

# Personalizing Your PowerBuilder Applications

## Open up a window of possibilities

ARTICLE BY CHANCE



**T**ext messaging is cool. Voice messaging is even cooler. Wouldn't it be great to be able to add that functionality to your PowerBuilder apps as well? For example, when an event occurs within your application, let your users receive real-time notification via SMS or voice messaging to their cell phones! This article will focus on writing Web services to do just that using the Strike Iron SMS and PhoneNotify messaging services.

After reading this article you should be able to:

- Have an understanding of PowerBuilder 10.5.1 Web services
- Send SMS and voice messages out of your PowerBuilder applications
- Utilize PowerBuilder 10.5.1 custom SOAP headers in your .NET Web services calls

Figure 1 shows what the finished product will look like. Now, let's go about building it!

### Reflection

It's been almost a year since I last visited Web services. For the most part, the steps for creating one are the same but I'm not. (I forget more things.) That being said, I thought I'd take this opportunity to review from the ground up how you go about consuming a Web service in PowerBuilder. Let's take a look.

The first thing you'll want to do is to include the necessary SOAP support in your project. This is encapsulated in a PBD called pbwsclient105.pbd and is located in your .../PowerBuilder/Shared directory. Add it to your project as shown in Figure 2. [Editor's Note: PowerBuilder 10.5 also allows you to import the PBNI objects from the PBX/DLL file directly into a PBL rather than using a separate PBD.]

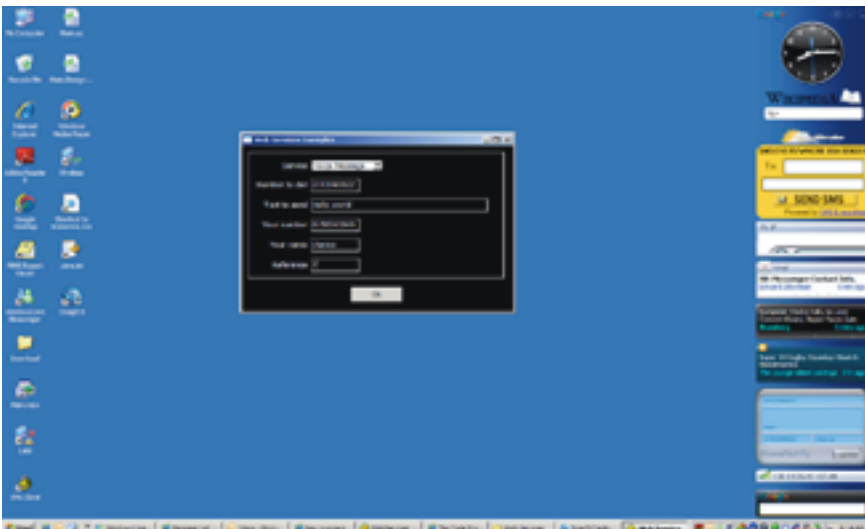
The next thing you will want to do is create a Web service proxy. The easiest way to do this is to access the Web Service Proxy Wizard. You can find it by choosing File->New and choosing the project tab (see Figure 3). The first thing the wizard will ask you is whether you want to use the .NET engine (see Figure 4). You should check the box indicating yes. In case you're wondering, the alternative is to use the EasySOAP implementation of Web services. This is the precursor to the current incarnation and is no longer being developed nor does it support custom headers. Unless you have some kind of backward compatibility issues, you would want to use the newer .NET engine. Next, specify the WSDL location (Web Services Description Language) that will bring in the functions and structures needed to call the Web services you are interested in (see Figure 5). Finally, when you have finished the wizard, you'll need to deploy your proxy. You can do this by going to Design->Deploy Project while the proxy definition is open in the painter (see Figure 6). You are now ready to call your Web services.

