Chapter 7. July 2007

Welcome to the July 2007 edition of IBM Information Server Developer’s Notebook. This month we answer the question;

> What are Web services, and how and why would I use them with DataStage and/or QualityStage ?
>
> *(Which when expanded, also becomes; What are loose coupling, SOA, SCM, UDDI, SOAP, model-view-controller, and named frameworks.)*

In short, we are going discuss all of the relevant terms and technologies above, and provide examples that detail how to deliver this functionality using IBM Information Server.

Software Versions

All of these solutions were *developed and tested* on IBM Information Server version 8.01, on the Microsoft Windows XP/SP2 platform. IBM Information Server allows for a single, consistent, and accurate view of data across the full width of the corporate enterprise, be it relational or non-relational, staged or live data. As a reminder, the IBM Information Server product contains the following major components;

> WebSphere Business Glossary™, WebSphere Information Analyzer™, WebSphere Information Services Director™, WebSphere DataStage™, WebSphere QualityStage™, WebSphere Metadata Server and Metabridges™, WebSphere Metadata Workbench™, WebSphere Federated Server™, Classic Federation™, Event Publisher™, and Replication Server™, and Rational Data Architect™.

Obviously, IBM Information Server is a large and capable product, addressing many strategic needs across the enterprise, and supporting different roles and responsibilities. This month we give focus to DataStage/QualityStage Enterprise Edition, Web Information Services Director (WISD), and Rational Data Architect (RDA) components of IBM Information Server.

7.1 Terms, terms, terms

The first programming language this month’s author ever learned allowed one to create database entry screen forms and printed reports. Every time you created a new database entry screen form program, you used an array of primary keys to maintain a current list of records that the end user operator was browsing. The

© Copyright IBM Corp. 2008 All Rights Reserved. US Government Users Restricted Rights-Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

program would perform simple math on a pointer into the array, allowing the user to view the next, previous, first and last records. Every time the user moved between records, the program would go back to the database and fetch a fresh copy of the full width data record, while the array only stored primary key information. Every language I have moved to since that time, I have followed the same design; an array of primary keys, a pointer into the array and to the current record, and fetching on every move to a new record to receive a fresh copy of the full data record. This has become my design pattern for constructing database entry screen forms.

Design patterns, model-view-controller

As a term, *design pattern* is defined to be a formal way of detailing a solution to a recognized problem. *Model-view-controller (MVC)* is perhaps the most basic and core design pattern used in modern software engineering. With MVC, the following is said to be true;

- A single or set of source code files or application is organized into three functional areas-
  - The *view* represents the presentation layer, the user interface. On a Web form, the view may be written in HTML, while a graphical two-tier program may write the view in Java/J2SE Swing or RCP.
  - The *model*, or data model, represents the persistence layer; the calls to and from a data store, commonly a relational database server.
  - The *controller* sits between the view and the model, and ties end user initiated program events to the persistence layer, to database calls. In effect, a given end user button click invokes a given database routine.
- The advantages towards using MVC include-
  - Given a 1000 line application, one-third may represent the view, one-third the model, and one-third the controller.

    When you upgrade the version of the database, or migrate between database platforms, only those 300 lines that are database dependent need to be (re)tested. The view and the controller are not affected.

    If you need to deploy to or support a new user interface, for example, a standard desktop Web browser based application now needs to run on mobile devices, again you need only change and test one-third of the application.
  - Engineering skill-sets can become specialized and more effective. The Web interface programmer can give focus to a productive end user experience, while the database programmer can give focus to highly scalable designs and routines.
  - And more.

© Copyright IBM Corp. 2008 All Rights Reserved. US Government Users Restricted Rights-Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

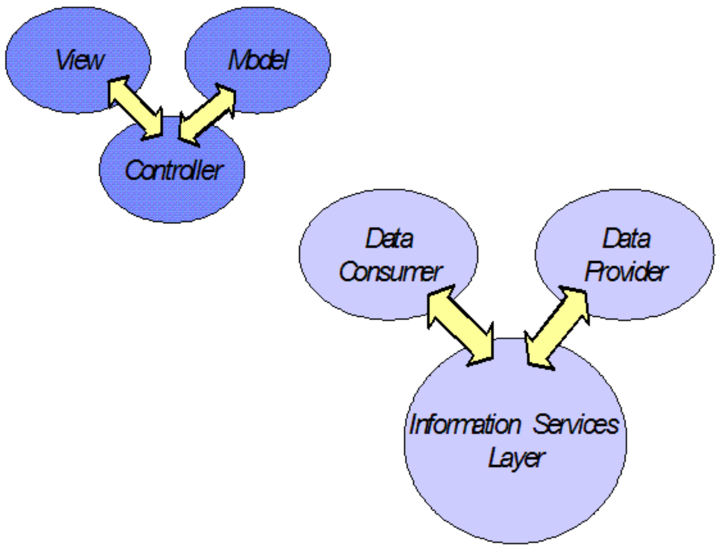


Figure 7-1 Design pattern, model-view-controller, left side of image.

While a design pattern is a logical concept, a blueprint, there are *frameworks* which are the physical constructs used to deliver design patterns; they are the houses. Frameworks can include libraried source code, object hierarchies, tooling, and more. Java/J2EE Struts, Java/J2EE Spring, Java/J2EE Faces (JFace), Java/J2SE-RCP, Ruby, Python, Microsoft [dot].Net, PHP, and dozens more, are all frameworks that support the design pattern of MVC.

The MVC design pattern is attributed to Xerox Labs, circa 1979, and Trygve Reenskaug with the Smalltalk-80 language. MVC is also famously re-introduced in the 1995 book, *Design Patterns: Elements of Reusable Object-Oriented Software*, by Gamme, Helm, Johnson, and Vlissides.

Coupling; loose-coupling, decreased coupling

It was stated above that one-third of a given application is all that would be affected if the database server version or platform were changed; the model would need to be (re)tested, while the view and controller should remain unaffected. The degree to which that statement is true is dependent on characteristics and strength of the framework being used (to what extent does the framework allow for decreased coupling), and the quality of the your implementation. (How well you wrote your code and followed MVC.)

A *tightly coupled* application would have all of its source code compiled into one binary executable; by example, function-A can only call a specific image of function-B, which is bound inside the same executable address space. In a

© Copyright IBM Corp. 2008 All Rights Reserved. US Government Users Restricted Rights-Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

loosely coupled application, function-A uses some dynamic binding or communication method to invoke function-B, which may be variably located or defined.

As a term, *loose coupling* defines the resilient relationship between two or more functional entities. Loose coupling may also use the terms of *service-requestor* and *service-provider*. Loose coupling is considered a desirable capability for applications or organizations that change frequently. Other terms and ideas related to loose-coupling include;

- An application may be loosely coupled by time and/or format or location.
  - A *loosely coupled application by time* uses a message driven middle-ware component like IBM MQ/Series. IBM MQ/Series is supported by IBM Information Server (IIS), more specifically the DataStage, QualityStage, and Federated Server components to IIS.
  - A *loosely coupled application by format* allows for-
    - Existing data element order and/or type to be changed.
    - Omission of data elements.
    - Ability to handle new data elements.
  - A *loosely coupled application by location* allows a given function to be location independent; meaning, the given function or service reference on the (network) is done by logical name, not by address.
- In general, the advantage to decreased coupling (loose coupling over tight coupling), include-
  - Lower individual program element maintenance burden; the prior example being the migration of the database and its impact to the overall application.
  - Greater ability to respond to dynamic requests; ability to add or delete hosts, relocate or replicate application assets, ability to modify a given application entity's function or definition, and related.
  - Ability to re-use older or existing corporate assets, including the ability to Webify legacy applications.
  - Others.
- In general, the basic disadvantages to decreased coupling include a small amount of system overhead to support the (dynamic run time binding of functions), and then the increased complexity of the architecture when taken as a whole.

© Copyright IBM Corp. 2008 All Rights Reserved. US Government Users Restricted Rights-Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Web services

On most levels, invocation of a Web service is no different than a remote procedure call; similar to having function-A invoke function-B, where the two functions may reside in different address spaces (they share no variables or allocated resource). Having a given function or program invoke a Web service allows for decreased coupling.

