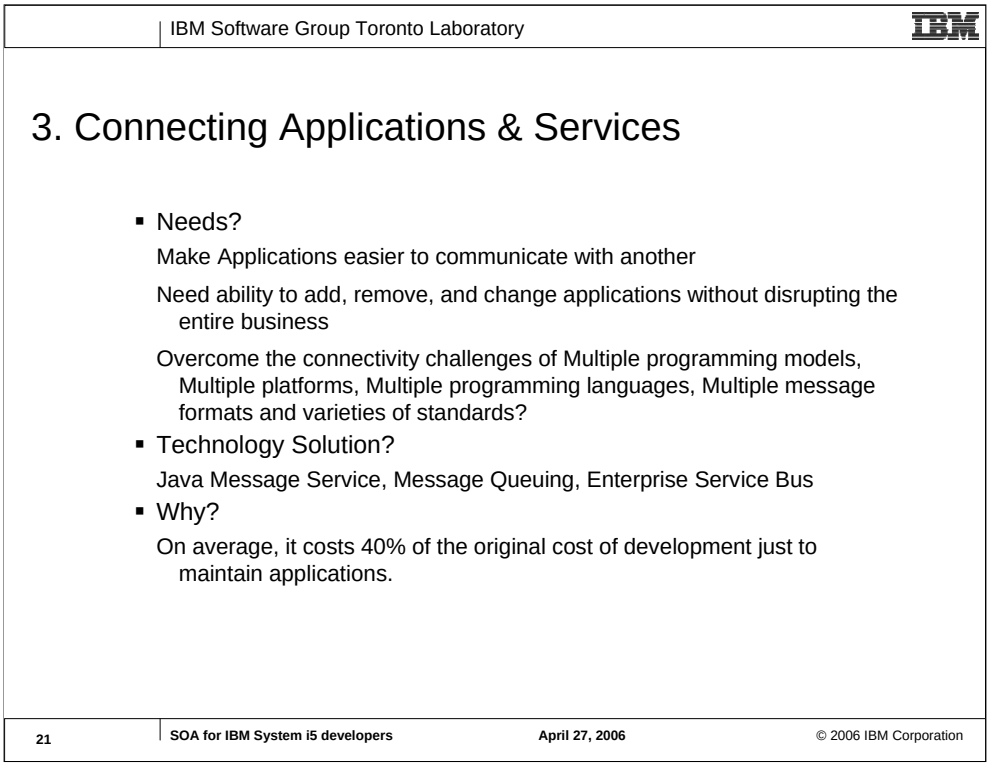
Unlike earlier integration solutions Web services are built on ubiquitous internet standards such as HTTP and XML that are readily available to both customers and their partners. This means that customers don't have to force their partners to use expensive proprietary solutions thus reducing the barriers to integration.

Web services are platform neutral, not requiring specific hardware or operating systems which also makes them vendor neutral providing customers with greater flexibility and freedom of choice. They are also implementation neutral and do not require a specific programming language or programming model – for example they can be implemented in COBOL accessing a CICS application, using C# in the .Net programming model, in Java using the J2EE programming model or a variety of other languages.

Another important Web services concept is that service consumers (those using your service) do not need to know the implementation details of the service. The application using your service does not need to know if the service is implemented using .Net or J2EE or another technology and if the service provider changes implementation technologies in the future it does not affect the users of the service. Likewise the user of the Web service can change his application without requiring the service provider to make any changes. This leads to lower future maintenance costs to both the service provider and service consumer.

Web Services allow businesses to:

Describe application functionality (services) without regard to the implementation details so an application using the service doesn't need to know if the service is implemented in Java, C# or any other language, the requesting service only needs to know the input required by the service provider and the output that will be returned – this is why we say that Web



Needs?

We need to expose our internal stock purchase application over the Web to some of our most important business partners who will allow their employees to purchase stock from our company. We need an easy way to provide application to application integration over the Web.

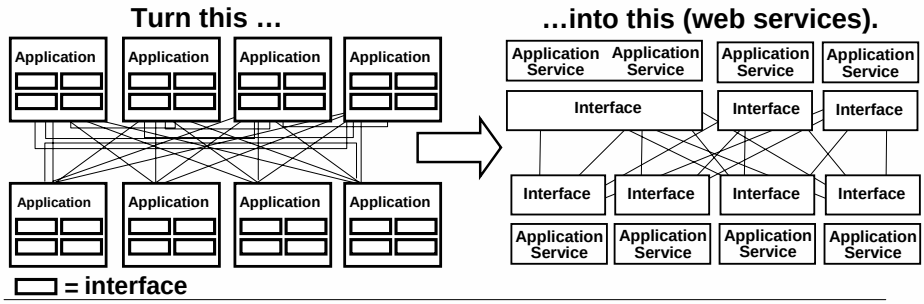
Technology Solution?

Web Services

Why?

Web Services provide application-to-application integration and are the first integration technology that has been agreed upon by all major vendors including Microsoft. Another very important aspect of Web services is that they can communicate using ubiquitous network protocols that everyone has – SOAP over HTTP. Note that other protocols can be used such as SOAP over JMS but when exposing applications over the internet to the widest variety of users the most likely implementation will be SOAP over HTTP.

Web Services are a good start…



- ✓ 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

But separate connection points still leaving bloated interfaces ….

So how does web services decouple the interfaces from their applications?

1) It uses a programming model that allows a rich abstraction of both the business app and the interface.

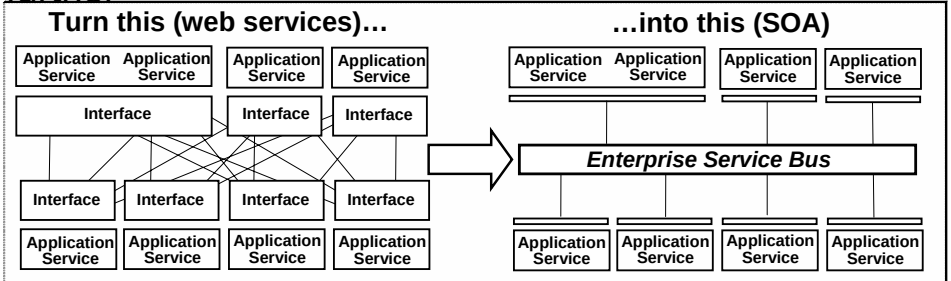
2) By abstracting the interfaces, they can be clearly separated from the business applications.

3) This enables you to reduce the number and complexity of those interfaces and.

4) It allows you to reuse both the interfaces and the business applications.

The problem is that you still have to build, find, and manage all of those interfaces somewhere.

The Enterprise Service Bus shrinks those interfaces further



- ✓ Decouples the point-to-point connections from the interfaces
- ✓ Allows for dynamic selection, substitution, and matching
- ✓ Enables more flexible coupling and decoupling of the applications
- ✓ Enables you to find both the applications and the interfaces for re-use

RESULT → Greater Business Responsiveness

The ESB shrinks the interfaces further.

1) It virtualizes the interface, or in other words, it decouples the point-to-point connections from the interfaces themselves.

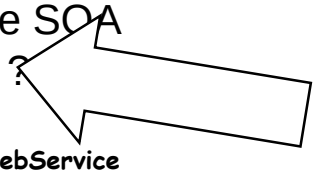
2) The interfaces are put into a third party broker which helps you manage the interfaces better.

3) The enables faster and more flexible coupling and decoupling of applications.

4) Because you can find all of the applications and the interfaces, you can then reuse both.

