



IBM Software Group

IBM® Rational® Web Developer V6
IBM® Rational® Application Developer V6
Web Service Tools



@business on demand.

© 2004 IBM Corporation
Updated January 25, 2005

Goals

- Understand Web Services development tools in IBM Rational Web and Application Developer V6

Agenda

- Overview and new functions
- Creating Web Services Provider and Clients
- Additional Web Services Development Capabilities
- Testing Web Services
- Validators and Preferences
- UDDI Registry support

Section

Web Services Development Tools - Overview

Web Services: Overview

- IBM Rational Web and Application Developer V6 provide a complete Web Services development environment

Web Services Provider

- Web Services Wizard
- WS-I Validators
- Configure Handlers 
- Web Services Explorer
- SOAP Monitor 
- Sample JSPs
- Web Services Explorer
- UDDI Test registry

Create



Test



Publish

Consume



Test



Discover

Internet

Web Services Client

- Web Services wizards
- Deployment editors
- Configure Handlers 
- Web Services Explorer
- SOAP Monitor 
- Sample Java™ Server Pages (JSPs)
- Web Services Explorer
- Discovery wizard (Faces JSP)
- Web Service Wizard



• IBM Rational Web and Application Developer V6.0 provide a complete suite of development tools focused on Web Services. There are wizards, editors, and other tools for creating Web Services and creating clients which call Web Services.

• Web Services can be generated for JavaBeans and Enterprise JavaBeans (EJB) following the latest industry specifications. Web Services can also be generated for DB2® databases. For interoperability, Web Services, WS-Interoperability (WS-I) can be enforced and verified. New to V6 is support for WS Handlers. WS Handlers can be created on the tab of the module (whether it is a Web module, EJB module, etc.) Deployment Descriptor or from the Project Explorer view. WS Handlers allow for non-business processing to occur on requests and responses to and from Web Services. Security and performance enhancements are two areas of processing which are best handled within a Handler. The SAAJ 1.1 APIs are also best used within handlers as they work with the different objects which are sent and received.

• Once Web Services are created, they can be tested with one of the test facilities included with IBM Rational Web Developer or IBM Rational Application Developer V6.0. Sample JSP™ files for the web developer wanting to run this code on an application server are available, or the Web Services Explorer may be used to quickly test the Web Services using a dynamic client. In addition to these tests, you can use the SOAP monitor to view SOAP traffic and validate SOAP messages to be WS-I compliant.

• After a Web Service is completely tested, it may be published to a UDDI (Universal Description, Discovery, and Integration) Registry. Universal Description, Discovery, and Integration (UDDI) is a specification that defines a way to publish and discover information about Web services. UDDI registries come in two forms: public and private. Both types comply with the same specifications. A private registry enables you to publish and test your internal e-business applications in a secure, private environment. IBM Rational Web Developer and IBM Rational Developer V6.0 both include a private UDDI registry. The private UDDI test registry is fully UDDI v3 compliant. Using the Web Services Explorer, a Web Service may be published to either a private or public UDDI registry. The Web Services Explorer is fully UDDI V2 compliant and can interact with a UDDI V3 registry over

WADP\Dev\WebServiceTool.ppt

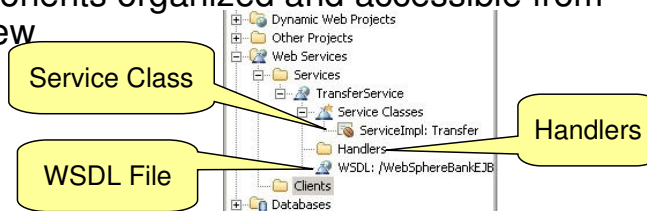
Page 5 of 34

• Clients can also be created for Web Services. A developer can use the Web Service Wizard or the Web Service Discovery Dialog to either discover a Web service (WSDL file) that exists online or in your workspace. For the Web Service Discovery Dialog, right click in Page Data view of a Faces JSP data and select New > Web Service. The New > Web