The list that follows is meant to explain some of the terms and conditions related to Web service invocation and delivery. Generally, the tooling and server software involved in the delivery of Web services hide all of this complexity from you. The following is offered;

Ê Generally, the [full] Web service network and communication protocol stack from bottom to top involves-

  – Physical layer; Ethernet physical layer, ISDN, SONET, etcetera, could be viewed as the layer at or just above the copper wire or fiber optic cable.

  – Data link layer; Ethernet, FDDI, Frame Relay, 802.11, and others.

  – Internet layer; IPV4, IPV6, IPSec, others.

  – Transport layer; TCP(/IP), TCP(/UDP), others.

    TCP/IP is a general communication and protocol stack supporting local area networks. Other networks can obviously support Web services, (Novel, others), TCP/IP is just the most common. TCP/IP can support numerous and concurrent application types to include file sharing, electronic mail servers, and others, in addition to Web services.

    By means of analogy TCP/IP is what a getting a dial tone or open line to another destination is to your telephone. An absolute destination on a TCP/IP network is the unique combination of an IP address and a port number, the combination of which forms a socket. A socket would equate to your telephone number.

  – Application layer; HTTP, DHCP, DNS, FTP, SAMBA, POP3, others.

    HTTP (Hypertext Transfer Protocol) is a request response protocol between a client (service requestor) and a server (service provider). HTTP sits atop, relies upon, TCP/IP. HTTP would be in the list of protocols or services like file sharing (SAMBA), and mail servers (POP3), listed above.

    HTTP encoded messages have a standard message header, body, and so on. HTTP encoded messages can be viewed as ASCII text.

    Continuing the telephone analogy above, speaking HTTP over TCP/IP would be like a voice or then facsimile transmission over a phone line.

  – (Further to the application layer);

© Copyright IBM Corp. 2008 All Rights Reserved. US Government Users Restricted Rights-Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

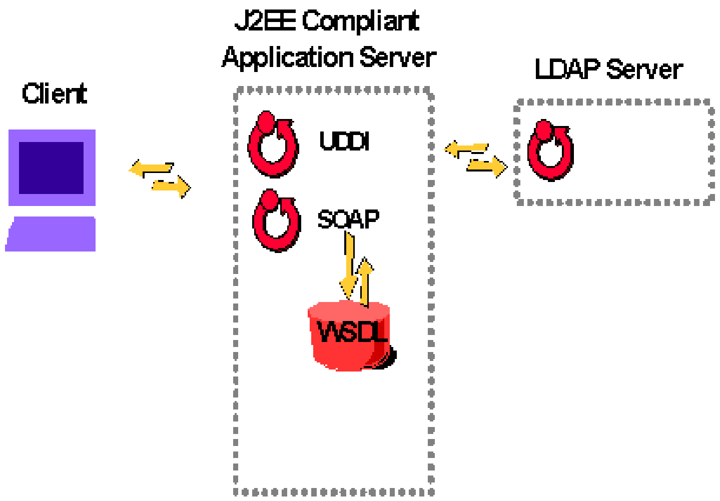


Figure 7-2 Web service terms.

SOAP (also referred to as Simple Object Access Protocol) sits atop HTTP and is the specific protocol that supports Web service invocation and delivery. A SOAP message is bundled inside an HTTP message. SOAP messages also have a header, and a body, and introduce the concept of an envelope, which has further routing and control data.

SOAP messages can also be viewed as ASCII text, and are XML formatted via a well defined standard, a standard XML schema defined to support Web services.

Continuing the telephone analogy, SOAP would be like speaking French or English with perfect syntax and vocabulary on a voice connection from your telephone.

Ê Other Web service related terms and ideas-

– A *WSDL file (Web Services Description Language,* pronounced "WizDull") is another XML formatted ASCII data file that defines a given Web service. This is another Web services related artifact that the tooling generates or consumes automatically for the developer.

A WSDL file is similar in concept to a Java abstract class; it lists the given Web service's identifier (name), routing instructions, and its various methods (functions), input and return values, and so on.

The WSDL file does not provide the program code for a given service, it merely describes what the service does. (Remember the concepts of loose coupling and decreased coupling.)

© Copyright IBM Corp. 2008 All Rights Reserved. US Government Users Restricted Rights-Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

- A *UDDI (Universal Description Discovery and Integration) server* is normally located inside an existing Java/J2EE compliant application server; it is often a subsystem to that server. Where a standard DNS server allows you to take a given computer name and locates its address on a network, a UDDI server does the same for Web services; where are they located, their descriptions (their WSDL files), and so on.
- An *LDAP (Lightweight Directory Access Protocol) server* is not directly related to Web services. An LDAP server would be one means for a Web service to authenticate the service requestor’s identity and provide for secure access to Web services and, in effect, network resources.
- An EAR (Enterprise Application Resource) file is the basic unit of deployment with Java/J2EE compliant applications and application servers. An EAR file is a Zip file with a rigidly defined structure to include; a standard table of contents (a standard manifest file), certain standard and required configuration files, the compiled application program code proper, and more.

Service oriented architecture (SOA)

Thus far we have defined model-view-controller, decreased (loose) application coupling, and why these are good things to make use of. Further, we defined how Web services work, and are the modern defacto standard to deliver loose coupling. Now it is time to define service oriented architecture (SOA).

SOA is not a synonym Web services, although it is often viewed as such. SOA could be delivered via technologies other than Web services, including technologies like CORBA (Common Object Request Broker Architecture). (CORBA was big in the late 1980s and early 1990s as a means to provide heterogeneous database access, two-phase commit, and related function.) The word architecture is at the root of service oriented architecture and is meant to define SOA as a *design principle*. Design principles are one layer above design patterns, discussed earlier, and both are logical terms, not physical instantiations of an idea.

As an architecture, SOA is meant to define numerous goals, including;

Ê Service abstraction; what is viewable as a service requestor, and what implementation details are kept hidden.

Ê Service autonomy; how services is encapsulated, recoverable, and what defines the concept as a recoverable unit of work.

Ê Service discoverability; including standards for UDDI and related.

Ê Service reusability; standards related to division of service.

Ê Service composability; the ability to coordinate and assemble compisite services.

© Copyright IBM Corp. 2008 All Rights Reserved. US Government Users Restricted Rights-Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

- Service optimization; guaranteed service levels, performance and availability.
- Others.

There are other terms in the area of SOA, like SCA (Service Component Architecture), SCM (Service Component Modelling), BPEL (Business Process Execution Language), and others. All of these terms hold promise and further refinement over SOA as a means to promote greater code re-use and discovery inside a corporate intranet.

IBM Information Server and Web services

As mentioned previously, the IBM Information Server (IIS) product is organized into several components, including; DataStage/QualityStage (DS/QS), Federated Server, WebSphere Information Services Director (WISD), Rational Data Architect (RDA), and others. The following is offered;

- DS/QS jobs can be Web service providers and/or Web service consumers.
- Federated Server queries can also be Web service providers and consumers.
- WISD is the IIS component we use to deploy Web services. Web services are hosted inside a Java/J2EE compliant application server. WISD automates and manages all of the technical detail involved in this process.
- RDA has many functional capabilities, In the context of Web services, one area in particular we use inside RDA is the Web Services Explorer (WSE). WSE was previously called the Universal Test Client (UTC). WSE allows you to view and retrieve WSDL files, invoke your own or third party Web services, view SOAP content, and more.

In the remainder of this document, we will take a very simple DS/QS job and expose it as a Web service. We will use WISD to deploy this Web service, and then use RDA to invoke and test this Web service.

7.2 Example DS/QS job deployed as Web service

To keep our example as simple as possible, we are going to create a DataStage/QualityStage (DS/QS) job that adds two numbers and returns a result. (Obviously there are more strategic examples of DS/QS jobs that one could provide as a Web service on the corporate intranet.) Also our example uses a DS/QS job. We do not demonstrate exposing a Federated Server query or stored procedure as a Web service, although the procedure is nearly identical.