AGENDA

- Overview WDSc 6.0.1 and the AD Roadmap
- Service Oriented Architecture SOA
- WebServices what are they ?
  - **Provide a RPG/COBOL program as a WebService**
  - **Call a WebService from RPG/COBOL**



What Are Web Services

- Function that can be programmatically invoked over a network
- Basically remote procedure calls built on open standards and proven technologies
  - Lots of new standards around Web services to ensure interoperability in heterogeneous environments
  - Underlying implementations built on proven technologies like
    - XML
    - HTTP
    - Messaging middleware
    - Security standards

Why use Web Services

- Because:
  - They are the latest hype
  - They are hot
  - They are cool
  - They are useful
  - Somebody asks you to provide one for a certain task
  - Somebody provides one and asks you to use it

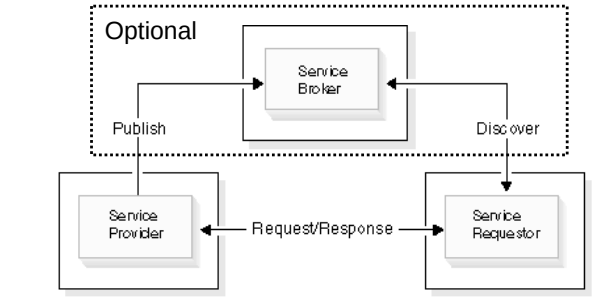
Real World Examples

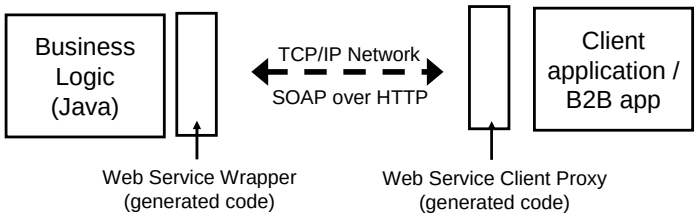
- Shipping and receiving
  - FedEx, UPS
  - Quotes
  - Track parcels in transit
  - Schedule pickups
- Address validation
  - UPS
- Credit Checks
- Stock quotes
  - Everyone
- Commerce
  - eBay, Amazon

Integration

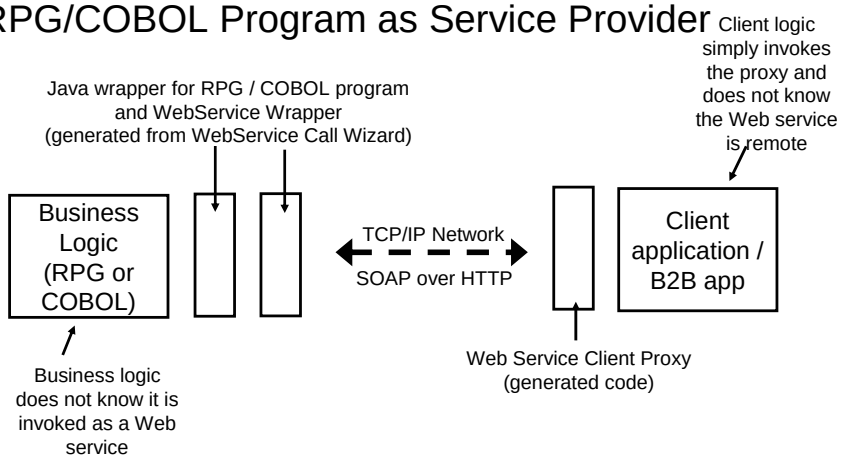
- **How do Web services enable better integration?**
  - **Based on open standards**
    - **The protocols are well defined**
  - **Heterogeneous**
    - **Not tied to a specific hardware or software platform**
      - **Java and .NET**
  - **Dynamic**
  - **Loosely coupled**
  - **Build on top of existing knowledge, frameworks and code libraries**
    - **XML**
    - **Security**
    - **Transportation**

Overview





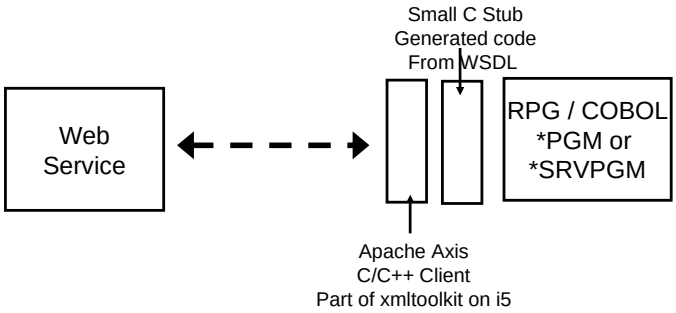
RPG/COBOL Program as Service Provider



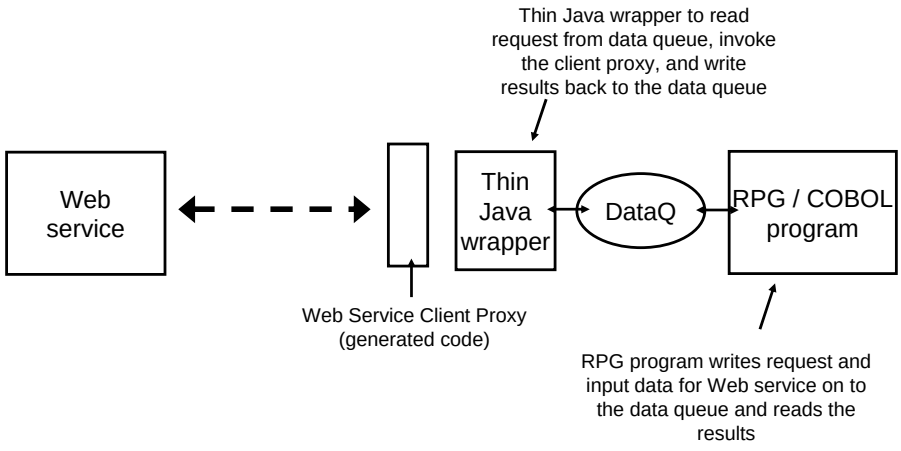
RPG Calling Web Services

- So far we have focused on Java Web services and invoking an RPG program / service program as a Web service
- But what about calling a Web services from RPG?
  - Apache C / C++ Axis client
    - or
  - Generate a Java client proxy to the Web service then invoke Java proxy from RPG using:
    - Data Queue
    - Java Method Calls from RPG
    - Any other form of RPG -> Java communication

RPG Calling Web Services – Axis Client



RPG Calling Web Services – Java Client



Technologies used

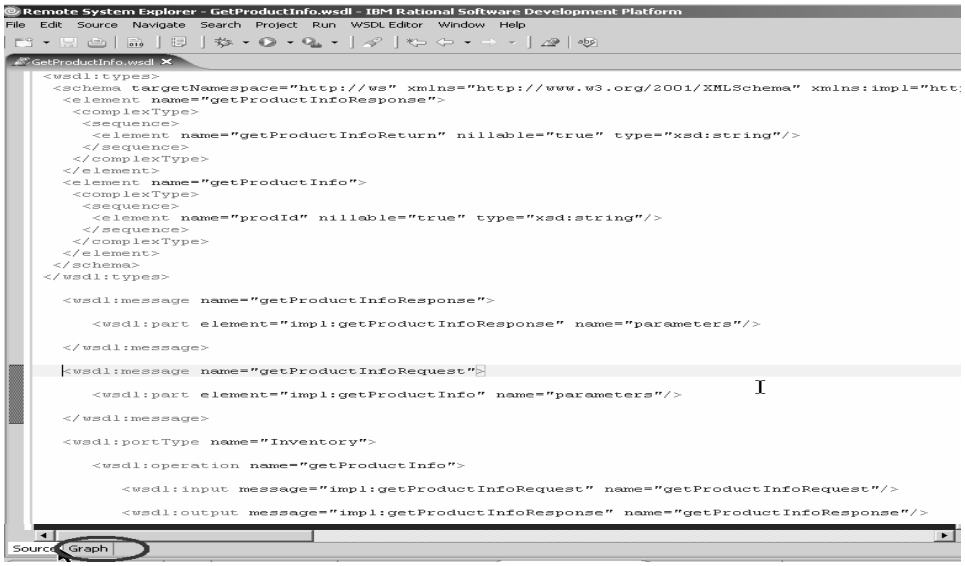
Web Services – Acronym Soup

- WSDL
  - Web Services Description Language
  - Standard way to describe the interface to a Web Service
- SOAP
  - Simple Object Access Protocol
  - Standard way to exchange information during Web Service invocation