New Features Overview



- J2EE 1.4 Web Services and WebSphere Application Server V6 support
 - ▶ Web Service Client Deployment Descriptor editor merged into the module Deployment Descriptor editor, as per J2EE specification
 - ▶ V5 and V6 command line tools available
- Easy Handler creation and registration support
- New WS-Interoperability (WS-I) options
- Web Service components organized and accessible from Project Explorer view
 - ▶ Service Classes
 - ▶ WSDL files
 - ▶ Handlers



The biggest updates for Web Services Development Tools is J2EE 1.4 Support and WebSphere Application Server V6 support

Web Service projects (Services and Clients) are now viewable in the Project Explorer. There is a Web Services folder added to the first line in the Project Explorer view, where Web Services and Web Service clients are listed. For both the Web Service and Web Service Client, there is quick access to the service classes, handler information, externally discovered services (WSIL), and the WSDL itself. For J2EE 1.4 and J2EE 1.3 Web Services only, the Project Explorer looks at the different deployment descriptors to check the J2EE level and searches WSIL to correctly display the Web Services existing throughout the workspace. In V5, you had to search through folders to find all of your Web Service files.

The Web Service Client Deployment Descriptor editor merged into the module Deployment Descriptor editor, as per J2EE specification.

New Features Overview (cont.)






- Automatic creation, startup and injection of TCP/IP Monitor between clients and services using the new SOAP Monitor
- Improved WSDL editor with new rename, drag and drop, and WS-I creation support with skeleton objects
- Fully UDDI V3 compliant private UDDI test registry included
- Additional preferences and default settings simplify Web Service Wizards
- Revamped Sample Gallery and Tutorial Gallery with refreshed Web service samples and tutorial



The list of improvements is quite lengthy. Here are a few more of the major enhancements which have been included.

Supported Web Services Specifications





- JAX-RPC 1.1
 - ▶ Updated mappings (any unsupported type is mapped to javax.xml.soap.SOAPElement) 
- JSR 921 (JSR 109)
 - ▶ Client deployment model is now defined in J2EE modules (web.xml, ejb-jar.xml, and application-client.xml) 
- SAAJ 1.2
 - ▶ Core SAAJ classes, such as SOAPElement, now extend the DOM package (org.w3c.dom) 
- WSDL validation
 - ▶ WSDL 1.1 Specification
 - ▶ XML Schema 1.0 Specification



As for specs that have improved, here is a quick list of what has been introduced. Hinged on J2EE 1.4, there is JAX-RPC 1.1 with updated mappings, JSR-921 (also known as JSR-109 1.1) has a noticeable change of dropping the webservicescient.xml in lieu of adding all JSR-921 information in the web.xml.


SAAJ 1.2, or SOAP Attachments API for Java, is the new technology that solves the problem that SOAP could not communicate binary objects like pictures, audio files, etc. SAAJ 1.2 puts this information into SOAP headers like SAAJ 1.1 did. Core SAAJ classes such as SOAPElement now extend the org.w3c.dom package. Customers do not have to serialize/deserialize to use SOAPElements any more.

Supported Web Services Specifications (cont.)

- **WS-I SSBP (Simple SOAP Binding Profile)** 
 - ▶ Derived from WS-I Basic Profile 1.0 requirements related to the serialization of the envelope and its representation in the message
- **WS-I AP (Attachment Profile)** 
 - ▶ Support for SOAP with Attachments and MIME bindings
- **WS-Security 1.0** 
 - ▶ WS-Security 1.0 + X.509 token Errata 3 + WS-I BSP draft (Basic Security Profile)
- **UDDI V3 and V2** 
 - ▶ Multi-registry topologies, increased security features, improved WSDL support, a new subscription API, and core information model advances

Web Service Server and Runtime Support

- Wizards help manage server and runtime combinations

Server	Runtimes	Comments
 WebSphere Application Server V6	IBM WebSphere (upgraded for J2EE 1.4)	Supports J2EE 1.4 Web Services
WebSphere Application Server v5.1	IBM WebSphere, IBM SOAP	Supports JSR 101/109 Web Services
WebSphere Application Server v5.0.x	IBM WebSphere, IBM (Apache) SOAP	<ul style="list-style-type: none"> Supports JSR 101/109 Web Services IBM SOAP supports the old Apache SOAP Web Services
Tomcat v3.2	IBM (Apache) SOAP	IBM SOAP supports the old Apache SOAP Web Services
Tomcat v4.0, v4.1, v5	Apache Axis 1.0, IBM (Apache) SOAP	Apache Axis 1.0 supports JAX-RPC spec.