Create the DS/QS job proper

1. Launch the DataStage/QualityStage Designer component of IBM Information Server.

© Copyright IBM Corp. 2008 All Rights Reserved. US Government Users Restricted Rights-Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

2. Create a new parallel job, equal to what is shown in Figure 7-3.
   a. From the DS/QS Designer -> Palette view -> Real Time drawer, drag and drop a WISD Input object, and a WISD Output object onto the parallel canvas
   b. From the Palette view -> Processing drawer, drag and drop a Transformer object onto the parallel canvas.
   c. Position and connect these three objects as shown in Figure 7-3. Rename each object as you see fit.

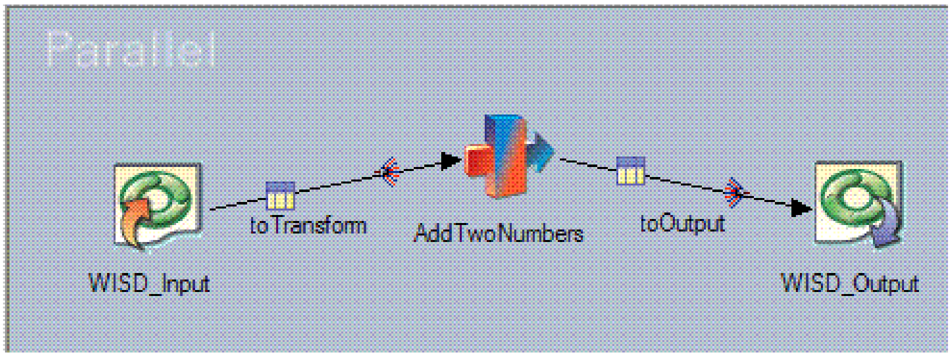


Figure 7-3 Basic DS/QS job using WISD Input and WISD Output.

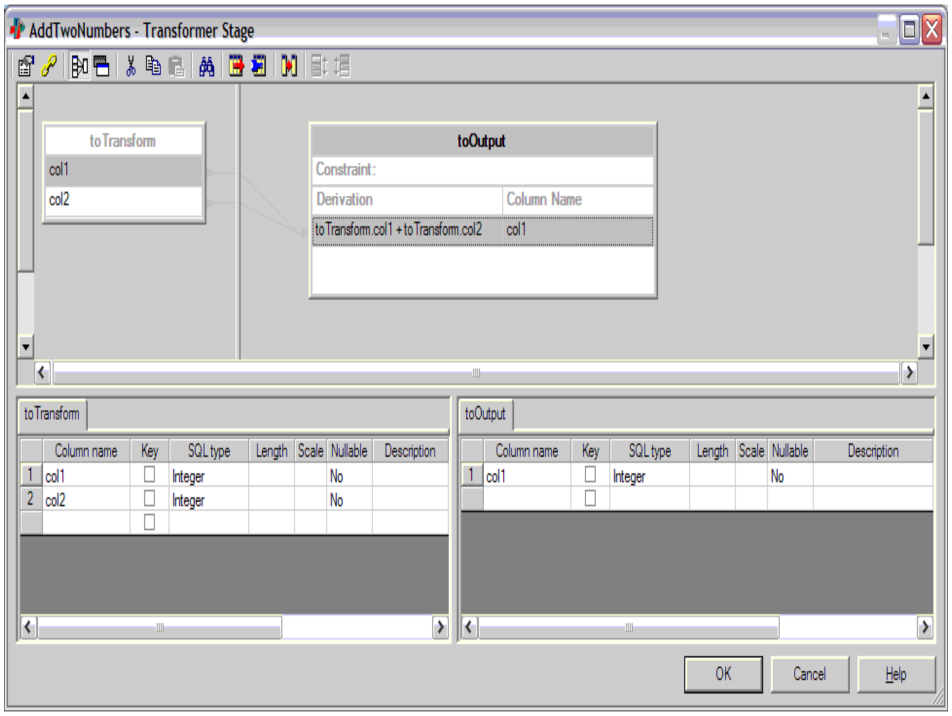
   d. Double-click the WISD Input object, and access Output -> Columns.
      i. Create two (input) columns of type integer. We called our columns, col1 and col2.
      ii. Click OK to save any changes.
   e. Double-click the Transformer object, and manage the user interface to equal what is shown in Figure 7-4.
      i. col1 and col2 should already appear in the upper left portion of the display; these are the two integer columns we created for the WISD Input object.
      ii. In the upper right portion of the display, under toOutput -> Column Name, right-click and select -> Insert New Column.
      iii. In the bottom right portion of the display, change the column name of the newly created output column, and set its SQL type to integer. We called our output column, col1.
      iv. Back in the upper right portion of the display, right-click on the line for col1, in the column entitled Derivation.

         Select -> Edit Derivation in the context menu that is produced.

© Copyright IBM Corp. 2008 All Rights Reserved. US Government Users Restricted Rights-Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Use the ellipsis button (“...”) to access -> Input Column. This shortcut will allow you to quick pick expressions like, toTransformer.col1, etcetera.

v. Finish by making this derivation equal to what is displayed in Figure 7-4, and click OK to save changes.



AddTwoNumbers - Transformer Stage

toTransform
- col1
- col2

toOutput

Constraint:

Derivation	Column Name
toTransform.col1 + toTransform.col2	col1

toTransform

	Column name	Key	SQL type	Length	Scale	Nullable	Description
1	col1	☐	Integer			No	
2	col2	☐	Integer			No	
		☐					

toOutput

	Column name	Key	SQL type	Length	Scale	Nullable	Description
1	col1	☐	Integer			No	
		☐					

Figure 7-4 Completed Transformer stage.

f. Double-click the WISD Output object, and access Input -> Columns. You should see one input column of type integer named, col1. No activity should be required here.

Click OK to exit.

At this point, the DS/QS job to support Web services is constructed, all that remains is some configuration of the job, and then deployment and test.

3. Set DS/QS job properties to support Web services.

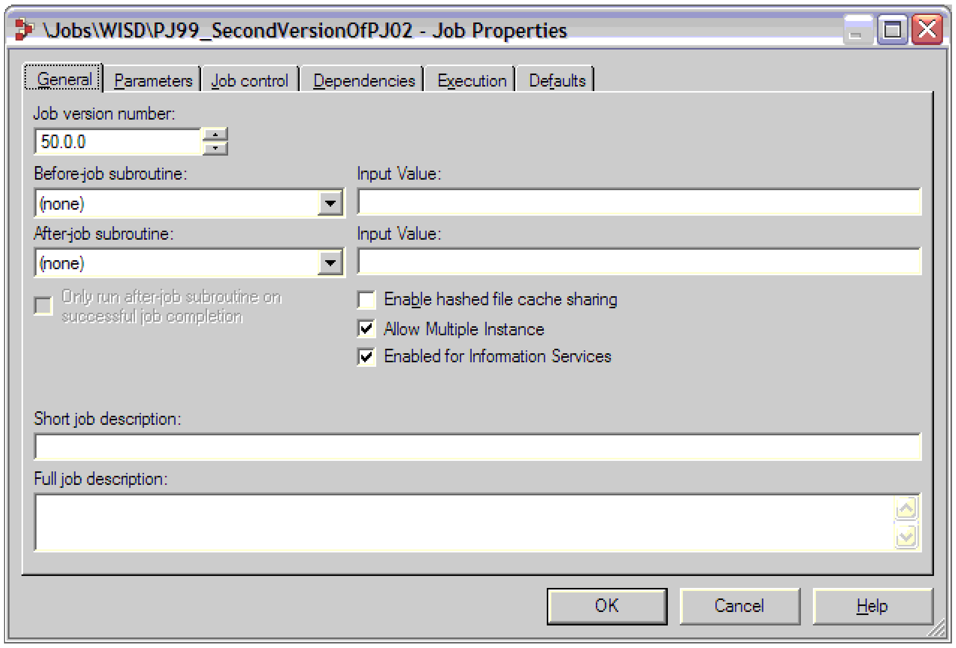
a. From the DS/QS Designer menu bar, select Edit -> Job Properties.

b. Under the General TAB, check both Allow Multiple Instances and Enabled for Information Services. Example as shown in Figure 7-5.

© Copyright IBM Corp. 2008 All Rights Reserved. US Government Users Restricted Rights-Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Checking these visual controls allows the given DS/QS job to operate as a Web service.

c. Click OK to save and exit.