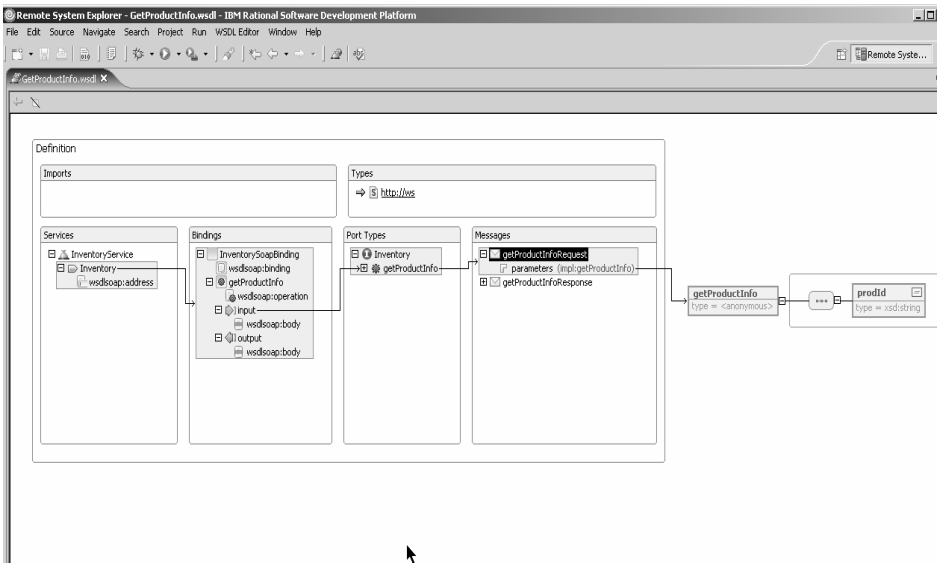
Web Services Description Language (WSDL)

- Web Services Description Language
  - Standard for describing networked, XML-based services
  - Created by Web service provider to describe each Web service
  - This single document tells client everything it needs to know in order to invoke the Web service:
    - Name of the Web service
      - Input and output parameters
    - Binding
      - SOAP
      - HTTP GET / POST
      - MIME (Multipurpose Internet Mail Extensions)
    - Transport mechanism
      - HTTP
      - Message queues (i.e. Java Message Service (JMS))
      - FTP
      - SMTP (email)
    - Address information

WSDL editor in WDSc

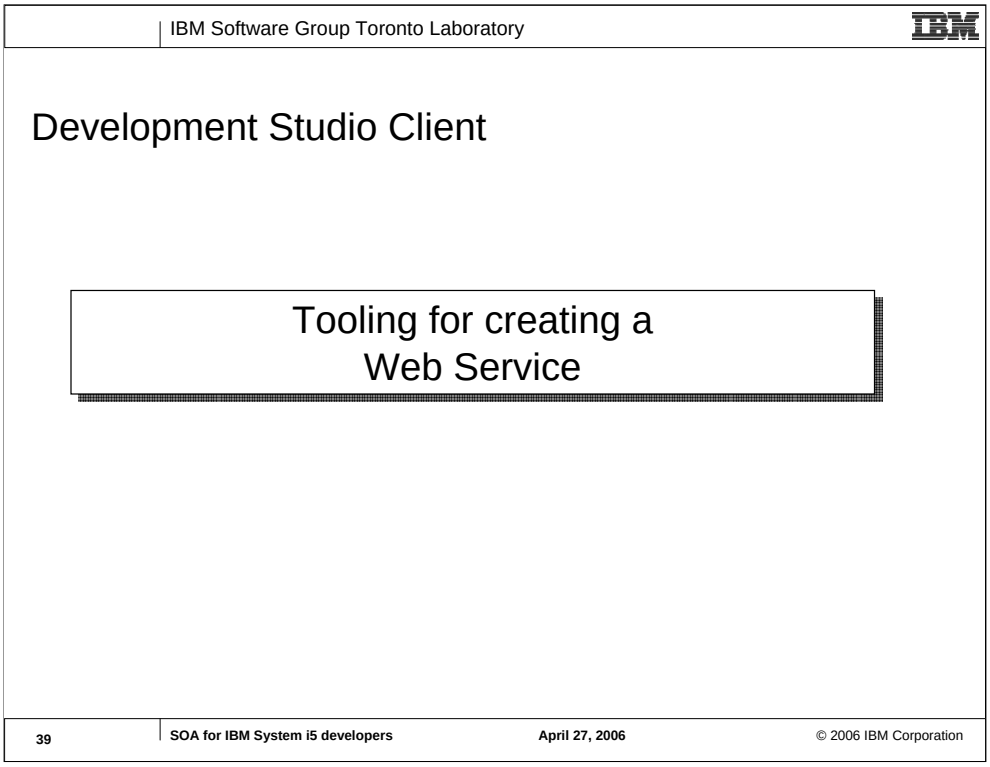


WSDL editor in WDSc



Simple Object Access Protocol (SOAP)

- Simple Object Access Protocol
  - Protocol for exchanging information in a distributed environment
  - SOAP messages are sent from sender to receiver (XML documents)
  - Independent of transport layer
    - SOAP over HTTP
    - SOAP over message queues
  - SOAP message contains
    - Envelope
      - Request – what to invoke and parameters
      - Response – results from Web service
    - Encoding rules
      - How application-defined datatypes are encoded in message
    - Communication styles
      - Remote procedure call
      - Message-oriented (or document-oriented)
        - Incoming request is an XML document



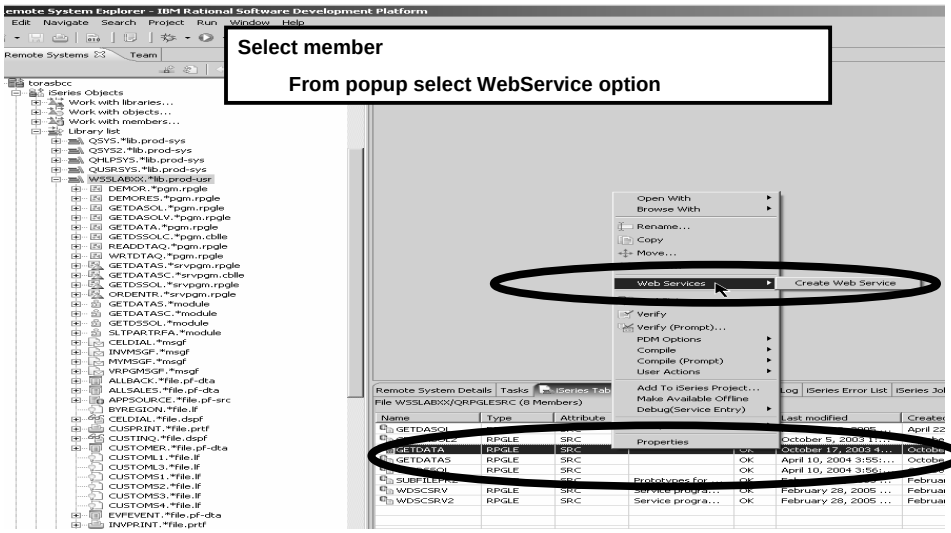
ADTS has been the traditional method for developing and maintaining server-side iSeries applications. WebSphere Development Studio Client for iSeries, V5.0, includes new highly integrated and highly extendible tools for iSeries RPG, COBOL, C, C++, CL and DDS development. These new tools offer programmers a development experience that is consistent with the experience for developing Java, Web, Web Services, and XML applications, lowering the learning curve for all.

These new generation tools include the Remote System Explorer for a PDM-like experience, and iSeries projects for team-based development (together with a Eclipse-compliant software change management repository). They offer rich support for exploring the file system, compiling/building, editing, running, and debugging. The iSeries Projects support enables effective team support leveraging any iSeries or LAN resident source repository which supports Eclipse.

Tools

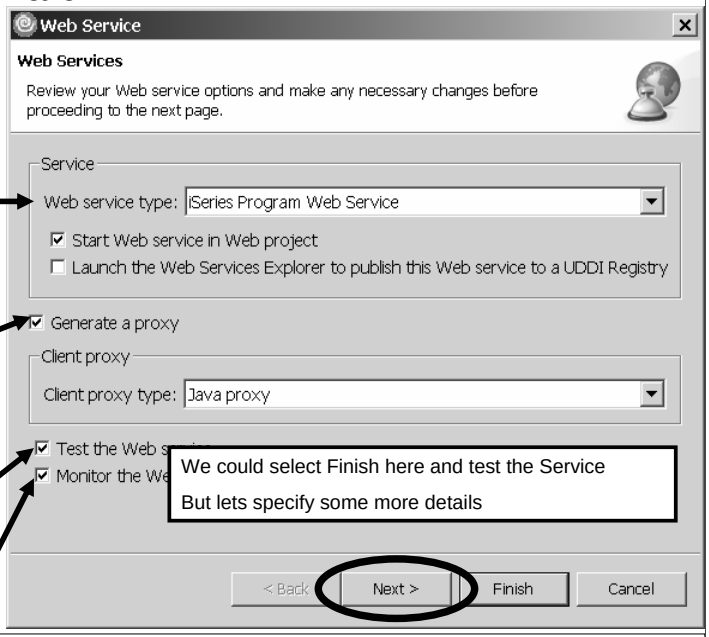
- **Web Services Tools in WDSc**
  - **Create**
    - **Web services from:**
      - **ILE RPG/COBOL source**
      - **Java Beans**
      - **Enterprise JavaBeans (advanced edition only)**
      - **SQL Queries**
      - **DB2 Stored Procedures**
      - **DB2 XML Extender calls**
    - **Web service client proxy given a WSDL document**
  - **Test**
    - **Test your Web services in the WebSphere test environment**
    - **Generate test code to test generated Web service client proxies**
  - **TCP/IP Monitor**
    - **Useful for debugging SOAP messages between provider and consumer**

WebService Wizard
Use Remote Systems Explorer to select member



Web Service Wizard

- Click Finish & Done!
- Drop-Dead Simple!