Section

Creating Web Services Providers and Clients

Creating Web Services

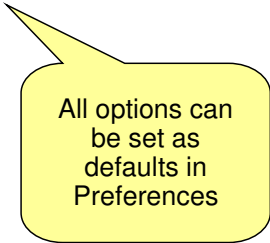
- Web Services may be created in multiple ways
- Top-down
 - ▶ Design or import a WSDL file and generate a skeleton JavaBean or skeleton Stateless Session EJB
 - ▶ Developer must add implementation
- Bottom-up
 - ▶ Design or import a JavaBean class, Stateless Session EJB (only available in IBM Rational Application Developer), or DADX document (for DB2 Document Access Definition Extension that defined a Web Service) and generate a Web Service



DB2 XML Extender uses an XML document format called Document Access Definition (DAD) to define the mapping between XML and relational data. The Document Access Definition Extension (DADX) file specifies a Web service. It does this by using a set of operations that are defined by SQL statements, by a list of parameters, and by DAD file references. Operations are similar to programming methods that you can invoke. You can use XML collection operations to generate and store XML documents. You can use SQL operations to query and update the database and call stored procedures.

Options for creating Web Services

- Publish Web Services (WSDL) to UDDI registry
- Create client application
- Enable monitor capabilities
- Test Web Service
- Specify the Web Service runtime
 - ▶ IBM WebSphere (recommended – J2EE)
 - ▶ IBM SOAP (based on Apache SOAP 2.3)
- Select Server
 - ▶ WebSphere Application Server V6.0 (required for J2EE 1.4 Web Services)
 - ▶ Other WebSphere Application Servers
 - ▶ V6 server must be started or it will be started if local
- Select message Style and Use
 - ▶ Document Literal (recommended)
 - ▶ RPC/Literal
 - ▶ RPC/Encoded
- Enable Security
 - ▶ XML encryption or XML Signatures or both



All options can be set as defaults in Preferences

Web Service Provider for Stateless Session EJB

- Stateless Session Beans have multiple Web Service deployment options
- SOAP/HTTP
 - ▶ Creates router module (Web Project) and artifacts to receive the SOAP/HTTP request and call the EJB
 - ▶ Follows JAX-RPC, JSR 921 (JSR 109) specification
- SOAP/JMS
 - ▶ Creates router module (EJB Project) with Message Driven Bean to receive the SOAP/JMS request and call the EJB
 - ▶ Follows JAX-RPC, JSR 921 (JSR 109 specification) with enhancement of JMS protocol as the actual transport



Web Service Client support

- Create Client proxy from WSDL file
- Different client options
 - ▶ J2EE Web project
 - ▶ J2EE EJB project (IBM Rational Application Developer only)
 - ▶ J2EE Application client project
 - ▶ Stand-alone Java project (JAX-RPC only)
- Web Services runtime, Server, and XML Security options available
- Generated client proxy can be used directly or added to client application



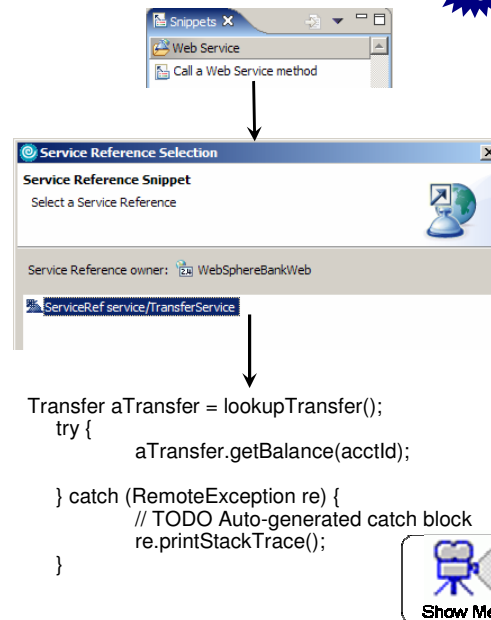
Section

Additional Web Services Development Capabilities

Web Service Snippet


New
v6

- Developers can easily call Web Services with Web Services Snippet
- Defined Service Reference or new Service Reference can be specified
- Operation on service can be selected and parameters specified
- Build path needs to be updated after adding snippet