\Jobs\WISD\PJ99_SecondVersionOfPJ02 - Job Properties

General | Parameters | Job control | Dependencies | Execution | Defaults

Job version number: 50.0.0

Before-job subroutine: (none)
Input Value:

After-job subroutine: (none)
Input Value:

☐ Only run after-job subroutine on successful job completion

☐ Enable hashed file cache sharing
☑ Allow Multiple Instance
☑ Enabled for Information Services

Short job description:

Full job description:

Figure 7-5   DS/QS job, Edit -> Job Properties.

4. Compile and save the DS/QS job.

   You are now done with the DS/QS Designer, and may exit this component of IIS.

Create an Information Services Connection

5. Launch the IBM Information Server Console component of IBM Information Server.

6. The next task we need to complete is to *create an Information Services Connection*.

   This task is equitable to creating a New Data Store when performing standard Information Analyzer activities. In effect, we are identifying the locations of the application server where our Web service will operate from, and the DataStage server where our source DataStage job is located.

   a. From the Home icon, select Configuration -> Information Services Connections. Example as shown in Figure 7-6.

© Copyright IBM Corp. 2008 All Rights Reserved. US Government Users Restricted Rights-Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

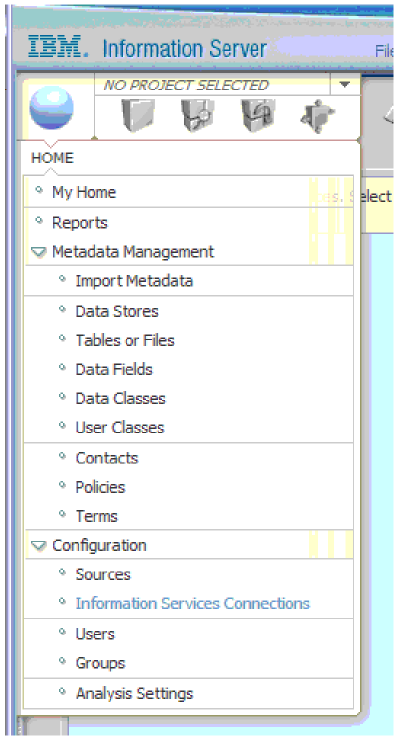


Figure 7-6 Creating an Information Services Connection.

b. Click the menu command in the upper right portion of the screen entitled, Tasks -> New.

c. Manage the user interface to equal what is displayed in Figure 7-7.

   i. Set the Connection Name to a value of your choosing. We always name our connections "somethingConn".

   ii. Set Information Provider Type equal to DataStage and QualityStage. (The other options is DB2 and/or Federated Server objects.)

   iii. Set Agent Host equal to your host name.

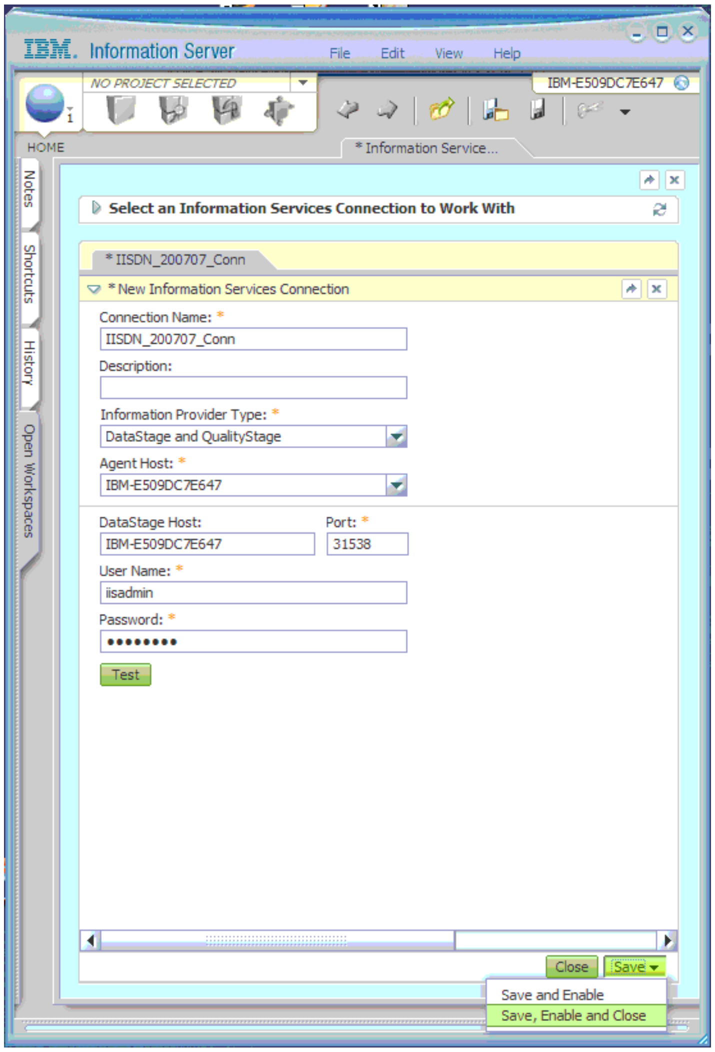
   This value is the name of the host where the Web services will operate, and where the Java/J2EE compliant application server was installed.

   In our case, this host was our development box, and WebSphere Application Server was our application server.

   iv. Enter the correct username and password, and click Test.

   v. Click Save -> Save, Enable and close when you are done.

© Copyright IBM Corp. 2008 All Rights Reserved. US Government Users Restricted Rights-Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.



Connection Name: * IISDN_200707_Conn
Description:
Information Provider Type: * DataStage and QualityStage
Agent Host: * IBM-E509DC7E647
DataStage Host: IBM-E509DC7E647
Port: * 31538
User Name: * iisadmin
Password: * ●●●●●●●●

Figure 7-7   Creating an Information Services Connection, second and final screen.

Create a WISD project and application

7. If you have not already done so, launch the IBM Information Server Console component of IBM Information Server.

8. The next task we need to complete is to *create an Information Services Project*.

   In this context, a project is a logical term, a container or grouping for our (possibly) numerous DataStage/QualityStage jobs.

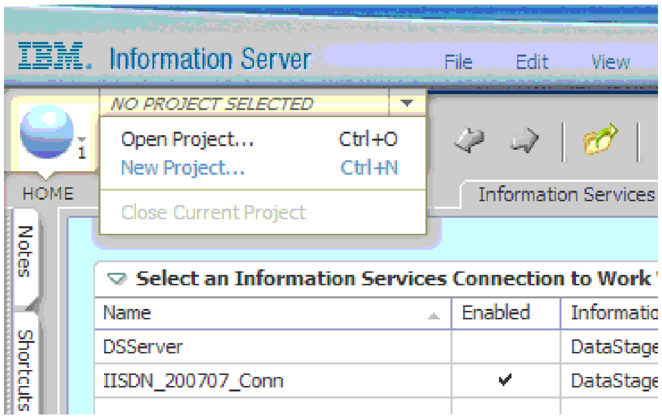
© Copyright IBM Corp. 2008 All Rights Reserved. US Government Users Restricted Rights-Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

a. From the tool bar, select {Current Project Name} -> New Project. If you do not have a current project, the value in this visual control displays, *NO PROJECT SELECTED*. Example as shown in Figure 7-8.

b. In the dialog box that is produced select a project type of, Information Services.

c. In this same dialog box, give the project a name. We called our project, "IISDN_200707_Proj".

d. Click OK to save and exit.



Name	Enabled	Informatio
DSServer		DataStage
IISDN_200707_Conn	✓	DataStage

Figure 7-8 Creating a new project inside IBM Information Server Console.

9. At this point we have a project, now we need to create an application. In this context, an application represents our (single) DataStage/QualityStage job, and the project represents a collection of jobs.

a. From the tool bar, select Develop -> Information Services Application. Example as shown in Figure 7-9.