Default is iSeries Web Service type

The Java bean proxy will provide a remote procedure call interface to the Web service

We will test the Web service after it is created

We will monitor the SOAP traffic for this Web service

We could select Finish here and test the Service

But lets specify some more details

Configuring the Web Service- Edit Program/Parameters

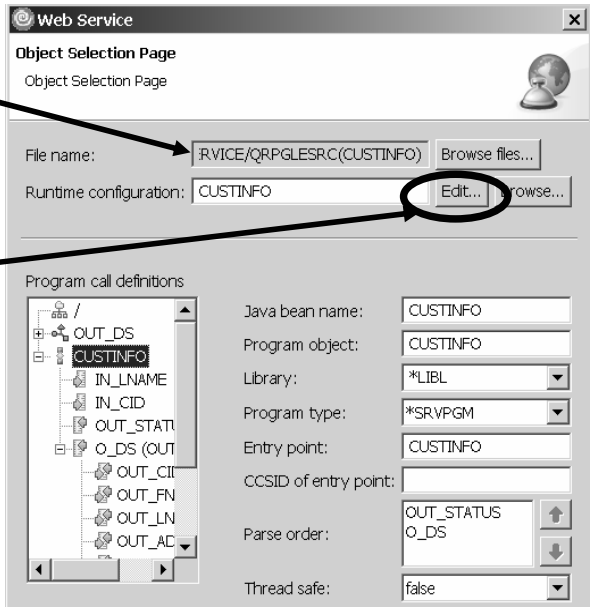
-File name shows the Program source file the wizard was launched from (valid types incl. ILE RPG, COBOL, PCML)

-**Browse files** to select a different file

-The default Runtime configuration is taken from the System i5 connection

-**Edit** to change the configuration

-**Browse** to choose an existing configuration (.config file)

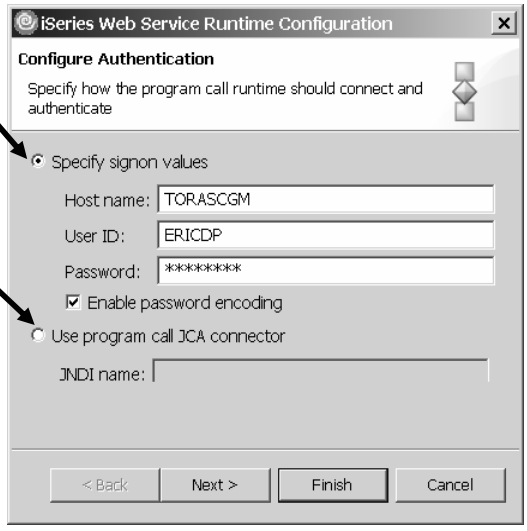


- Selecting the Program (**CUSTINFO**), you can change the Library, Program type, Program Object etc. (in most cases the defaults are sufficient)
- Expand the program to see program parameters
- Selecting a parameter (e.g. **IN_CID**), you can change Usage type, CCSID

Configuring the Web Service- Runtime Configuration

Signon information is defaulted from System i5 connection

Alternatively, can use the Program Call Java Connection Architecture connector for signon and connection management (advanced product only)



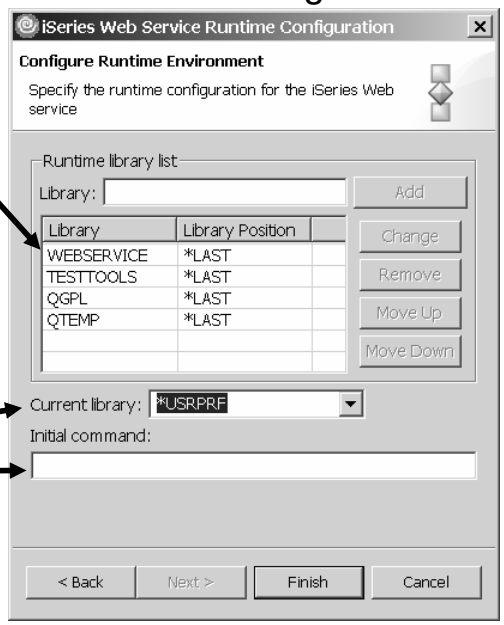
Configuring the Web Service- Runtime configuration

-Libraries required by the program defined here, and searched in this order

- The default Runtime library list information is taken from the System i5 connections' user libraries

-Specify the current library, or choose one of the options (*USRPRF leaves the current library set to the value defined in the user profile)

-In the Initial command field, specify a host command to run before the Web service is invoked

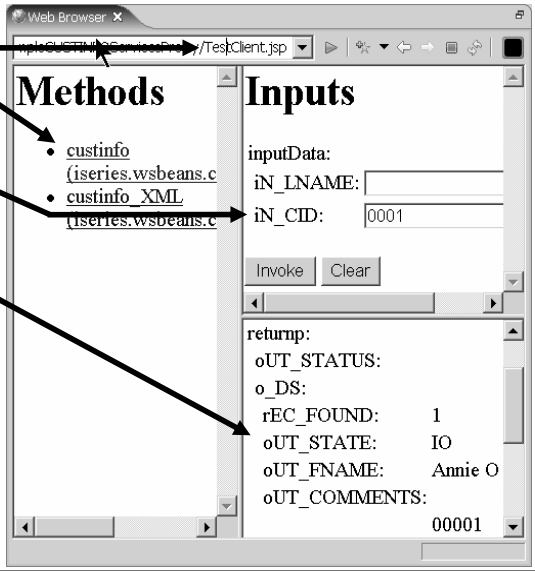


Testing the Web Service

After wizard completes, the generated Test Client JSPs are run on the server (selected option “Test the Web service” in the wizard)

Select one of the methods to test, then enter the input data, Customer ID “0001”

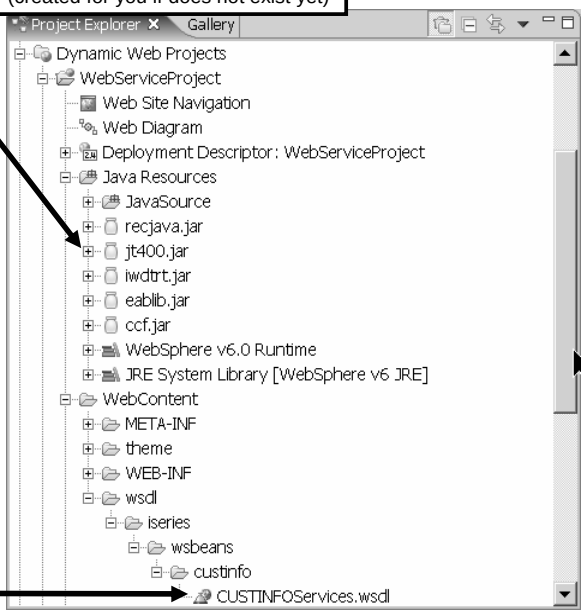
Click “Invoke” and the results of the Web service are displayed, Customer ID “0001” is associated with Annie O!