The screenshot illustrates the workflow of the Web Service Snippet tool. It starts with a 'Snippets' window containing a 'Web Service' snippet. An arrow points to a 'Service Reference Selection' dialog box where a service reference is selected. Another arrow points to the resulting Java code snippet.

```
Transfer aTransfer = lookupTransfer();
try {
    aTransfer.getBalance(acctId);
} catch (RemoteException re) {
    // TODO Auto-generated catch block
    re.printStackTrace();
}
```

 Show Me

Deployment Descriptor Editors

- J2EE 1.4 deployment descriptors
 - ▶ Provider configuration
 - webservices.xml
 - ▶ Client configuration
 - web.xml, ejb-jar.xml, application-client.xml replace webservicessclient.xml
- Editors used to customize deployment characteristics
 - ▶ Configuring handlers
 - ▶ Configuring WS-Security
- WebSphere specific information stored separately
 - ▶ ibm-webservices-bnd.xmi and ibm-webservices-ext.xmi for provider binding and extension information
 - ▶ ibm-webservicesclient-bnd.xmi and ibm-webservicesclient-ext.xmi for client information



IBM Rational Web Developer and IBM Rational Application Developer still support J2EE 1.3 Web Services and the webservicessclient.xml editor used to configure Web Service clients.

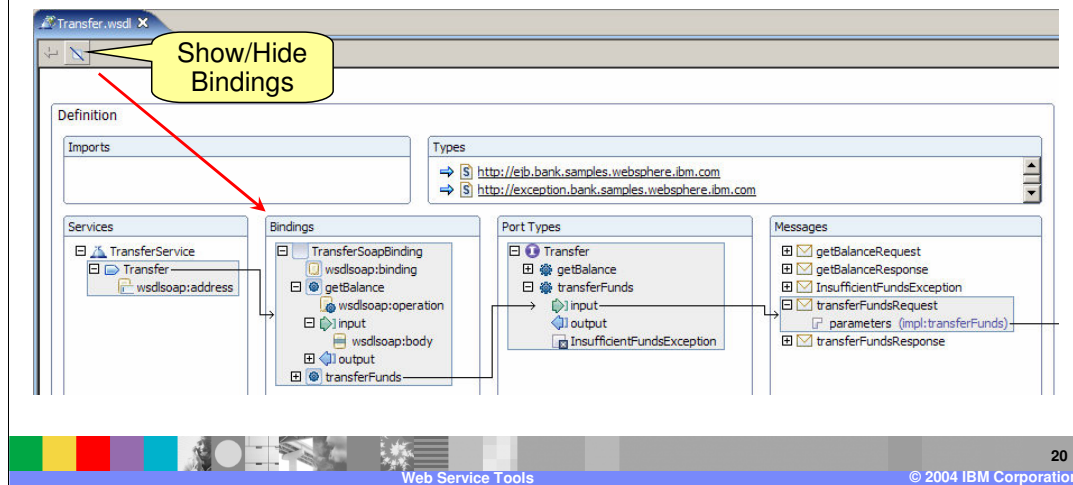
Authoring WSDL files



- New WSDL files can be created or existing WSDL files may be updated with WSDL editor
- New option to create skeleton WSDL file with a Port Type, Operation, and Message generated with default names
 - ▶ Reduces time to create common elements
- Renaming elements propagates changes to other generated elements
 - ▶ Rename of operation will also rename generated request and response messages
- Different SOAP and HTTP binding options available

