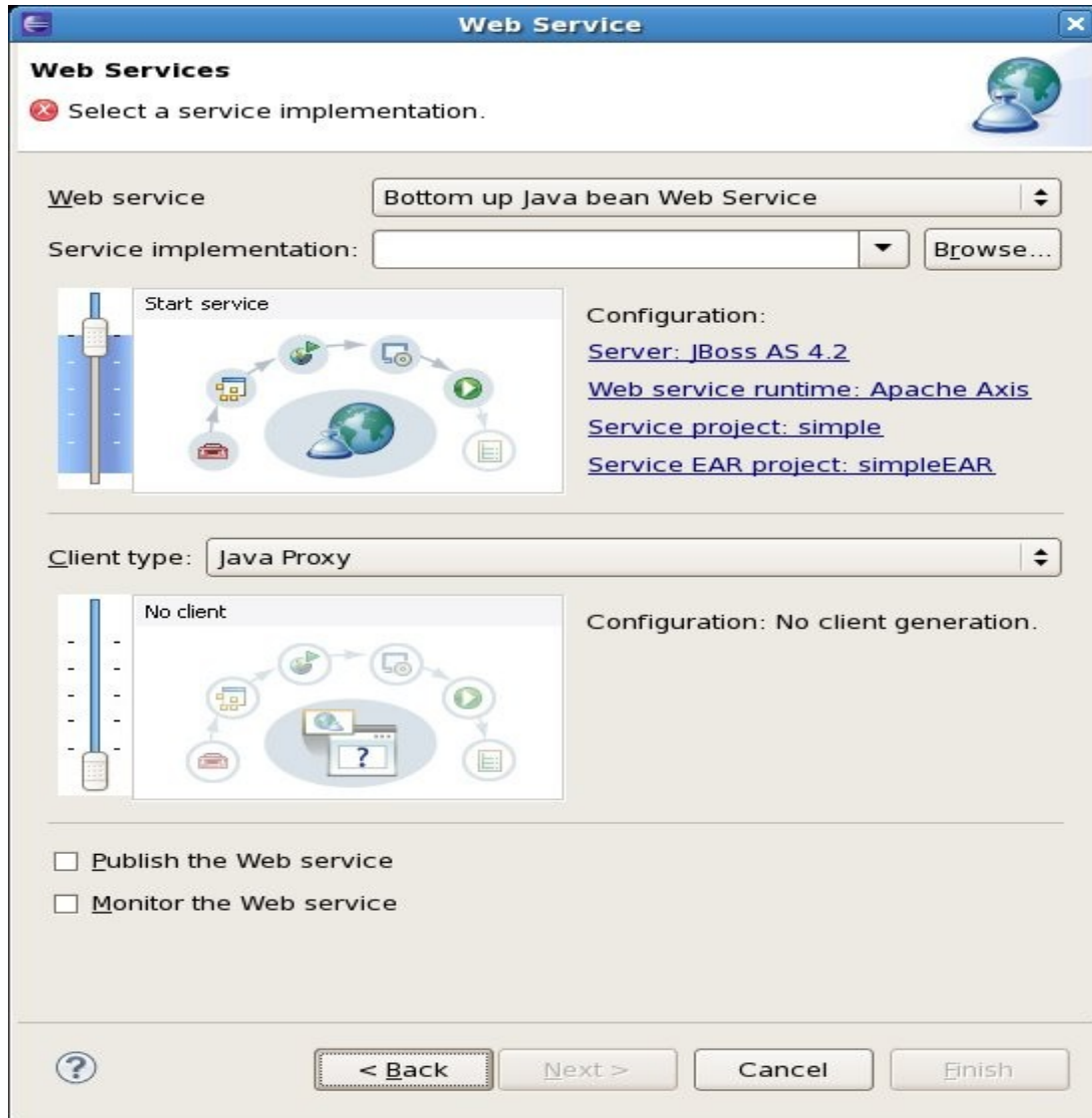


## WTP Web Service Wizard Evaluation

April 26, 2010

Though much of JBoss Tools is based on WTP technologies in one way or another, the Web Service wizard available from WTP is not the most straightforward thing to use or extend.

Let's just look at the first page of the wizard.



The first thing you must do is specify in the "Web service" drop-down what type of web service you are creating - either a Bottom up Java bean Web Service from an existing POJO or a Top Down Java bean Web Service. Is this really necessary? Based on the initial input, we should be able to ascertain which style we need.