- As you will see shortly, there are many different ways to test the Web service

Generated artifacts

- The selected Service Project (created for you if does not exist yet)



-iSeries Toolbox for Java classes (jt400.jar) and Program Call Runtime jars

  -Used by the *Services classes to manage iSeries host connections and invoke iSeries programs

  -If IBM WebSphere is the Web service runtime, the following classes are also generated:

    - Classes for marshaling and unmarshaling SOAP data (*_Deser and *_Ser)

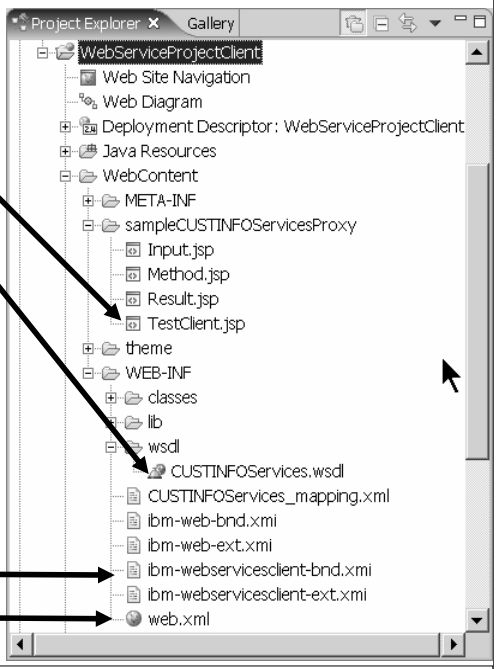
    -A Service Endpoint Interface (SEI) class- the Java interface corresponding to the Web service port type being implemented

- A WSDL file

  -This XML file fully describes the implemented Web service

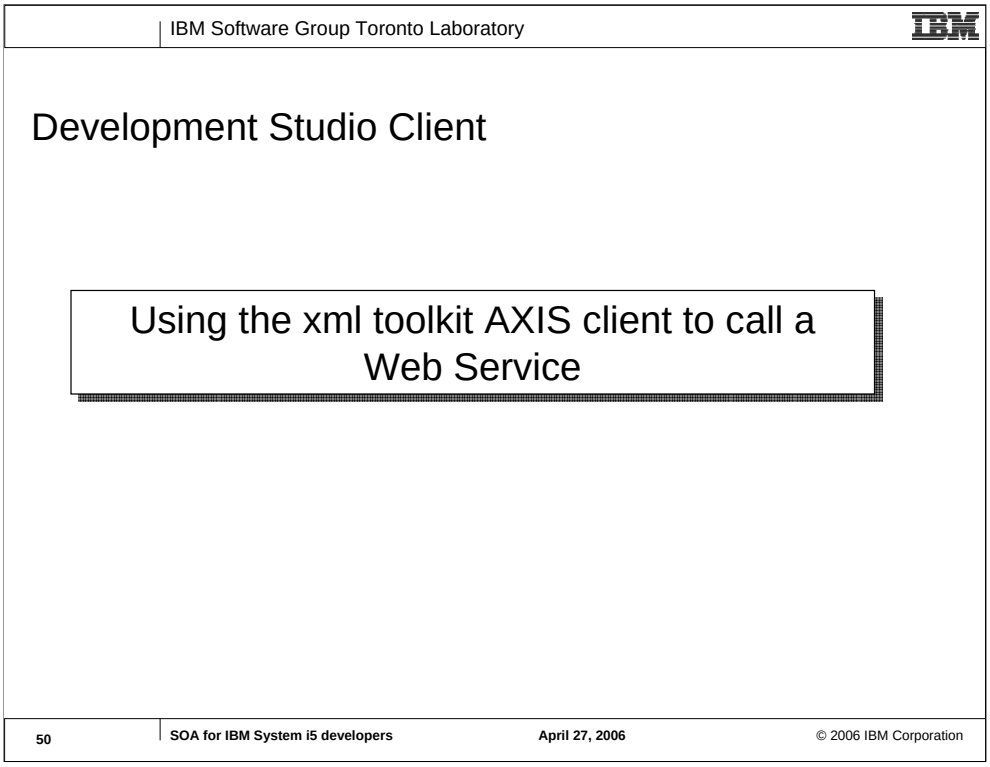
Generated artifacts – Test Client

- Test Client JSPs- run TestClient.jsp on server (contains Input, Method, Result JSPs)
- WSDL file for the Web service- provided by the implementer of the Web service (identical copy of WSDL in the Web service project)
- In the case of a Web service client, what is generated depends on the J2EE level of the project
  - If the project has a J2EE level of 1.4, modifications will be made to the existing deployment descriptor (web.xml) for that project
    - Edit the bindings and extensions with Deployment Descriptor editor (changes reflected in bindings and extensions XMI files)
  - If the project has a J2EE level of 1.3, webservicesclient.xml will be generated
    - Edit the bindings and extensions with Web Services Client Editor



Testing the Web Service more ways to do it!

- Saw earlier that can use the Web Service wizards' Test Page to test
  - Launches external browser (default) before wizard completes
- Also saw that can test via Test Client JSPs
  - launches the products internal browser (Full featured Internet Explorer) after wizard completes
- Foils at the end of this presentation show more testing capabilities using the WebService Explorer



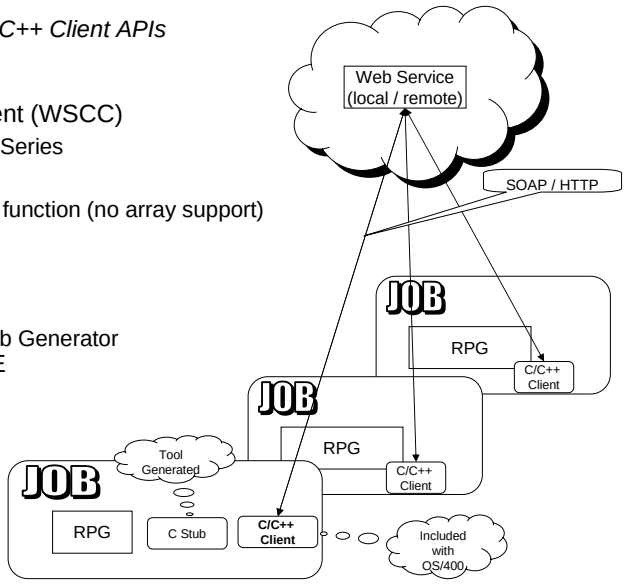
ADTS has been the traditional method for developing and maintaining server-side iSeries applications. WebSphere Development Studio Client for iSeries, V5.0, includes new highly integrated and highly extendible tools for iSeries RPG, COBOL, C, C++, CL and DDS development. These new tools offer programmers a development experience that is consistent with the experience for developing Java, Web, Web Services, and XML applications, lowering the learning curve for all.

These new generation tools include the Remote System Explorer for a PDM-like experience, and iSeries projects for team-based development (together with a Eclipse-compliant software change management repository). They offer rich support for exploring the file system, compiling/building, editing, running, and debugging. The iSeries Projects support enables effective team support leveraging any iSeries or LAN resident source repository which supports Eclipse.

RPG Service Requester

Apache AXIS Web Services C++ Client APIs

- Web Services C/C++ Client (WSCC)
  - 5733-XT1 XML Toolkit for iSeries
    - GA 8/2005
    - Tech Preview – limited function (no array support)
  - Open Source
    - Apache.org AXIS
  - Ported to i5/OS ILE
    - WSDL -> C++ or C Stub Generator
- Direct invocation from ILE
- Runs in the ILE Job

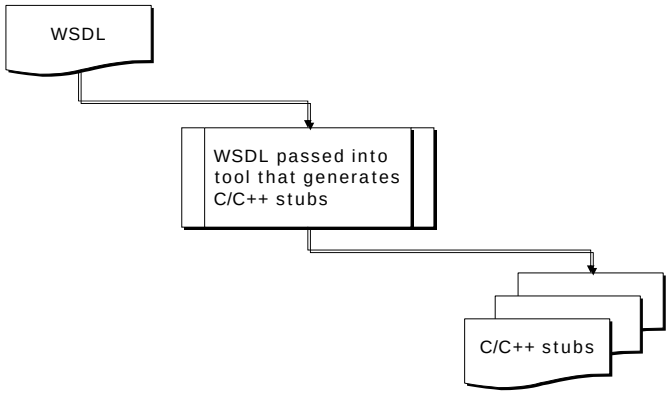


This new feature allows iSeries legacy applications to easily consume web services.