WSDL Editor

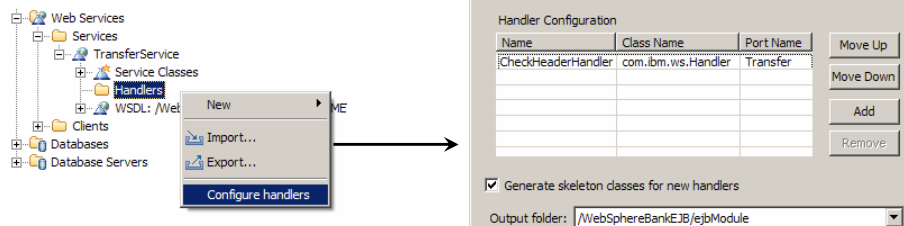
- Graph view provides visual development for WSDL
- Source view also available for working with WSDL directly
 - ▶ Code assist available in Source view



Adding and Configuration of JAX-RPC Handlers

- Handlers can be created to intercept requests or responses for a provider or client
 - ▶ Handler classes implement `handleRequest()`, `handleResponse`, and `handleFault()` methods
- Existing handlers may be registered with a port
- Skeletal handlers can be created and associated with a port
 - ▶ Developer must add logic to skeleton handlers
- Deployment descriptor updated
 - ▶ WS Handler tab in editor

**New
V6**



Web Service Tools

© 2004 IBM Corporation

21

Handlers are like interceptors at the request and response side for the provider and the client.

Handlers implement `handleRequest()`, `handleResponse`, and `handleFault()` methods as defined by JAX-RPC Handler interface.

The Handler configuration wizard allows you to add handlers to specific WSDL service and ports (operations). If handler does not exist, it will create skeleton Handler Java code. However, you need to provide the handler logic.

WS-Security Support in Tools



- WS-Security supported by Web Services wizards and Web Service provider and client deployment descriptors
 - ▶ Secure clients through EJB, Web, and Application Client Deployment descriptor editor
 - Settings stored in ibm-webservicesclient-bnd.xmi and ibm-webservicesclient-ext.xmi
 - ▶ Secure providers through webservices.xml editor
 - Settings stored in ibm-webservices-bnd.xmi and ibm-webservices-ext.xmi
- Authentication – username/password, certificate, LTPA tokens
- XML Digital Signature - verifies the integrity of SOAP message content
- XML Encryption - encrypts the content of a SOAP message



Transport as well as message level security can be enabled within IBM Rational Web Developer and IBM Rational Application Developer V6.

Full support is still provided in IBM Rational Web Developer and IBM Rational Application Developer V6 for specify WS-Security for J2EE 1.3 Web Services.

Section

Testing Web Services

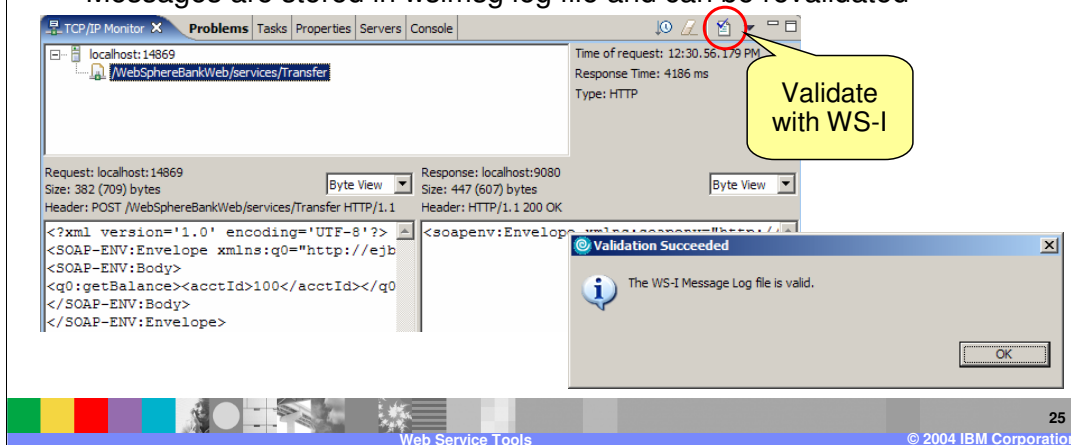
Unit Test of Web Services Providers and Clients