Calling a Web service is fairly straightforward. There is just a little setup work to do before the actual call. Namely, you'll need to add script similar to this. Of course, you will want to substitute in your own proxy names but the setup is the same.

```
//phone notify service
p_phonotify lws_phoneservice
```

```
//Soap connection
SoapConnection sc
```

```
sc = CREATE SoapConnection
sc.CreateInstance(lws_phoneservice, "p_phonotify")
```



▲ Figure 1

### Example Usage

There is no doubt that this is a fun project but does it have any business application? I would argue, yes. For example, I used to work on inventory control software for bookstores. Processing special orders was one area that would require customer notification. The way it worked was a customer would special order a title and that information was captured in the system. Then, when the book arrived from the publisher, it would be flagged to be taken away from regular inventory and the clerk would phone the customer to let them know their item was available for pickup. Why not let the computer call them as part of that processing? Perhaps it's not as personal, but it's certainly more efficient. In fact, you could probably make a case for automating any event in your systems that requires unattended user notification.

### PhoneNotify Voice Messaging Service

The PhoneNotify Voice Messaging Service allows you to deliver on-demand voice notifications, including alerts and two-way interactive voice messaging without expensive hardware, software, or even a dedicated platform. You may have received one of these before: a robot voice phone call for lack of a better term (although, you can use your own voice). If you are interested in establishing your own account, you can do so for a fee of \$4.99/month plus a nominal per-message charge. Here's a link to the sign-up page: <https://secure.cdyne.com/QuickNotifySignup/>.

### Sample Code

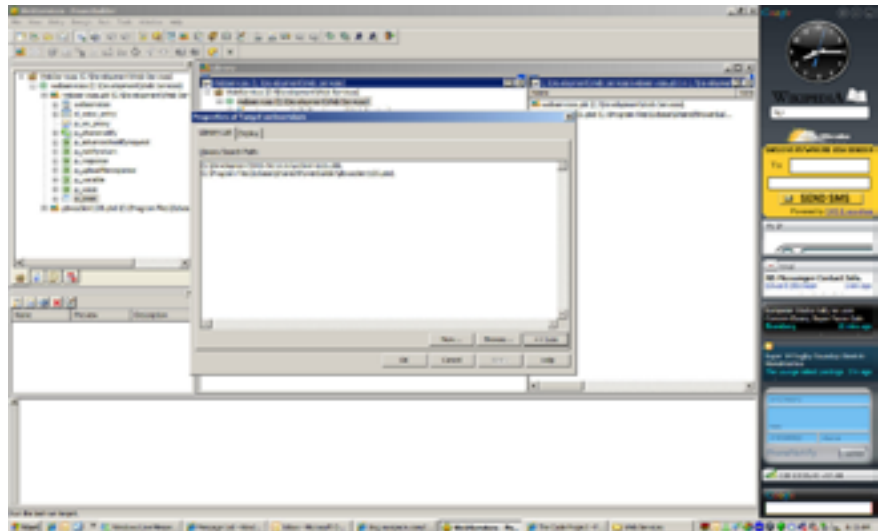
If you look Listing 1, most of it is data entry validation. In fact, the workhorse of the function (i.e., the code that actually sends a voice message) is simply a one liner. Easy.

### The Strike Iron Text Message Service

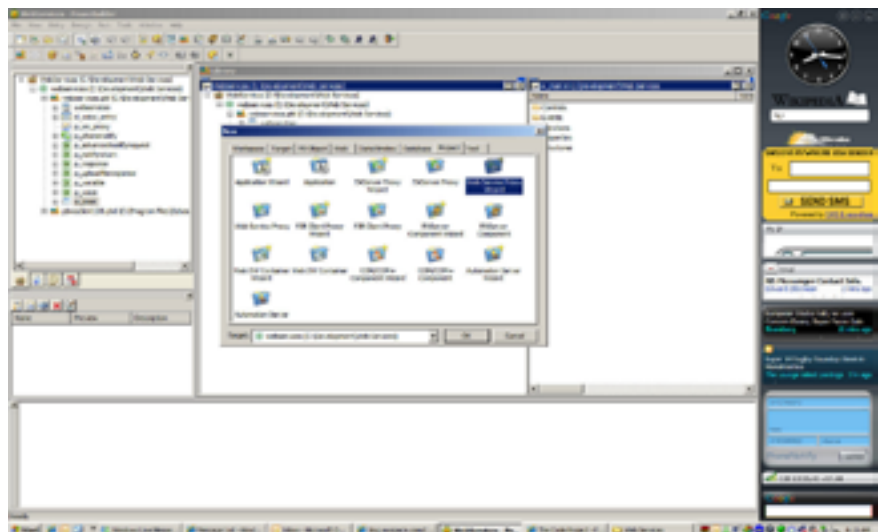
The Strike Iron Global SMS Pro Web Service enables an SMS (text) message to be sent to virtually any mobile phone instantly. This service supports over 416 carriers in 160 countries. In addition, the success of message delivery is tracked and reported using the Web service. There is a cost for this service, but you can sign up for a free test drive. If you give them your e-mail address, you'll be able to send 25 messages free of charge. You can check it out by going here: <http://www.strikeiron.com/info/HowToTrial.aspx>. There is also a general information page that you may want to read that's quite useful. You can find it here: <http://www.strikeiron.com/ProductDetail.aspx?p=190>