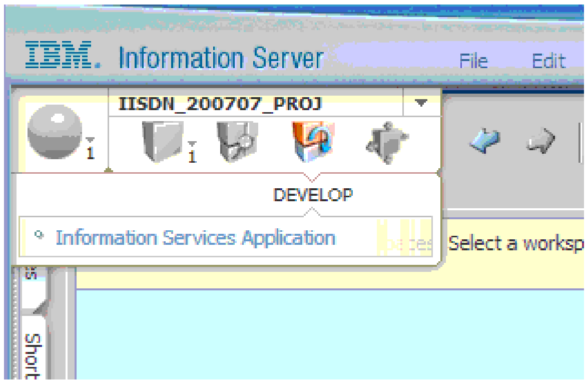
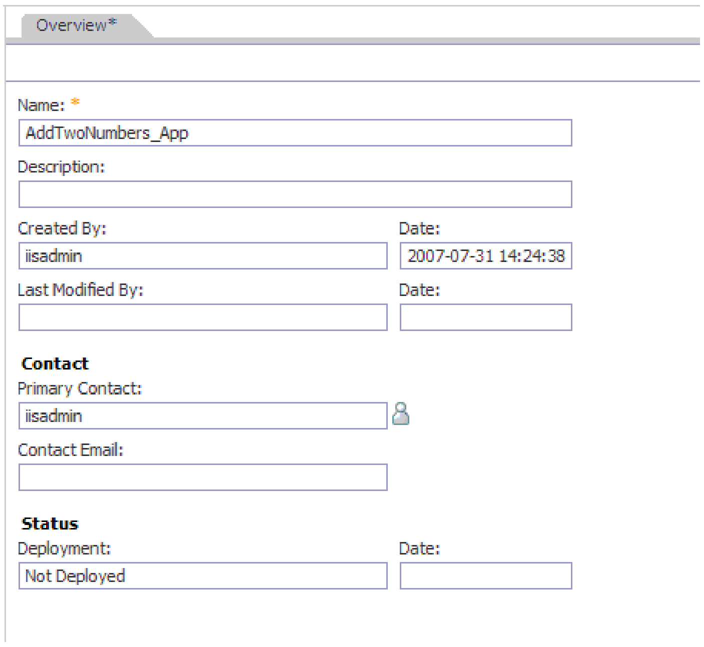


Figure 7-9 Creating a new application inside IBM Information Server.

b. Click the menu command in the upper right portion of the screen entitled, Tasks -> New.

c. In the next screen that is produced, we need only to enter an application name. Example as shown in Figure 7-10. We called our application AddTwoNumbers_App.



Name: *
AddTwoNumbers_App
Description:

Created By: iisadmin
Date: 2007-07-31 14:24:38
Last Modified By:
Date:

Contact
Primary Contact: iisadmin
Contact Email:

Status
Deployment: Not Deployed
Date:

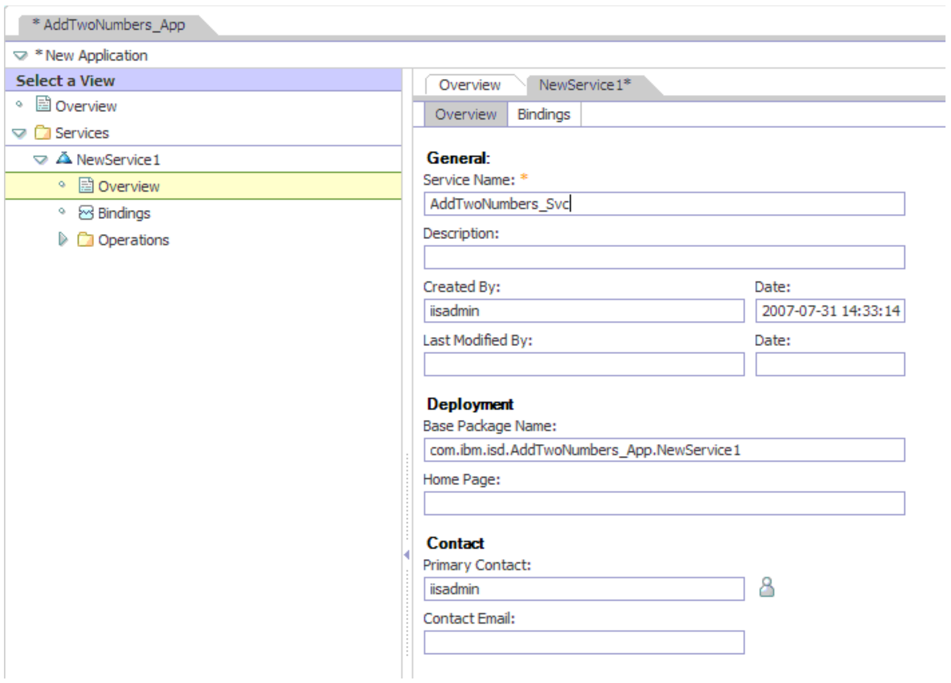
Figure 7-10 Creating a new application, Overview screen.

d. On the bottom right portion of the display, click the button entitled, Save Application.

e. On the left side panel of the display is the Select a View area, with a hierarchical display of Overview and then Services. There are no services for our application currently.

   Near the bottom of this panel is a drop down menu command, New -> Service. Select this option.

f. In this screen, we need to give this new service a name. We called our service, AddTwoNumbers_Svc. Example as shown in Figure 7-11.



Service Name: * AddTwoNumbers_Svc
Description:
Created By: iisadmin
Date: 2007-07-31 14:33:14
Last Modified By:
Date:
Base Package Name: com.ibm.isd.AddTwoNumbers_App.NewService1
Home Page:
Primary Contact: iisadmin
Contact Email:

Figure 7-11 Creating a new application, service overview screen.

g. Click the button entitled Save Application.

h. Move to the Bindings TAB, and click the drop down menu button entitled, Attach Bindings. Select -> SOAP over HTTP.

   All of the default values (SOAP Style, SOAP Action) are fine here.

   SOAP over HTTP is the communication protocol that our service will support; protocols and related were discussed above. In this context, a binding is just a recorded value.

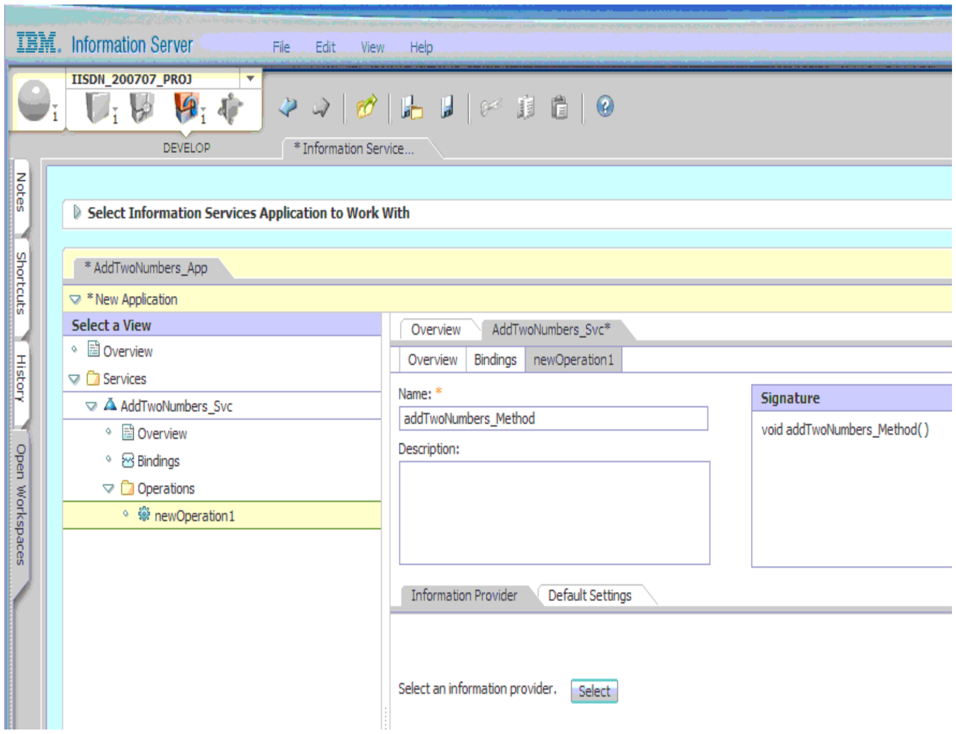
i. Once again, click the Save Application button.

