Jetty Offers Legacy Programs via Web Services

Enterprise Manager's new Web Services tool makes it easy to connect your application to BBj®'s new built-in Jetty Server, and ultimately to the world. The Jetty HTTP Server opens your application's API to clients anywhere on the Internet. Reliance on Internet standards such as HTTP and XML simplifies interoperability. Web Services help bridge the gap between BBj and other languages. If you need to get two systems working together to share data or business logic, whether they are across the room or across the globe, Web Services are a great solution. This article explores BBj's ability to turn an existing application into a Web Service with very little effort.

Create a Web Service

Enterprise Manager's new Web Service Configuration tool makes it easy to create a new Web Service. Selecting a name, working directory, config file, and namespace (see **Figure 1**) creates a new empty API, ready to populate with methods from your application.

BBj implements a Web Service method as a simple CALL to a BBx® program. Adding a new method to the API requires filling in only a few fields. First, select the public name of the method; the method name the Web Service clients will see in the >>

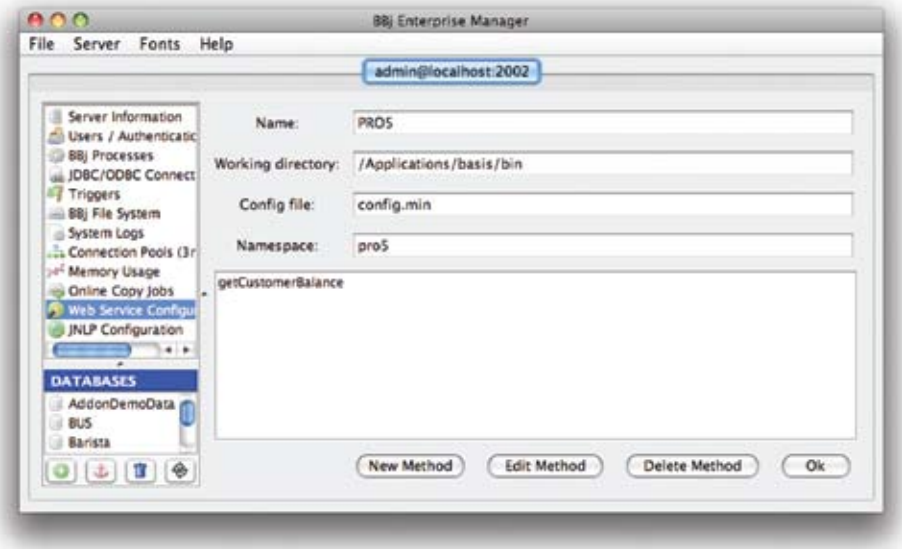

Figure 1. A Web Service defined in Enterprise Manager

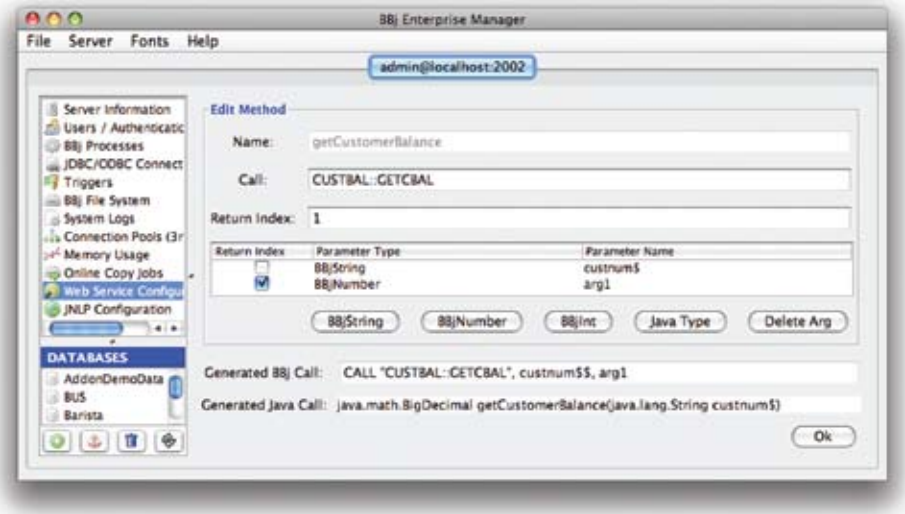

Figure 2. The getCustomerBalance Web Service method

By Jason Foutz
Software Programmer

public API for your Web Service. This example uses "PRO5" as the name of the Web Service, and getCustomerBalance as the method name, a long name to help users of the service determine the purpose of the method. Next, provide the name of any BBx program that includes an ENTER statement. The CUSTBAL program looks up customer balances. Add the CALL arguments the client should provide. The GETCBAL entry point requires a BBjString to identify the customer, and a BBjNumber to report the current balance. In BBx, all the variables of an ENTER statement are visible to the calling program, but a Web Service method may only return one variable to the client. Selecting a check box next to one of the listed types will instruct the Web Service method to use the argument at that position as the return value, as shown in **Figure 2**.