### Sample Code

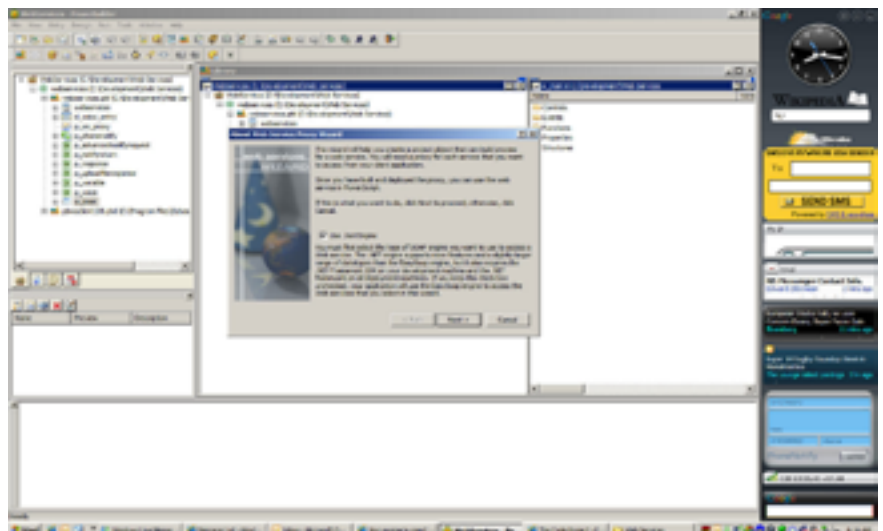
If you look at this code, most of it is also data entry validation. In fact, the workhorse of the function (i.e., the code that actually sends a text