- Web Services can be tested in different ways
- Web Services Explorer
 - ▶ Most common and easiest way to test Web Services
 - ▶ Interface for testing Web Services provider generated dynamically from WSDL
 - ▶ WS-Security currently not supported
- Web Services Sample JavaServer Pages
 - ▶ Set of JSPs which are generated when the appropriate option is selected during creation of Web Services provider or client
 - ▶ Provides a Web interface to invoke a proxy call to a Web Service and provides a starting point for custom clients
- Universal Test Client
 - ▶ Invokes Web Service provider through proxy class
 - ▶ WS-Security currently not supported
- Component Testing for Web Services



SOAP Message Monitor

- Captures HTTP Requests and Responses including SOAP messages
- Enabled when Web Service is created with Monitor Web Service option selected or separately on test server
- Requests and Responses can be validated for WS-I compliance
- Messages are stored in wsimg log file and can be revalidated



TCP Monitor view adds WS-I functionality, so now you can validate WS-I message log files. The log.wsimg that is produced can be saved in project location of your choice, and if the project has multiple WSDL files, the wizard gives you an option to select which WSDL file you want.

Send the Web service traffic through the TCP/IP Monitor, which allows you to watch the SOAP traffic generated by the Web service and to test this traffic for WS-I compliance. To validate the traffic for WS-I compliance, click the Validate WS-I Message Log File button in the TCP/IP Monitor toolbar. This will generate a log file that lists elements of your Web service that are not WS-I compliant.

So why is WS-I important? It is important to keep a web service standard non-proprietary in order to make all web services interoperable. In order to keep one company from setting the standard, the WS-I group is made up of people from many companies that work towards of making web services communicate over the same standards.

Testing in a Production Environment

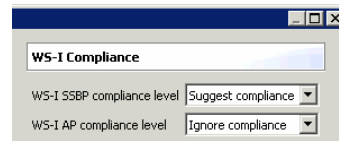
- Web Services developed in IBM Rational Web and Application Developer use “localhost” in the service endpoint information in WSDL if generated for local server
 - ▶ For clients calling from another system, service end point information must be updated
- Client applications can be developed to set the end point – pass as parameter or read from properties file
 - ▶ Generated Client proxy has setEndPoint() to allow change of endpoint
- The WSDL file from a deployed Web Service can be exported from the production Application Server with the correct host information and clients may be regenerated

Section

Validators and Preferences

WS-I Compliance Levels

- Web Services may be checked for WS-I compliance for improved interoperability with other Web Service implementations
- Level of compliance checking is configurable
 - ▶ Per workspace – set in workspace preferences
 - ▶ Per project – set in project properties
- Require
 - ▶ WSDL validator reports errors for non-compliances
 - ▶ Tools block choices that will yield a non-compliant service
- Suggest
 - ▶ WSDL validator reports warnings for non-compliances
 - ▶ Tools warn about choices that will yield a non-compliant service
- Ignore
 - ▶ WSDL validator will not detect WS-I non-compliances
 - ▶ Tools will quietly allow choices yielding a non-compliant service