© Copyright IBM Corp. 2008 All Rights Reserved. US Government Users Restricted Rights-Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

10. Thus far we have created a project and an application. now we are at the point where we actually add our DataStage/QualityStage job.

   a. On the left side panel of the display is the Select a View area, with a hierarchical display of Overview and then Services. Browse and expand the available selection until you see an try entitled, Operations -> newOperation1.

      Double-click the newOperation1 entry. Example as shown in Figure 7-12.



Name: * addTwoNumbers_Method

Description:

void addTwoNumbers_Method( )

Select an information provider.

Figure 7-12 Creating a new application, new operator screen.

   b. In the new operator screen, change the Name to equal a value other than newOperator1. We called our operator, addTwoNumbers_Method.

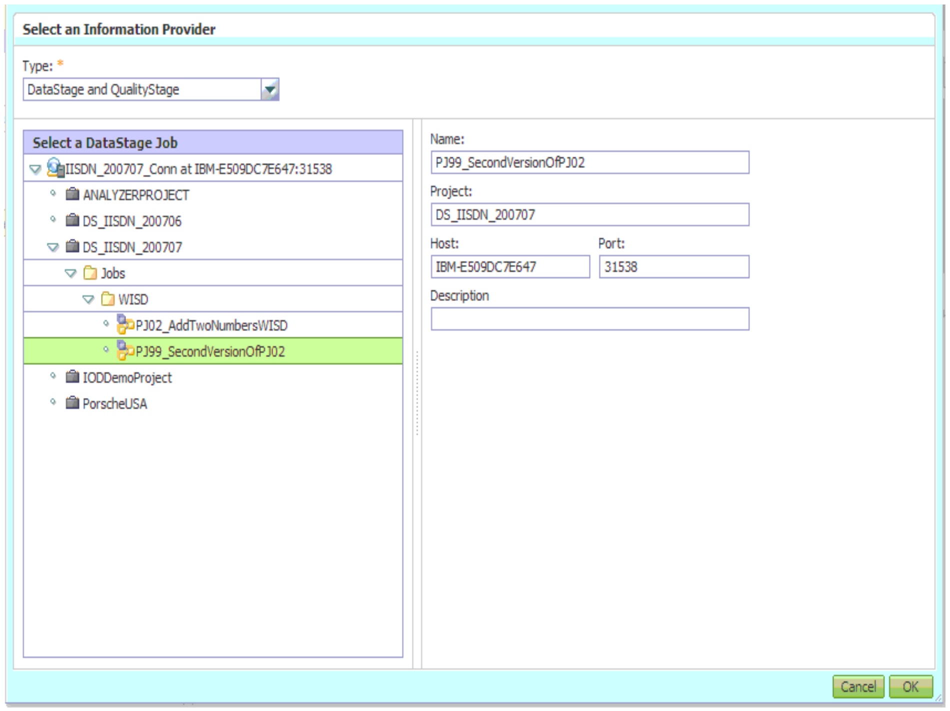
      Java allows for a given application to have numerous methods or functions. In our case, our DataStage/QualityStage job only has one method, which basically is to run; we could have numerous DataStage/QualityStage jobs, each as its own application.

   c. Click the button entitled, Select an information provider -> Select.

   d. In the drop down list box entitled, Select an Information Provider -> Type, choose DataStage and QualityStage.

© Copyright IBM Corp. 2008 All Rights Reserved. US Government Users Restricted Rights-Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

This action will populate a hierarchical list of DataStage Servers, and then Projects, and then Jobs from which we may choose. Example as shown in Figure 7-13.


Select an Information Provider

Type: DataStage and QualityStage

Select a DataStage Job

- IISDN_200707_Conn at IBM-E509DC7E647:31538
  - ANALYZERPROJECT
  - DS_IISDN_200706
  - DS_IISDN_200707
    - Jobs
      - WISD
        - PJ02_AddTwoNumbersWISD
        - PJ99_SecondVersionOfPJ02
  - IODDemoProject
  - PorscheUSA

Name: PJ99_SecondVersionOfPJ02

Project: DS_IISDN_200707

Host: IBM-E509DC7E647

Port: 31538

Description:

Figure 7-13 List of available DataStage/QualityStage Servers, Projects, Jobs.

e. In Figure 7-13, navigate to your actual DataStage Server, Project and then Job.

The values displayed in Figure 7-13 are; IISDN_200707_Conn at {hostname:port} -> DS_IISDN_200707 -> Jobs -> WISD -> PJ99_SecondVersionOfPJ02.

- IISDN_200707_Conn is the Information Services Connection we made above.
- {hostname:port} are properties from that connection object.
- DS_IISDN_200707 is the name of our DataStage/QualityStage project created with the DataStage/QualityStage Administrator.
- Jobs is a standard folder/grouping in the DataStage/QualityStage Designer.
- WISD is a new folder/grouping we made under Jobs.

© Copyright IBM Corp. 2008 All Rights Reserved. US Government Users Restricted Rights-Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

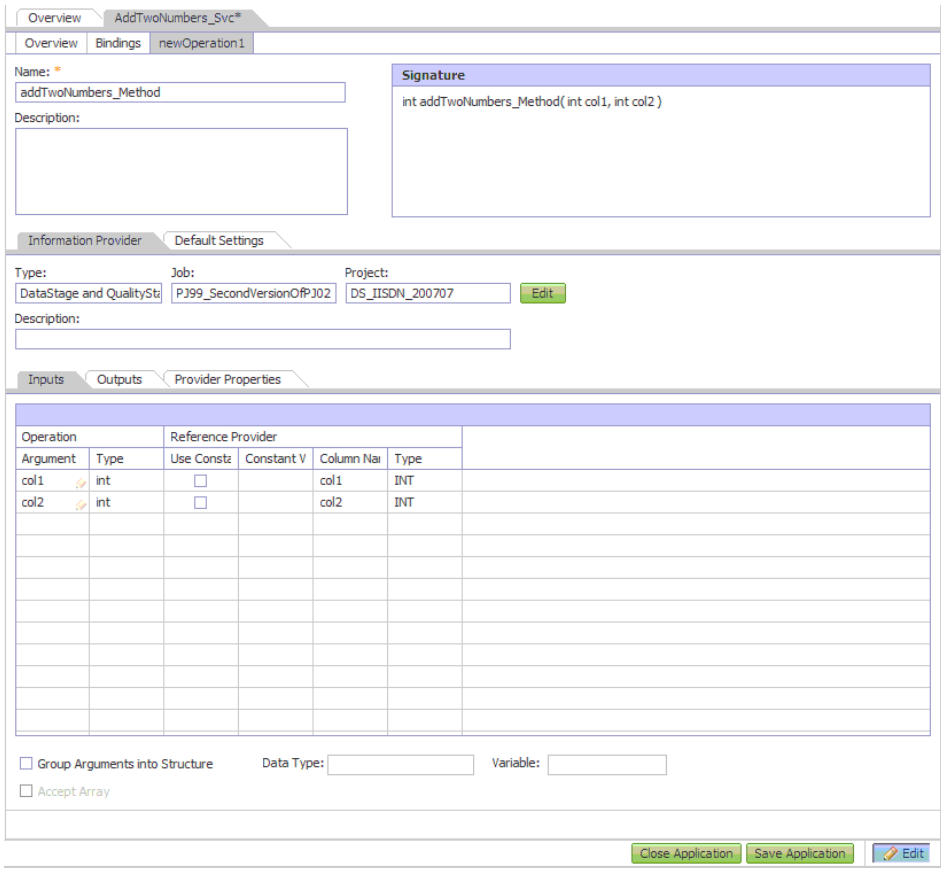
- PJ99_SecondVersionOfPJ02 is the name of the DataStage/QualityStage job we made above. (The first version was a test using shared containers. The version we display in this document is simpler. Both operate fine.)

f. When you are done navigating to your job, click OK.

g. Figure 7-14 displays the result of the step above. Figure 7-14 is a display very similar to most DataStage/QualityStage property screens.

All of these default values were imported from the DataStage/QualityStage job. No activity is required here.

h. Click the Save Application button.