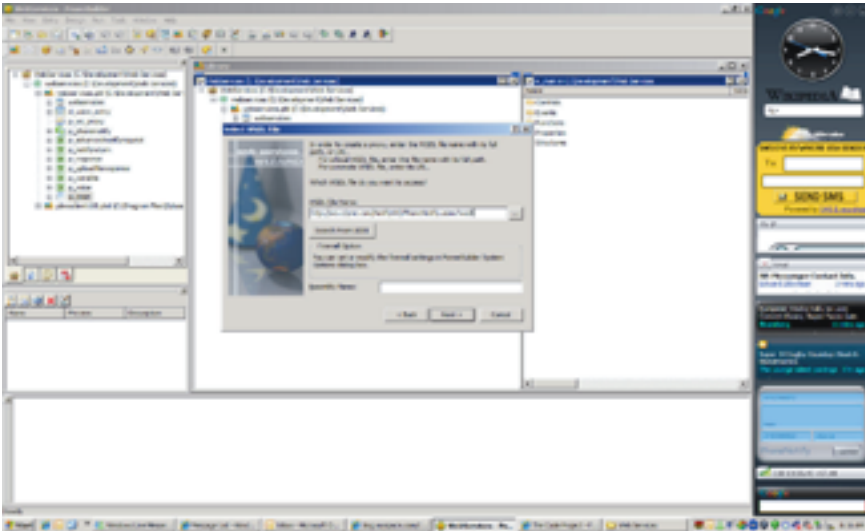
▲ Figure 2



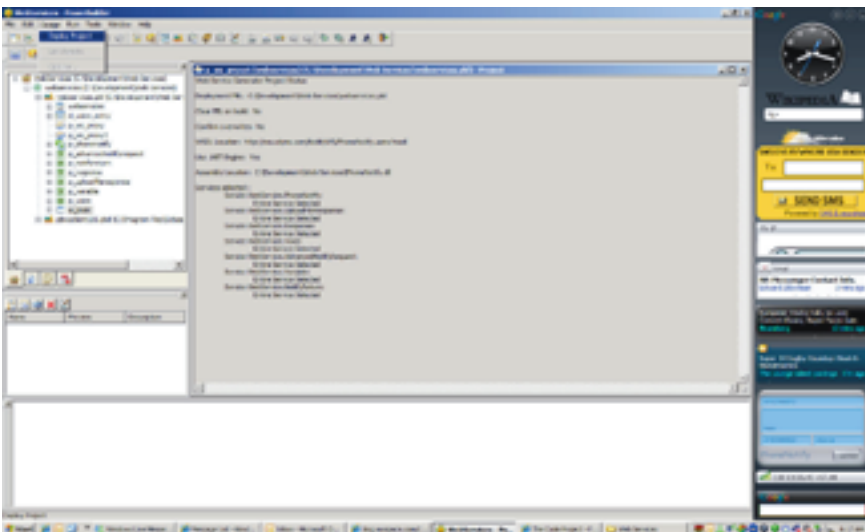
▲ Figure 3



▲ Figure 3



▲ Figure 5



▲ Figure 6

message is simply another one liner:

```
lws_textmsgservice.SendMessage(ls_ToNumber, ls_From-
Number, ls_FromName, ls_Text)
```

Also easy; however, what I have glossed over and will now revisit is the authentication method. In this case, the authentication is sent in the SOAP header (see Listing 2). Why do that? SOAP headers offer a method for passing data to and from an XML Web service method if the data is not directly related to the XML Web service method's primary functionality. For instance, an XML Web service might contain several XML Web service methods, each requiring a custom authentication scheme. Instead of adding parameters to each XML Web service method for the custom authentication scheme via a customer SoapHeader, it can be applied to each XML Web service method. This

is the case with the Strike Iron SMS Web service. [Editor's Note: See the SoapHeader class reference in the References section below for the values to pass for the various attributes that PowerBuilder does not automatically set default values for.]

The question is, when did PowerBuilder start supporting this functionality and how can you use it. The answer is 10.5.1 and the specifics are best answered by reading "what's new" in the help file. Specifically, you would want to look for "Custom header support for .NET Web services." One word of caution though, you must populate every attribute in your set functions structure. If you don't, the set function will fail, the SOAP header will not be sent, and you may spend hours asking yourself why. I know because I learned this the hard way.

### Conclusion

Adding Web services functionality to your PowerBuilder application opens a world of possibilities. There are literally thousands of services out there you can now incorporate into your own PowerBuilder applications. With Sybase's support for the .NET engine and custom SOAP headers, the playing field just got bigger. I encourage you to start exploring.

Happy coding! ■

### Resources

- SOAP Header Class: <http://msdn2.microsoft.com/en-us/library/system.web.services.protocols.soapheader.aspx>
- Custom header support for .NET Web services: See what's new in PowerBuilder 10.5.1 help
- PhoneNotify Web Services Overview: [http://wiki.cdyne.com/index.php/Main\\_Page](http://wiki.cdyne.com/index.php/Main_Page)
- PhoneNotify Service WDSL Location: <http://ws.cdyne.com/NotifyWS/PhoneNotify.asmx?wsdl>
- Strike Iron SMS Service Overview: <http://www.strikeiron.com/ProductDetail.aspx?p=190>
- Strike Iron SMS Service WDSL Location: [http://ws.strikeiron.com/globalsmspro2\\_5?WSDL](http://ws.strikeiron.com/globalsmspro2_5?WSDL)

### About the Author

Chance has been using PowerBuilder since 1999 and PocketBuilder since its inception. Ms. Chance graduated in 1996 with a degree in computer science from the University of Illinois. She has been a frequent contributor to PowerBuilder Developer's Journal and was a key presenter at Sybase TechWave 2005 with A "Real-Time Physical Inventory Solution Using PocketBuilder, ASA and a WiFi Connection." Prior to embarking on a PowerBuilder career, she focused on mobile I.P. by doing real-time embedded programming for the base radio controller group at Motorola as part of the iDEN/Nextel project.  
[chance@crwmail.com](mailto:chance@crwmail.com)