The Web Service named "PRO5" in the configuration dialogs defines a getCustomerBalance method. Any valid BBx program can provide the implementation of this method. The CUSTBAL program defines an entry point, GETCBAL that looks up customer balances, as shown in **Figure 3**.

```
0010 GETCBAL:
0020 ENTER CUSTNUM$,BAL
0030 LET CHAN=UNT
0040 LET TMPLT$="CUST_NUM:C(6):LABEL=CUST_NUM:, FIRST_NAME
0050 OPEN (CHAN)"data/CUSTOMER"
0060 DIM RESULT$:TMPLT$
0070 READ RECORD(CHAN,KEY=CUSTNUM$,ERR=0090)RESULT$
0080 LET BAL=RESULT.CURRENT_BAL
0090 CLOSE (CHAN)
0100 EXIT
```

Figure 3. The getCustomerBalance method for the "PRO5" Web Service

```
- <definitions targetNamespace="pro5" name="PRO5Service">
    <import namespace="http://bbjgenerated/" location="http://localhost:8888/webservice/PRO5?wsdl=1"/>
  - <binding name="PRO5ServicePortBinding" type="ns1:PRO5">
      <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    - <operation name="getCustomerBalance">
        <soap:operation soapAction=""/>
      - <input>
          <soap:body use="literal"/>
        </input>
      - <output>
          <soap:body use="literal"/>
        </output>
      </operation>
    </binding>
  - <service name="PRO5Service">
    - <port name="PRO5ServicePort" binding="tns:PRO5ServicePortBinding">
        <soap:address location="http://localhost:8888/webservice/PRO5"/>
      </port>
    </service>
  </definitions>
```

Figure 4. The WSDL served by BBj

Publish the Web Service

A Web Service does not provide any value unless the world can reach it. To make the "PRO5" Web Service available, you must publish the Web Service. In BBj, publishing the service makes the API available using the embedded Jetty Web Server. An XML document called a WSDL document exposes the Web Service API and associated information to Web Services tools of any programming language. To retrieve the WSDL document, simply use a Web browser to navigate to http://localhost:8888/webservice/PRO5?wsdl. An example of the document generated for the "PRO5" service appears in **Figure 4**.

As mentioned above, Jetty, a small and efficient Web Server, makes the Web Service available. Because Jetty uses standard technologies, both system and network administrators understand how to configure their systems and networks to provide access to it. Developers could write a custom networking protocol, client, and server in BBj, but such an implementation is challenging at best. Using the standard tools, technologies, protocols, and automatic features provided by BBj reduces the difficulty both for developers and administrators.

```
wsimport -B-wsdl http://localhost:8888/webservice/PRO5?wsdl
```

Figure 5. Create Web Service clients

Consume the Web Service in BBj

The Web Service is published and now waiting for a client to connect. The next logical step is to write a client that uses the WSDL document to provide an API for a Web Service client to use.

Java provides tools to automatically generate Java code to provide the Web Service API described by the WSDL document. Sun-derived JDK implementations provide this ability in a tool called wsimport, located in the **`jdk/bin`** directory. Issue the command in **Figure 5** to generate a set of Java classes.

These classes connect to your application, allowing BBj programs >>

```
rem  consume PRO5 web service

service! = new service.PRO5Service()
port! = service!.getPRO5ServicePort()
balance = port!.getCustomerBalance("000010")
print balance
exit
```

Figure 6. A sample BBj client to the "PRO5" Web Service

to use your methods. Such tools exist for many languages; the exact steps for constructing a client vary from language to language, but the core idea of using XML to represent the information being passed from the client to BBj and back remains the same. BBj requires a reference to the newly created classes. Therefore, it needs to be added to your classpath.

After Java has generated the Java classes to link a client to a Web Service and BBj is configured to use them, the code in **Figure 6** can be used to consume the Service.

Now, clients on any platform can call "PRO5" and get customer balances... even on their mobile device (see **Figure 7**)!

Summary

BBj provides all the tools necessary to open your application to the world. Enterprise Manager helps design an API that gives your clients the access they need to your applications. By leveraging HTTP and XML, configuration is far simpler than the alternatives. Web Services can even ease the pain of communicating with applications written in other languages. Make your application available to anyone anywhere with Web Services. ■

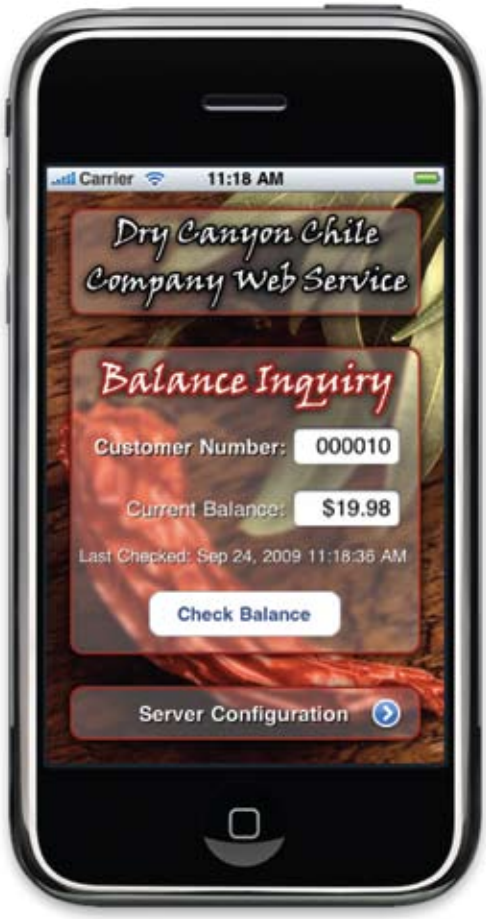

Figure 7. A mobile device using the "PRO5" Web Service

Turn a legacy (V)PRO/5 application routine into a Web Service

1. Install the latest version of BBj.

2. Follow the steps to search for reserved words in BBj using the (V)PRO/5 _keyword utility as documented in the online topic *Converting to BBj from Earlier Versions of BASIS Products* to ensure that the tokenized (V)PRO/5 application routine runs as expected in BBj.

3. Configure BBj to use Shared Locks so that it can simultaneously access data files that are used by your (V)PRO/5 application routine.

4. Create a new Web Service in Enterprise Manager.

5. Add a method to the Service that references your application routine.

6. Publish the Service.