How WSCC Works

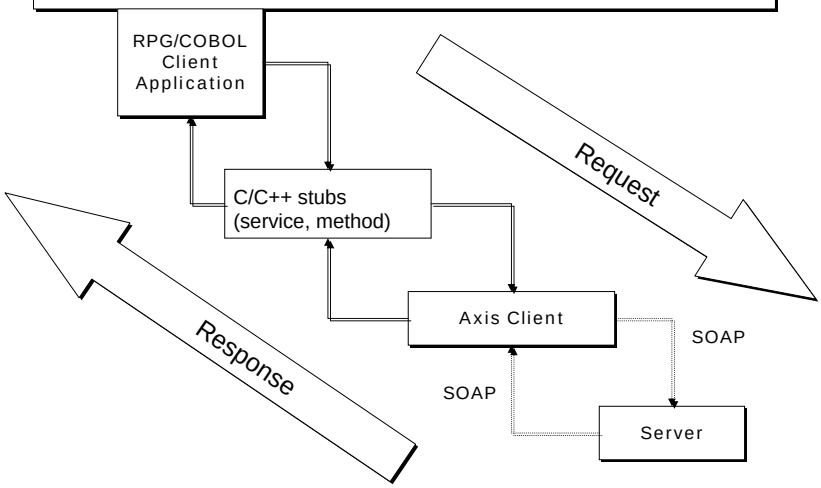
1. User generates stubs using java tool wsdl2ws.jar or shell script wsdl2ws.sh (preferred)

   Example (generates C stubs):
   wscc-1.0-OS400/bin/wsdl2ws.sh GetQuote.wsdl –lc -ogetQuoteStubs



How WSCC Works (cont.)…

2. User then creates application that uses the stubs to invoke Web service

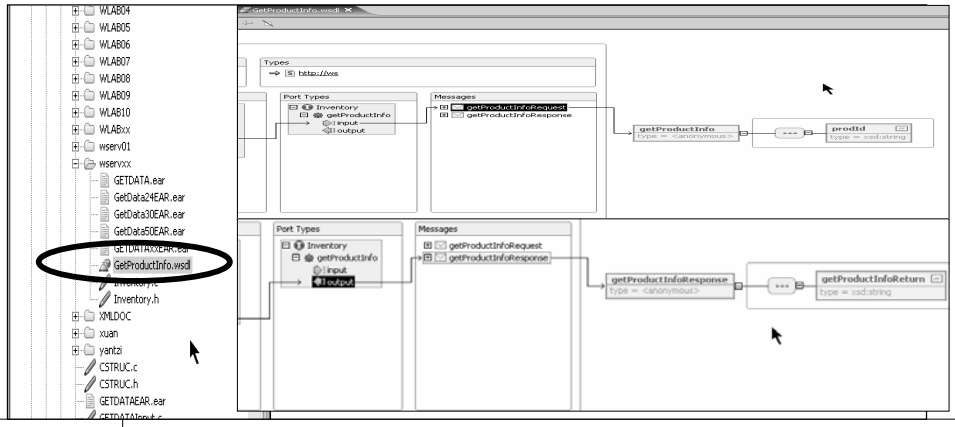


Example: How to call a WebService

- You get a WDSL that describes the following:
  - A WebService that
    - Excepts
      - Product number as a string
    - Returns XML document as a string with
      - Product name
      - Price
      - In stock info

RPG calling a WebService

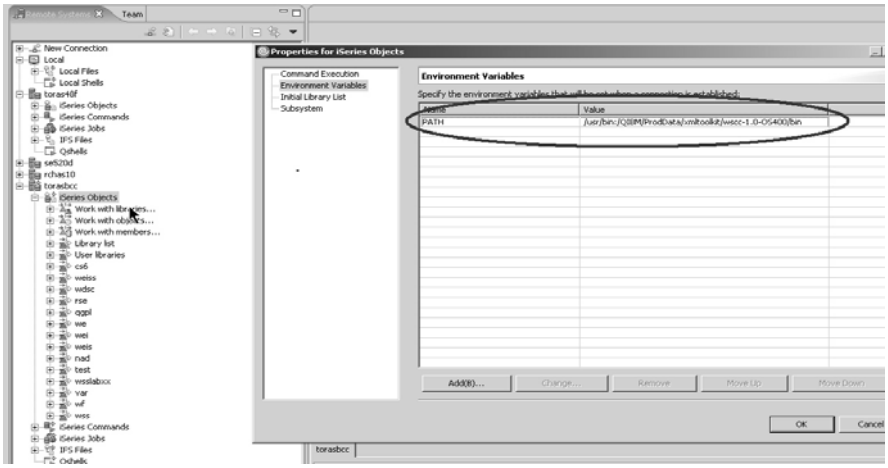
- Store WSDL on IFS
  - Run XMLtoolkit tool to extract C functions from WDSL
  - Write RPG prototypes
  - Write RPG logic to invoke C functions



Setup RSE environment for running tool in QSHELL

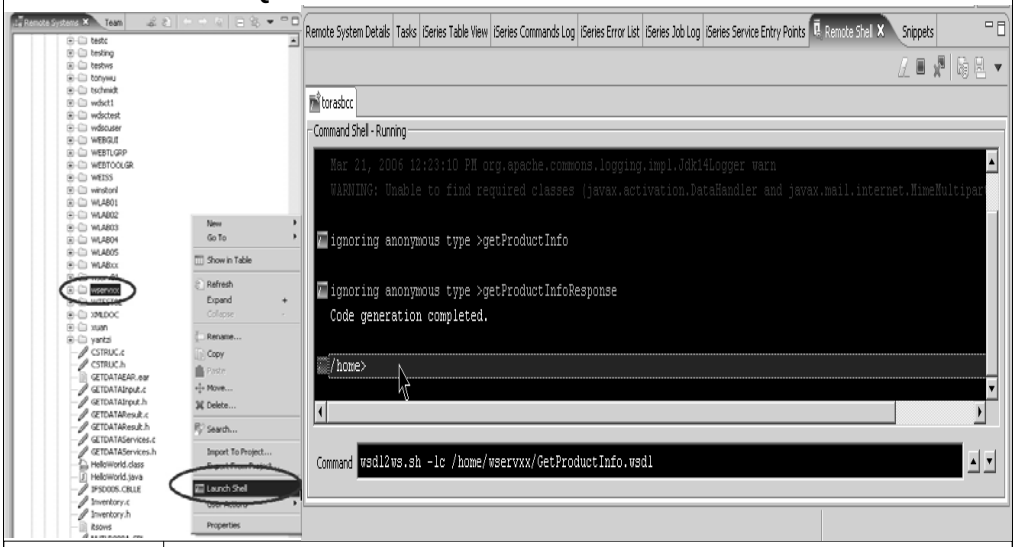
Set up RSE properties

➔Add xmltoolkit directory to path environment variable



Run the WSDL2 tool in QSHELL

Launch QSHELL from RSE and run the tool



Write RPG prototypes

Function get_Inventory_stub
to get stub for this WebService
specify server name if different from location in WSDL
Function destroy_Inventory_stub
to de-allocate stub

```
D* Prototype to get stub handle (always needed for AXIS client)
DgetStub          PR      *   EXTPROC('get_Inventory_stub')
D pEndPoint               *   Value  Options(*String)

D* Prototype to get rid of stub
DdestroyStub      PR          EXTPROC('destroy_Inventory_stub')
D pStockWS                *   Value
```

Write RPG prototype to invoke WebService

Function getProductInfo
- **first parameter to pass stub**
- **second parameter string containg product id**

Returns a pointer to a string

```
DgetProductInfo   PR                  *   EXTPROC('getProductInfo')
D pStockWS                            *   Value
D prodid                              *   Value Options(*String)
```

Write RPG logic to invoke WebService

Function getProductInfo
- **first parameter to pass stub**
- **second parameter string containg product id**

Returns a pointer to a string

```
 D* Pointer to point to string xmlstr which will contain XML doc
 D xmldoc          S                 *
 D xmlstr          S            500A   based(xmldoc)


stockWS = getStub('http://torasbcc:10032/IntelWS/services/Inventory');
  Monitor;

   //  invoke WebService
         // pass stub  and prodid
        // returns pointer to string

   xmldoc = getProductInfo(stockWS : %trim(prodId));

   xml-into product %xml( xmlstr );
```

Get Data from XML document into data structure

String returned from WebService:
<product>
<productid>PRS001</productid><name>Intel Pentium 4.0 Processor</name><price>340.49</price><stock>200</stock>
</product>
Datastructure product
Subfields Productid, name, price, stock

```
D xmlstr          S             500A   based(xmldoc)

Dproduct          DS                   inz
D productid                            like(prodid)
D name
D price
D stock


      xml-into product %xml( xmlstr );
```