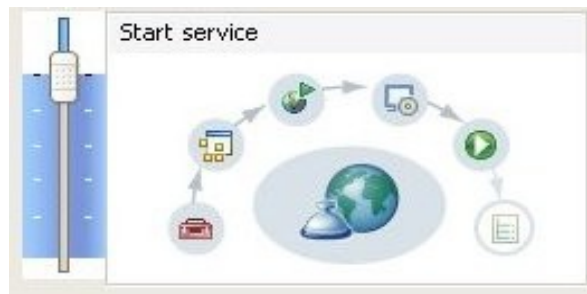


Second, what is a "Service implementation"? And why would this be a drop-down list? Things that appear in the drop-down seem to have been cached from earlier runs through the wizard. Would you ever find yourself needing to create the same service from the same WSDL more than once in a given day? Also on this line in the wizard you have a "Browse..." button, which isn't really all that helpful. If you're browsing for a Java class, the dialog assumes that the class has already been loaded into a project and is in the workspace in a source folder. If you're browsing for a WSDL file, it also assumes that the file has already been loaded into a project in the workspace.



As a developer, I often want to browse elsewhere on my system or on the internet to pull a particular file into a project directly if I haven't already done so. For example, I may have a Java class in an external folder in my file system I want to import directly into my web service project rather than assuming it was there before. Or in the case of a WSDL, what if I've stashed the WSDL file in a folder somewhere or want to import it directly from the internet from a particular URL?

Then there's the matter of the less than straightforward graphic for determining the level at which we're



integrating the service. There are multiple levels with a vertical slider - Develop, Assemble, Deploy, Install, Start, and Test.

Based on the context-sensitive help for the wizard, we see:

“Select the stages of Web services development that you want to complete using the slider:

- Develop: this will create the service code
- Assemble: this ensures that the project that will host the Web service or client gets associated to an EAR when required by the target application server
- Deploy: this will create the deployment code for the service
- Install: this will install the service on the chosen server
- Start: this will start the server once the service has been installed on it
- Test: this will provide various options for testing the service”

Typically we'd want more control of this process from a developer tooling standpoint. I'd like to get rid

of the image/slider entirely and simplify the list of options. What happens in cases, for example, where we don't want the user to create an EAR file? Or if they don't want the whole process to be run and they're just creating the project, which they want to deploy themselves? Though the slider isn't in itself a complicated control to understand, the level of detail here seems out of whack with what the typical user is trying to achieve.

Next is the "Configuration" section, which lumps the server, runtime, service project, and EAR project together as a series of links.

Configuration:

[Server: JBoss AS 4.2](#)

[Web service runtime: Apache Axis](#)

[Service project: simple](#)

[Service EAR project: simpleEAR](#)

I actually really like this approach, which combines a summary of the current settings (and defaults) as well as the ability to change the values by clicking on them. My biggest concern here is the fact that we are required to have an EAR project for some reason, when JBoss runtimes don't really need an EAR and are happy with a simpler WAR packaging. There doesn't seem to be a way to turn off the EAR requirement at this point.

The next bit is the "Client type" area, which allows you to configure a Java client for testing. This area has a similar image slider control and configuration section as the web service area above it, with links to further configuration options as you change the level of client on the slider.



Since there is a separate Web Service Client wizard available, I'm not sure this is truly needed here except as a convenience. It would be nice to be able to turn this whole section off.

Lastly we have the "Publish" and "Monitor" checkboxes.

Publish the Web service

Monitor the Web service

This to me as a new user would be even more confusing in view of the first image slider in the Web

services section above it. If you have the ability to "Deploy" and "Install" in that slider, what does "Publish" do? Well, "Publish" in this context means to publish it to an available UDDI repository. And it isn't immediately clear what monitoring the web service might do either.

Further pages are provided for particular runtime/server combinations, such as those used to configure the name of a generated WSDL file and which operations to expose for a bottom-up approach or where to put the generated code (i.e. package name/source folder) in a particular project for a bottom-up approach.

Developers seeking to contribute to the WTP wizard framework do have a few options to explore. For example, you could extend:

- Web service runtime (such as contributing JBossWS)
- Web service implementation type
- Web service client implementation type
- Service runtime (combination of WS implementation type, WS runtime, and project facets)
- Client runtime
- Environment
- Context
- Initial selection
- Project
- And EAR selection, if available

Little documentation is readily available, but I did find two resources:

- <http://www.eclipse.org/webtools/jst/components/ws/documents/ContributingWebServiceRTWithUI.html>
- <http://www.eclipse.org/webtools/jst/components/ws/documents/ContributingWebServiceRT.html>

Ultimately, though the WTP wizard framework does provide a great deal of functionality - it's simply not granular enough, nor easy enough for many users to wrap their heads around when compared to other tooling options such as MyEclipse, Netbeans, .NET, and so on.

As a result, we should seriously look at either working with WTP to help the existing web service wizard or create our own.