Name: *
addTwoNumbers_Method
Description:

Signature
int addTwoNumbers_Method(int col1, int col2)

Type: DataStage and QualitySta
Job: PJ99_SecondVersionOfPJ02
Project: DS_IISDN_200707
Description:

Operation		Reference Provider			
Argument	Type	Use Consta	Constant V	Column Nai	Type
col1	int	☐		col1	INT
col2	int	☐		col2	INT

☐ Group Arguments into Structure   Data Type: Variable:
☐ Accept Array

Figure 7-14 Final step in creating an application.

At this point we have created our project and application capable of supporting Web services. Now we are ready to deploy this project, and this activity is also done in the IBM Information Server Console.

Deploying a WISD project

11. If you have not already done so, launch the IBM Information Server Console component of IBM Information Server.

12. The next task we need to complete is to *deploy our WISD project*, containing our DataStage/QualityStage job.

    a. From the tool bar, select Develop -> Information Services Application. Example as shown in Figure 7-9.

    b. Highlight the entry entitled, AddTwoNumbers_App, and select Tasks -> Deploy, from the menu on the right side of the display.

    c. You are presented with a number of stepped screens where you can adjust binding protocols and the like. All of these settings are correct based on our prior activity.

       In the bottom right area of the display is a button to shortcut all of these steps; Click the button entitled, Deploy.

       This action will deploy our DataStage/QualityStage job, in the form of a standard Java/J2EE EAR File, to the specified application server. This step will present a Progress Bar, and may take some time to complete based on the capacity of your computer and/or network.

    d. When complete, the status column entitled, Deployment Status, will display Deployed, and a date and time. Your Web services application is now deployed.

We are now done with the IBM Information Server (IIS) Console. You may exit this component to IIS now. The remainder of our activities take place in the IIS Web Console, and Rational Data Architect.

Testing a Web service

In order to test our newly created Web service (or any Web service), we are going to invoke it via a supported protocol, which in our case is over standard HTTP. In order to invoke this or any Web service, we need to know the Web service's HTTP address.

We could locate a given Web service by browsing a Web services directory (a UDDI server), or by reverse engineering and locating the given Web service's WSDL file. Since our Web service is not currently listed in any UDDI server, we will choose the second path of reverse engineering and finding the WSDL file. This activity is completed using the IBM Information Server Web Console.

© Copyright IBM Corp. 2008 All Rights Reserved. US Government Users Restricted Rights-Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

13. Find the WSDL file associated with our newly created Web service.

   a. Log on to the IBM Information Server Web Console. As a Web application, this (program) is accessed through as standard Web browser.

   b. Select the Information Services Catalog TAB, and then Manage Deployed Services.

   c. Check the AddTwoNumbers_Svc, and click Open from the menu on the right side of the display.

   d. Under the Select a View area, click Binding.

   e. Then expand the selection entitled, SOAP over HTTP.

   f. Click the link entitled, Open WSDL Document. Example as shown in Figure 7-15.

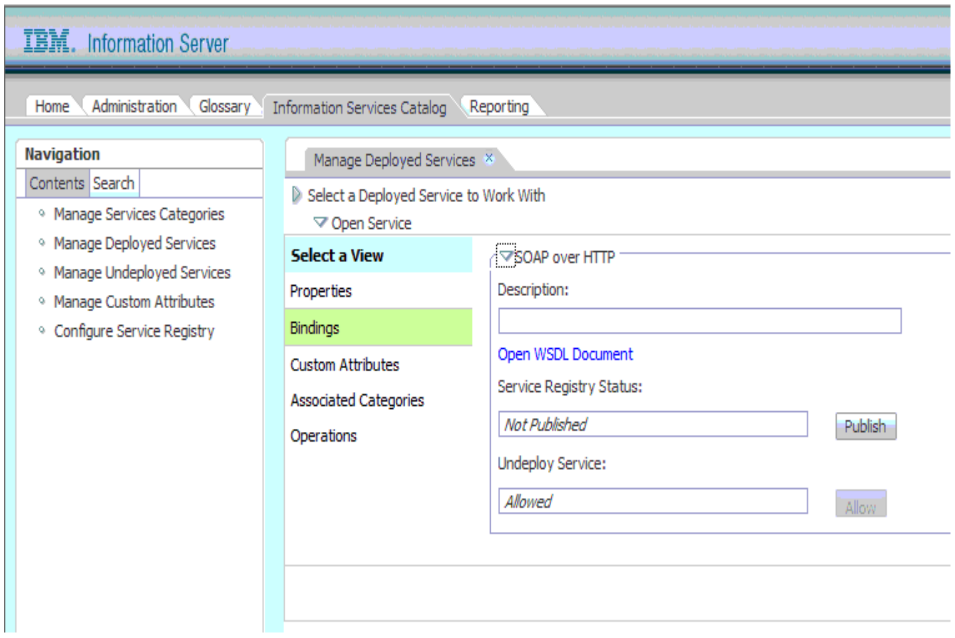
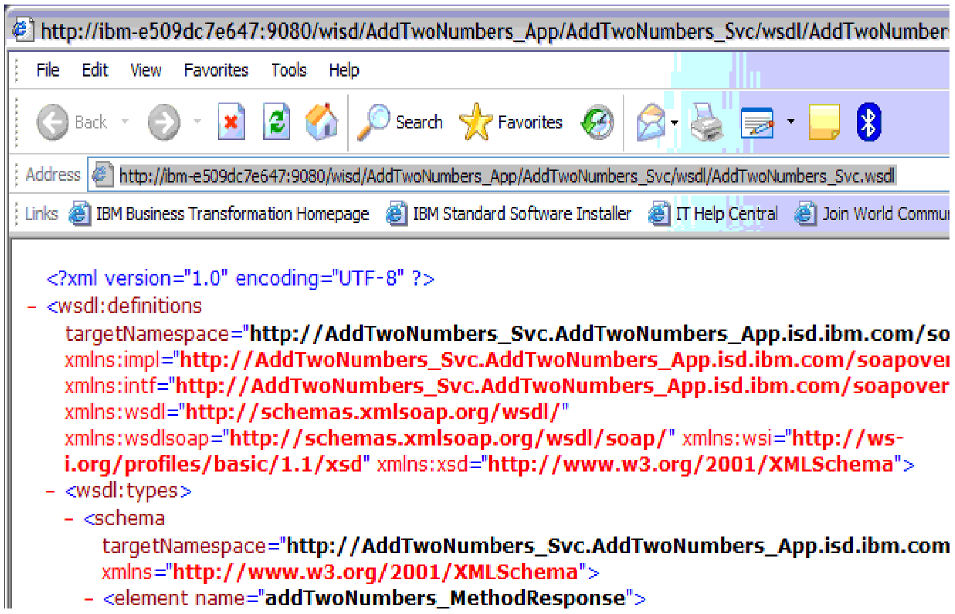


Figure 7-15 IIS Web Console -> Information Services Catalog -> Bindings

This action will open the WSDL document for this, our Web service, inside a new Web browser window.

What we really want here is the URL address in the Web browser window. Example as shown in Figure 7-16.

© Copyright IBM Corp. 2008 All Rights Reserved. US Government Users Restricted Rights-Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.


http://ibm-e509dc7e647:9080/wisd/AddTwoNumbers_App/AddTwoNumbers_Svc/wsdl/AddTwoNumber

Address: http://ibm-e509dc7e647:9080/wisd/AddTwoNumbers_App/AddTwoNumbers_Svc/wsdl/AddTwoNumbers_Svc.wsdl