Creating the program

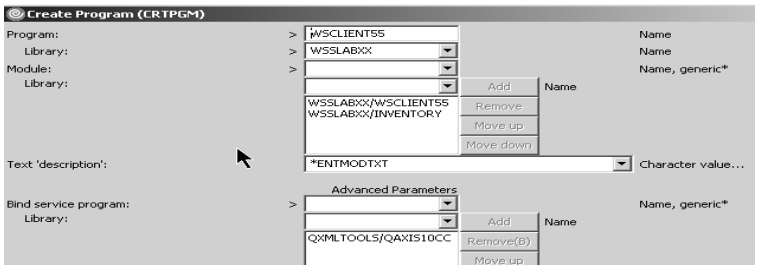
- Create module from RPG source
- Create module from generated C program
  - In RSE IFS right click on generated file select Compile > CRTMOD

    Add include directory:

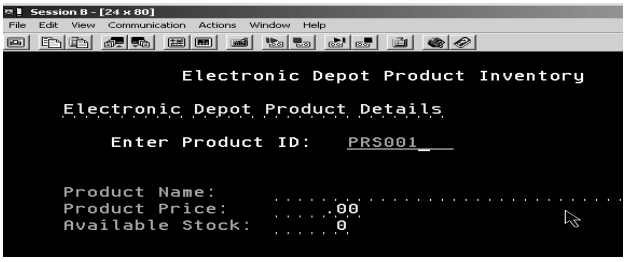
    /qibm/proddata/xmltoolkit/wscc-1.0-os400/include
- Create Program
  - In RSE right click on Module from RPG program select Create > Program….

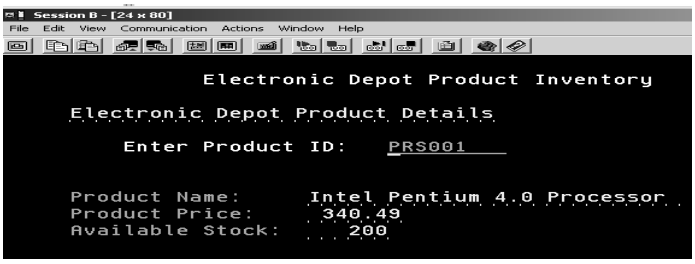
    Add C module

    Add Service program QXMLTOOLS/QAXIS10CC



Running the application





What's Next on the Horizon…

- Workflow
  - Business Process Execution Language (BPEL)
  - Now that you have created web services for major business processes you can tie them together into workflows
  - Okay, maybe next week :)

Summary / Questions

- **Roadmap: Web Services for Better Architecture**
- **Service Oriented Architecture**
- **What are Web Services**
- **Why use Web Services**
- **Web Services Technologies**
- **Web Services Tooling in WDSc**
- **Calling WebService from RPG**
- **Questions?**

Resources and References

- AXIS homepage
  http://ws.apache.org/axis
- IBM developerWorks – SOA and Web services
  http://www.ibm.com/developerworks/webservices/
- Web Services Interoperability Organization
  http://www.ws-i.org/
- WSCC Documentation
  PDF file is
  /qibm/proddata/xmltoolkit/wscc-1.0-OS400/docs/WSCC-1.0.pdf
  API documentation is
  /qibm/proddata/xmltoolkit/wscc-1.0-OS400/docs/api/index.html

Disclaimer

Acknowledgement:

- This presentation is a collaborative effort of the IBM Toronto AS/400 Application Development presentation team, including work done by:
  - *Don Yantzi, Eric D. Peters, George Farr, Claus Weiss, Al Grega, Nadir Amra*

Disclaimer:

- The information contained in this document has not been submitted to any formal IBM test and is distributed on an as is basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customers' ability to evaluate and integrate them into the customers' operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will result elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

Reproduction:

- The base presentation is the property of IBM Corporation. Permission must be obtained PRIOR to making copies of this material for any reason.

Appendix with more details

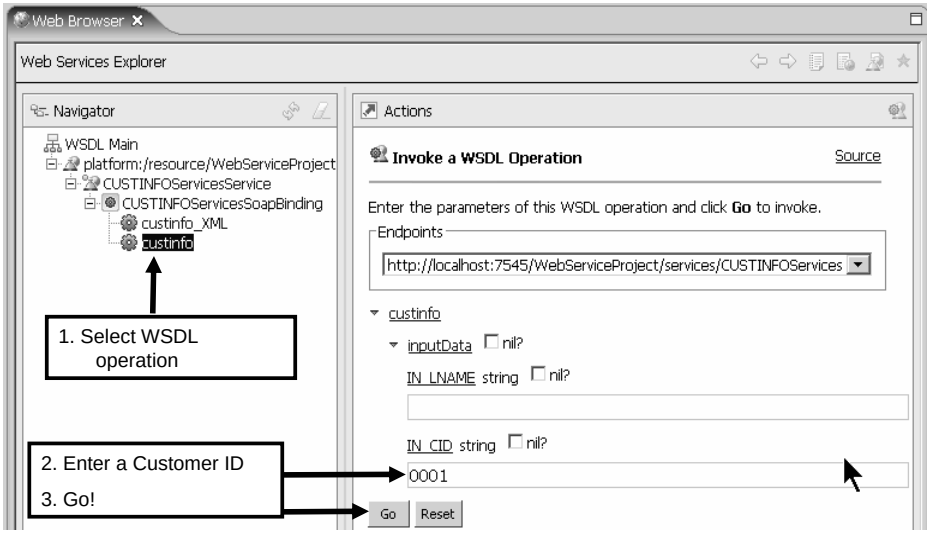
- WebServices testing
- Monitor SOAP traffic
- Getting WSDL

Testing the Web Service- Web Services Explorer

- Do not have to do everything the first time through the Web service wizard (generate the client, test the Web service etc.)
- After WSDL is generated you can (among other cool stuff):
  - Generate a Java Proxy Client (we did this all in 1 step)
  - Generate WSIL
  - Test the Web Service implemented from the WSDL
    - This opens the Web Services Explorer for testing

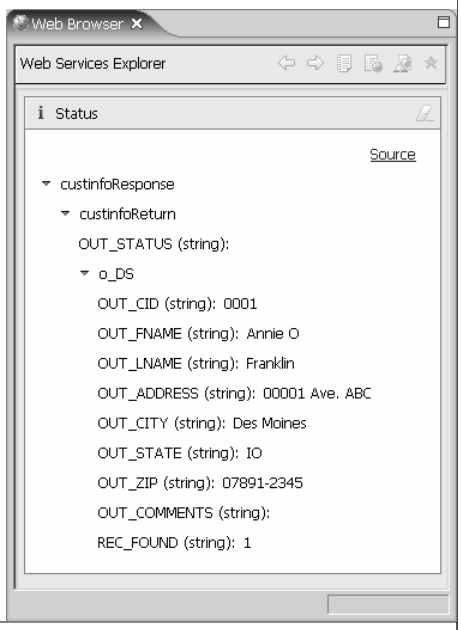
- If you are targeting WebSphere Application server, all the Web services (WSDL) and Web Services clients are accessible from the Web Services folder