WS-I SSBP - Simple Soap Binding Profile

WS-I AP - Attachment Profile

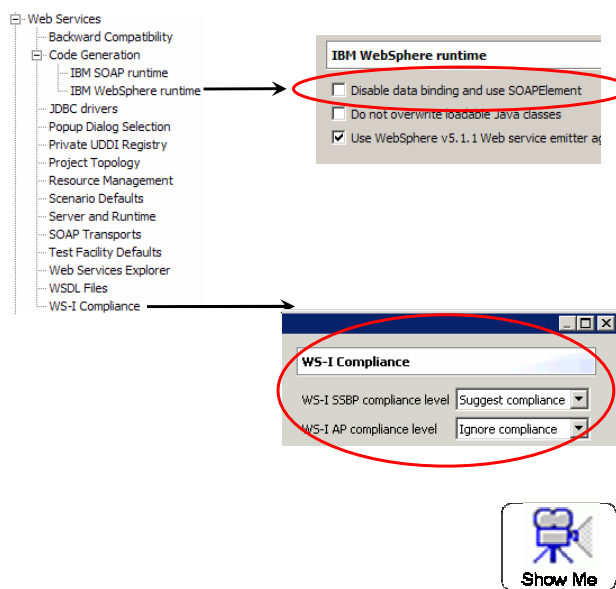
Simple Soap Basic Profile and Attachment Profile validators (WS-I) have been added to IBM Rational Application Developer V6.0. The level of compliance checking is configurable per workspace or per project. As for what each level means, Require means WSDL validator reports errors for non-compliances and our tools block choices that will yield a non-compliant web service. Suggest means that the WSDL validator reports warnings for non-compliances and our tools warn about choices that will yield a non-compliant service. Ignore means that the WSDL validator will not detect WS-I non-compliances and our tools will quietly allow choices yielding a non-compliant service.

The Web services WS-I validation tools support the level of WS-I compliance outlined in the WS-I Basic Profile 1.0, the WS-I Simple SOAP Binding Profile 1.0 (WS-I SSBP), and the WS-I Attachments Profile 1.0 (WS-I AP). You can choose to make your Web service compliant or non-compliant, depending on your needs. For example, encoded style (RPC/encoded), SOAP over JMS protocols, and secured Web services, do not comply with the WS-I Basic Profile.

NOTE: The support for WS-I AP compliance verification is not complete as the WS-I AP has not finalized. This value should be set to ignore at this time. Once WS-I AP is finalized, support will be available for validating Web Services with this as well.

Web Services: Preferences

- Most set defaults for Web Service creation or consumption
 - ▶ Web Services Runtime
 - ▶ Client type
 - ▶ SOAP transport
- Notable Preferences
 - ▶ Use SOAPElement
 - ▶ WS-I Compliance additions
 - ▶ Additional defaults values can be set



There are a couple new Web Service Preferences in IBM Rational Application Developer V6.0 and some that have been split up from V5.x. Most of these cover setting of defaults so you can create web services and web service clients quicker. Under Code Generation, the ability to disable data binding was there. However, in V6 the tools disable data bindings and uses SOAPElements instead.

WS-I Compliance has expanded from just WS-I BP (Basic Profile) to cover WS-I SSBP (Simple SOAP Binding Profile) and WS-I AP (Attachment Profile). Other preferences remain unchanged. Their sole purpose is to set defaults for newly created or consumed Web services.

Server and Runtime preference is on its own, allows you to set server and runtime level for all web service creation/consumption.

Client Type under the Project Topology option allows you to set the default type of web service (Web, Java, EJB, Application Client).

SOAP Transports split off to its own Preference (HTTP or JMS)

Test Facility Defaults allows you to set what default the wizard chooses when creating a test client or testing a web service.

Section

Publish and Discovery from UDDI Registry

UDDI Registry Support

- UDDI Test Registry included with IBM Rational Web and Application Developer
 - ▶ Full functioning UDDI registry for publishing services and retrieving service information
 - ▶ Custom categories can be added for publishing or searching
 - ▶ Same API as the real UDDI registry
 - ▶ Compliant with UDDI V3
- Web Services Explorer available for publishing and retrieving Web Services (WSDL file) from public and private UDDI registries
 - ▶ Publish WSDL files from a workspace
 - ▶ Import WSDL files into a workspace
 - ▶ Generate a client from a WSDL file
 - ▶ Test a published services
 - ▶ Compliant with UDDI V2 and works with UDDI V3 using V2 format

Section

Summary and Reference

Summary

- IBM Rational Web and Application Developer provide a rich Web Services development environment
- Full set of features included for developing, testing, and publishing Web Services

Trademarks, Copyrights, and Disclaimers

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

IBM	CICS	IMS	MQSeries	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
eIogo business	DB2	Series	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel 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.

Linux is a registered trademark of Linus Torvalds.

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

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. 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.

© Copyright International Business Machines Corporation 2004. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.