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions
    targetNamespace="http://AddTwoNumbers_Svc.AddTwoNumbers_App.isd.ibm.com/so
    xmlns:impl="http://AddTwoNumbers_Svc.AddTwoNumbers_App.isd.ibm.com/soapover
    xmlns:intf="http://AddTwoNumbers_Svc.AddTwoNumbers_App.isd.ibm.com/soapover
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsi="http://ws-
    i.org/profiles/basic/1.1/xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  - <wsdl:types>
    - <schema
        targetNamespace="http://AddTwoNumbers_Svc.AddTwoNumbers_App.isd.ibm.com
        xmlns="http://www.w3.org/2001/XMLSchema">
      - <element name="addTwoNumbers_MethodResponse">
```

Figure 7-16 WSDL file displayed in Web browser. URL address is what we want.

14. The Rational Data Architect (RDA) component of IBM Information Server has a capability called the Web Services Explorer (WSE). This capability within RDA gives us everything we need to test our Web service.

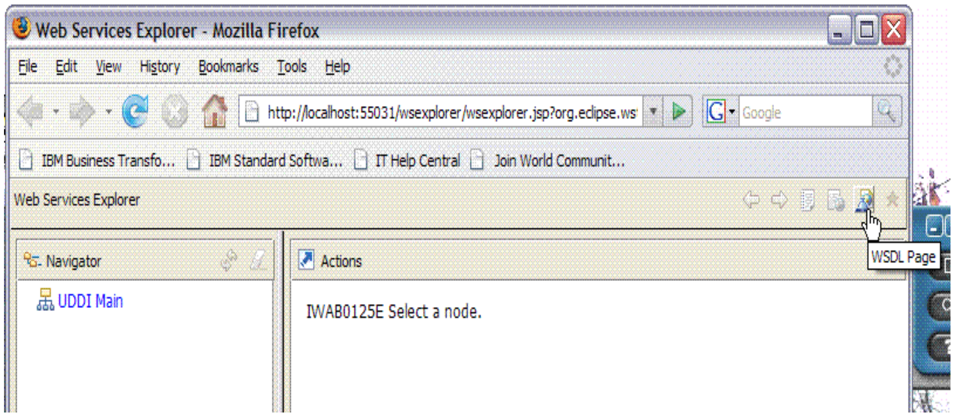
    a. Launch RDA.

    b. From the main menu bar select, Run -> Launch the Web Services Explorer.

       The Web Services Explorer (WSE) runs as a Web application inside a Web browser. Do not exit RDA; based on your configuration, RDA is hosting part of this application.

    c. From the WSE tool bar, click the WSDL Page icon. Example as shown in Figure 7-17.

       The WSE tool can explore UDDI registries, WSIL files and more. At this time we wish to load our WSDL file, located earlier.


Web Services Explorer - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:55031/wsexplorer/wsexplorer.jsp?org.eclipse.ws
Google
IBM Business Transfo... IBM Standard Softwa... IT Help Central Join World Communit...
Web Services Explorer
WSDL Page
Navigator
UDDI Main
Actions

IWAB0125E Select a node.

Figure 7-17   RDA, Web Services Explorer

d. On the left side of the display, click the link entitled, WSDL Main.

e. In the text entry field entitled, WSDL URL, paste the WSDL file address we reverse engineered above.

f. Click Go.

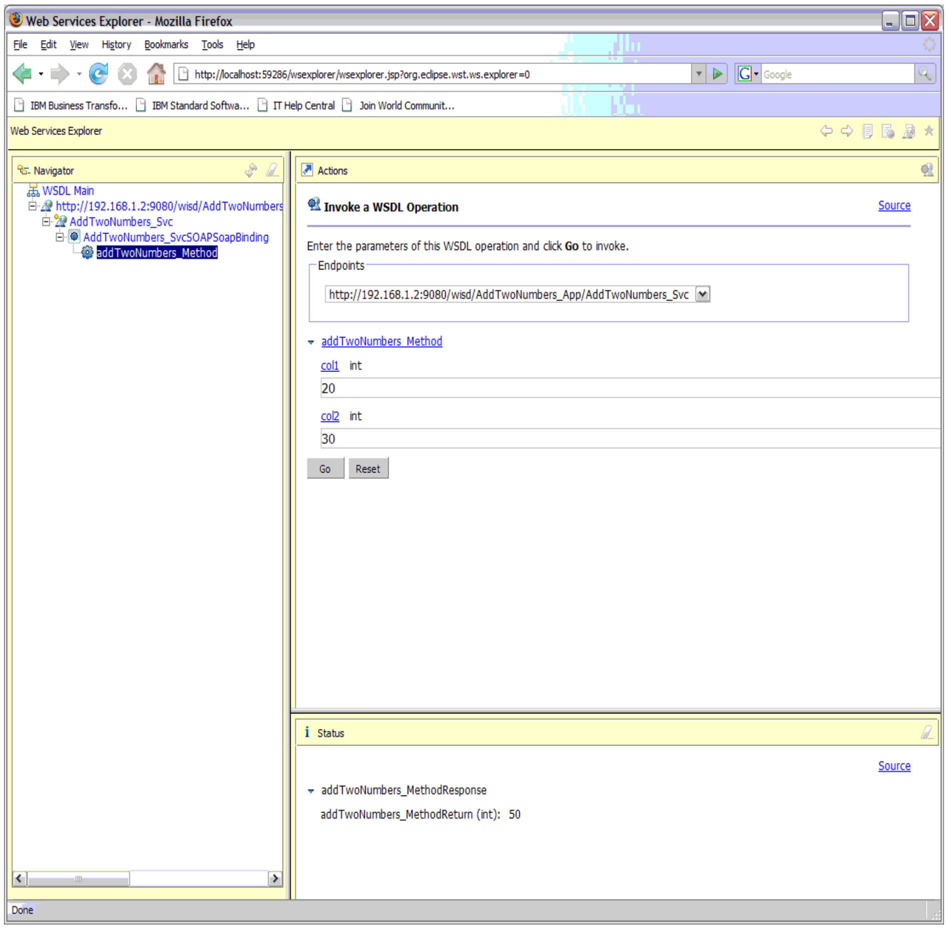
g. The action above will retrieve the WSDL file for our Web service containing our DataStage/QualityStage job.

   Manage the user interface to equal what is displayed in Figure 7-18. Essentially you want to navigate the tree on the left and highlight addTwoNumbers_Method.

   Enter values for the two input columns, col1 and col2, and click Go.

   Results will appear in the window below.

© Copyright IBM Corp. 2008 All Rights Reserved. US Government Users Restricted Rights-Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.


Web Services Explorer - Mozilla Firefox
File Edit View History Bookmarks Tools Help
IBM Business Transfo... IBM Standard Softwa... IT Help Central Join World Communit...
Web Services Explorer
Navigator
Actions
Go
Reset
Status
Done

http://localhost:59286/wsexplorer/wsexplorer.jsp?org.eclipse.wst.ws.explorer=0

WSDL Main
http://192.168.1.2:9080/wisd/AddTwoNumbers
AddTwoNumbers_Svc
AddTwoNumbers_SvcSOAPSoapBinding
addTwoNumbers_Method

Invoke a WSDL Operation — Source

Enter the parameters of this WSDL operation and click **Go** to invoke.

Endpoints: http://192.168.1.2:9080/wisd/AddTwoNumbers_App/AddTwoNumbers_Svc

addTwoNumbers_Method

col1 int: 20

col2 int: 30

Source

addTwoNumbers_MethodResponse

addTwoNumbers_MethodReturn (int): 50

Figure 7-18 Final result.

7.3 Summation

In this document, we reviewed or created:

We created and published a Web service hosted inside a Java/J2EE compliant application server that is fully network accessible and production capable. If you can, go to another workstation on your network and invoke this same Web service; it works, and its ready for business. The functionality of this Web service is provided for by a standard DataStage/QualityStage job, using the WISD Input and WISD output objects.

© Copyright IBM Corp. 2008 All Rights Reserved. US Government Users Restricted Rights-Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

After creation of our DataStage/QualityStage (DS/QS) job using the DS/QS Designer, the IBM Information Server Console was used to package and deploy this job. Our standard DS/QS job is now accessible via HTTP and allows the ability of loosely coupled activities inside a corporate intranet. Following the concept of model-view-controller, organizations may choose to create on line information store of both data and data cleansing routines to allow for more mobile and responsive information technology organizations.

Additional resources:

An IBM Redbook, currently in the draft phase, is available on this topic at,

http://www.redbooks.ibm.com/abstracts/sg247402.html?Open

Legal statements:

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating trademarks that were owned by IBM at the time this information was published. A complete and current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

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

Special attributions:

The listed trademarks of the following companies require marking and attribution:

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

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 trademarks of Microsoft Corporation in the United States, other countries, or both.

Microsoft trademark guidelines

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.

Intel trademark information

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

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

© Copyright IBM Corp. 2008 All Rights Reserved. US Government Users Restricted Rights-Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.