Testing the Web Service- Web Services Explorer

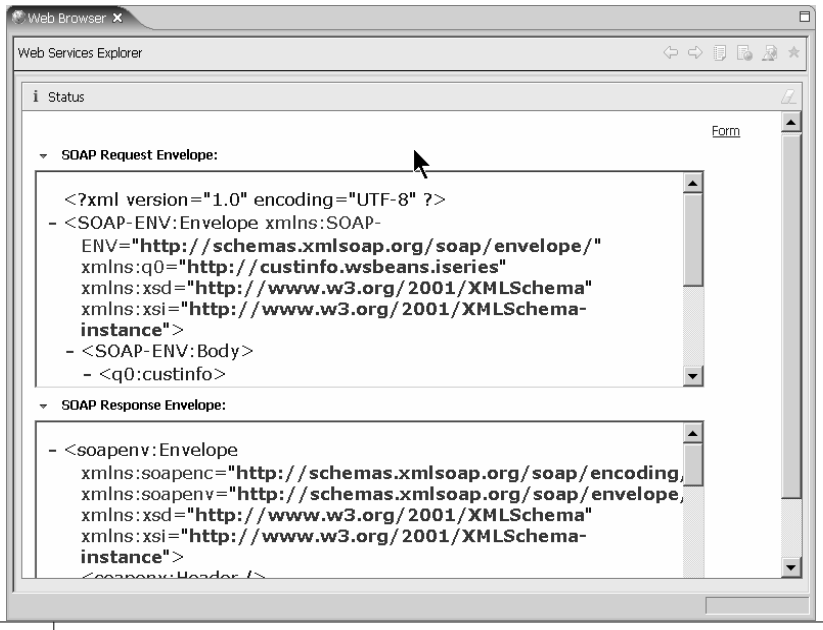


Testing the Web Service- Web Services Explorer

4. Results are shown in Status Pane
5. Click on **Source** to view the SOAP request and response messages (next slide)



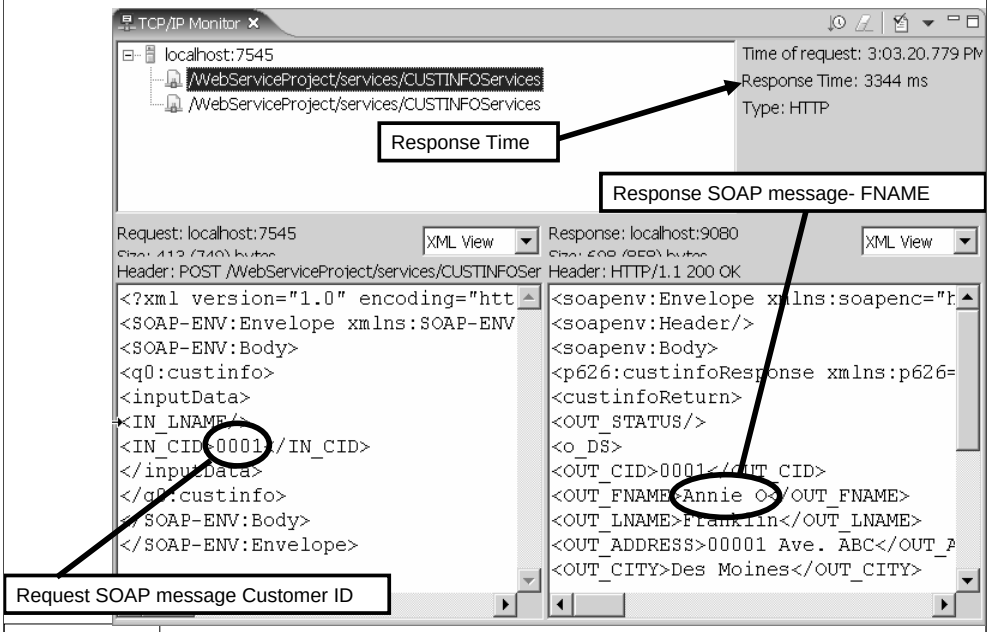
Testing the Web Service- Web Services Explorer



Testing the Web Service- Monitoring SOAP traffic

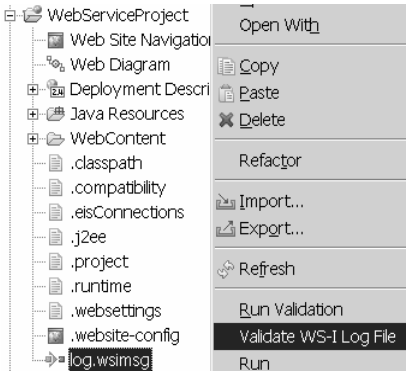
- Once you have created the Web service and Web service client, can monitor SOAP traffic using the TCP/IP Monitor
  - We selected to have the Web service wizard set up the TCP/IP Monitor (otherwise some manual steps)
  - The Monitor can ensure that traffic is WS-I compliant
  - You get a warning in the Web service wizard if the options you choose are not WS-I compliant

Testing the Web Service- Monitoring SOAP traffic



Testing the Web Service- Monitoring SOAP traffic

- In the TCP/IP port monitor, choose to generate a SOAP traffic Log file
- Invoke the Web service then validate the Log File for WS-I compliance
- The Log file is analyzed and a message indicates whether traffic is WS-I compliant



Web Services Client – Getting a WSDL File

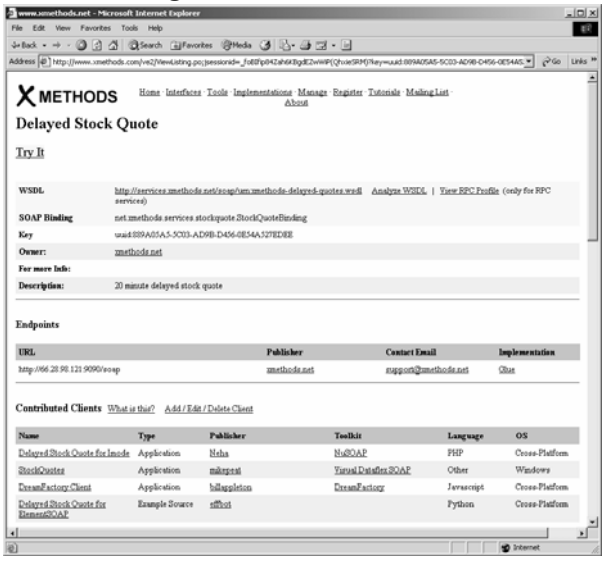
First you need to have a Web service you want to invoke.

Check out:
www.xmethods.com

Provides some Web services you can use for testing.

*All you need is a WSDL file, this provides all the information the wizard needs.

You can have the actual file or a URL to the file (and the wizard will automatically downlaod the file for you.)



WebService Client for C++ (WSCC) Components

- Option 12 of 5733XT1 (XML Toolkit for iSeries)
- Product install directory is /QIBM/ProdData/xmltoolkit/wscc-1.0-OS400
  - **bin/ contains wsdl2ws.sh tool to generate stubs (calls wsdl2ws.jar)**
  - **docs/ contain PDF document and API docs**
  - etc/ contains empty configuration file axiscpp.conf
  - include/ contains header files
  - lib/ contains service program symbolic links
  - prereqs/ contains jar files needed by wsdl2ws tool
  - samples/ contains sample code
  - WSDL2Ws/ contains wsdl2ws.jar that generates stubs
- Product install library is QXMLTOOLS
  - QAXIS10C.SRVPGM => SOAP Engine
  - QAXIS10HC.SRVPGM => HTTP Channel
  - QAXIS10HCS.SRVPGM => HTTP Channel SSL
  - QAXIS10HT.SRVPGM => HTTP Transport
  - QAXIS10X.SRVPGM => XML Parser

WSCC Components (cont.)…

- Prerequisites
  - XML Parser (XML Toolkit for iSeries, licensed program product ID 5733XT1, option 9)
  - 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

Note on Web Services Security

- **XML Digital Signature - Integrity**
  - Digitally signs elements in SOAP message to provide integrity
  - Receiver can be sure:
    - You sent the message
    - The message has not been changed in transit
  - Elements in SOAP message are not encrypted
- **XML Encryption - Confidentiality**
  - Encrypts elements of the SOAP message to provide confidentiality
- **Certificates – Authentication**
  - New in Web Services security

The World of Acronyms

- FTP - File Transfer Protocol
- HTTP - Hyper Text Transfer Protocol
- J2EE - Java 2 Enterprise Edition
- JSR - Java Specification Request
- SMTP - Simple Mail Transfer Protocol
- SOAP - Simple Object Access Protocol
- UDDI - Universal Description, Discovery, and Integration
- WSDL - Web Services Description Language
- WSFL - Web Services Flow Language
- WSIL - Web Services Inspection Language
- XML - eXtensible Markup Language

References

- IBM DeveloperWorks
  www.ibm.com/developer/webservices
- IBM WebSphere Studio Zone
  www.ibm.com/developer/websphere/zones/studio/
- W3C (standards)
  http://www.w3c.org/2002/ws
- Web Services Interoperability Organization (WS-I)
  www.ws-i.org
- Java Community Process
  www.jcp.org

Trademarks and Disclaimers

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

AS/400	**e-business on demand**	**i5/OS**
AS/400e	**IBM**	**OS/400**
eServer	**IBM (logo)**	**System i5**
@server	**iSeries**	

Rational is a trademark of International Business Machines Corporation and Rational Software Corporation in the United States, other countries, or both.
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.
Linux is a trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Other company, product or service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Some information addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

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

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.