**Billing Test Integrator**

Autor: José Júnior

GVT 14-08-2013

A aplicação “BillingTestIntegrator” foi desenvolvida para facilitar os testes e pedidos feitos para os serviços de Siebel 8 (GetCustomerOrderBillInfo) e Kenan (TransformerService e BillingAdapterService). Com a interface criada, é possível fazer pedidos para qualquer um destes serviços nos seus vários ambientes e de forma independente, ou seja, é possível buscar as informações de uma “customer order” (Siebel 8) no ambiente de desenvolvimento, fazer um pedido ao serviço “transformer” no Kenan de QA e chamar o serviço de “billing adapter” no ambiente de produção (se assim se pretender).

A aplicação foi construída sobre a plataforma Java 7, usando a biblioteca Java Swing. Para a visualização das partes gráficas da aplicação, pode-se usar o plug-in para Eclipse de nome “WindowBuilder” (<http://www.eclipse.org/windowbuilder/>), apesar de não ser aconselhado utiliza-lo no desenho de todas as partes gráficas, já que muita coisa foi feita de forma manual para um melhor desempenho, modularização, manutenção e legibilidade do código criado.

Para iniciar a aplicação, basta chamar o JAR (BillingTestIntegrator.jar) criado na pasta “build”, usando o Java 7 para tal.

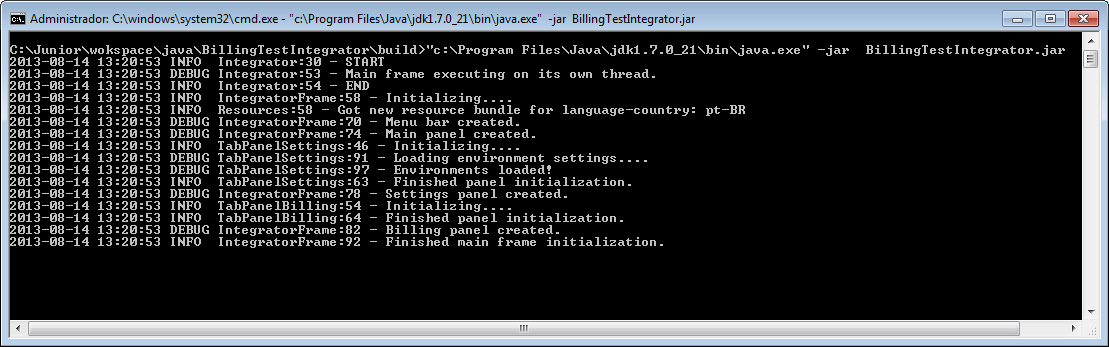


Figura – Iniciar a aplicação.

Caso obtenha o aviso:

**2013-08-14 13:16:07 ERROR Util$FileSystem:90 - Error while trying to read the object from the filesystem: integrator.settings (O sistema não pode encontrar o arquivo especificado)**

**2013-08-14 13:16:07 WARN TabPanelSettings:59 - Settings file could not be loaded! (First time running?)**

Não se preocupe. Isso quer dizer que o ficheiro que guarda as configurações da aplicação não pôde ser lido e será criado da próxima vez que guardar os dados dos ambientes de trabalho (explicado posteriormente).

Ao correr o projeto, a seguinte janela será mostrada:

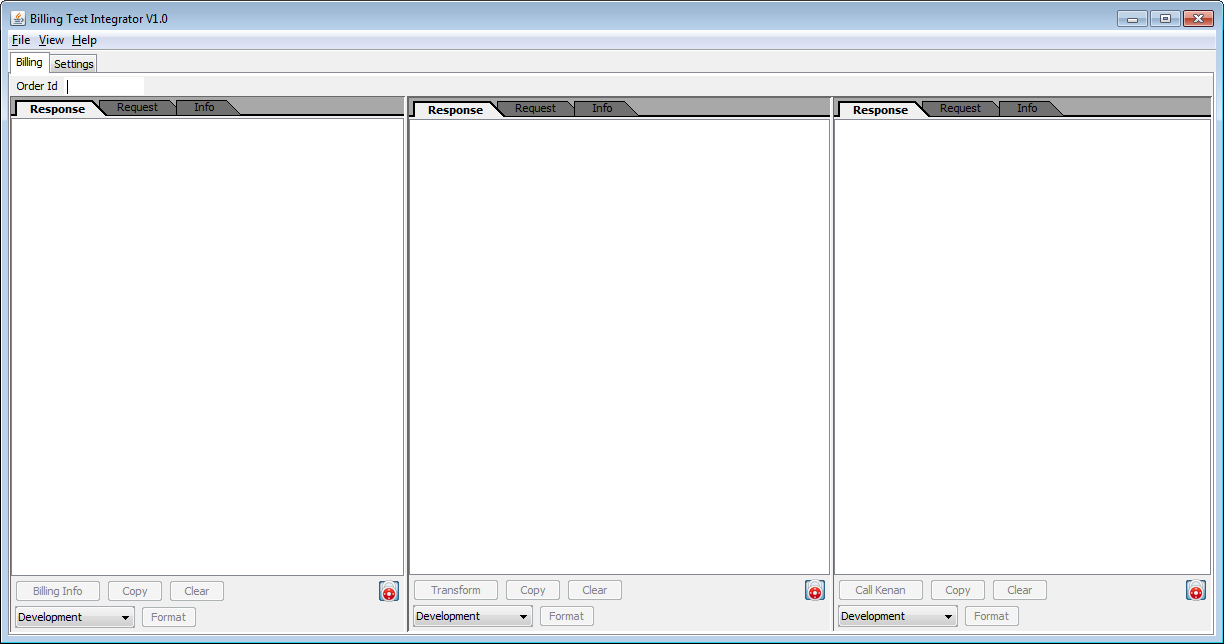


Figura – Janela inicial da aplicação.

A janela mostra uma frame com um “menu bar” ainda muito simples e operações ainda em testes. Por baixo, duas abas: “Billing” e “Settings”.

A janela de “Billing” mostra 3 painéis que servem para (em ordem, da esquerda para a direita): Chamada aos serviços GetCustomerOrderBillInfo (Siebel 8), TransformerService e BillingAdapterService (Kenan).

Em cada painel existem 3 abas. A primeira é a resposta à chamada ao serviço em causa, a segunda é o pedido que foi feito ao serviço e, por fim, a terceira contém informações específicas para cada serviço, que existe apenas para facilitar os utilizadores para não terem que procurar estes dados nos logs. Nesta primeira versão, apenas o serviço “Transformer” contém informações relevantes até o momento.

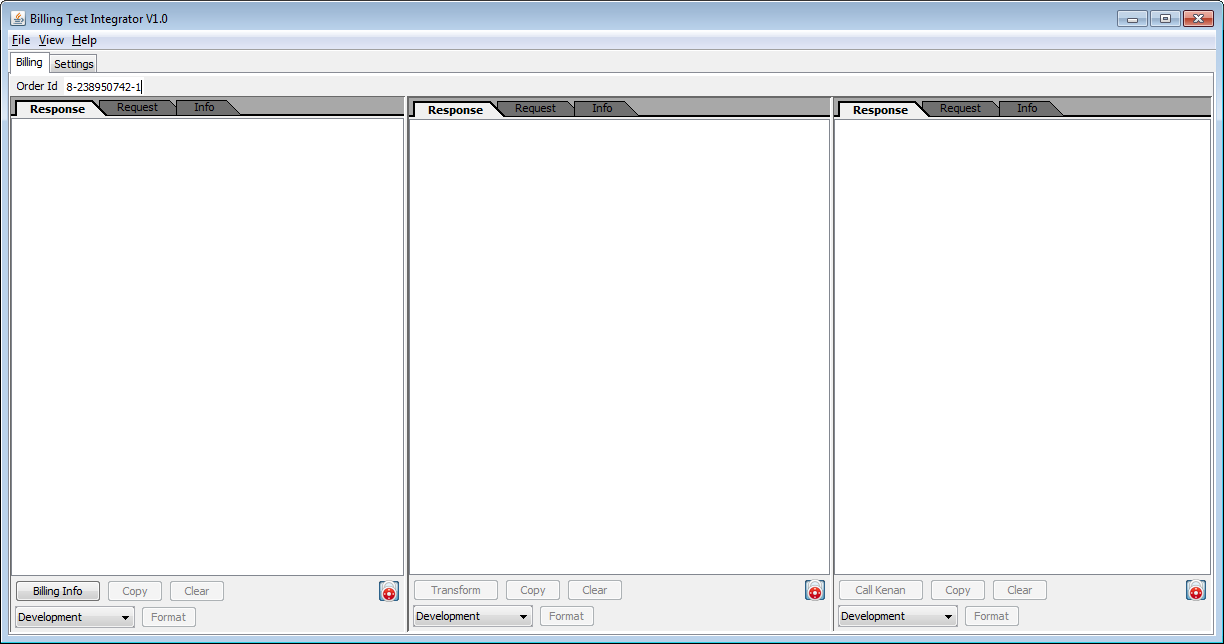


Figura – Inserção de “Order Id”

A inserção do “Order Id” habilita o botão para a pesquisa do mesmo no Siebel 8, através de SOAP request. É possível alterar o ambiente para o qual é feito o pedido através da “combobox” que existe em cada painel. Na figura seguinte, o pedido de informações da ordem do cliente é feita no ambiente de QA:

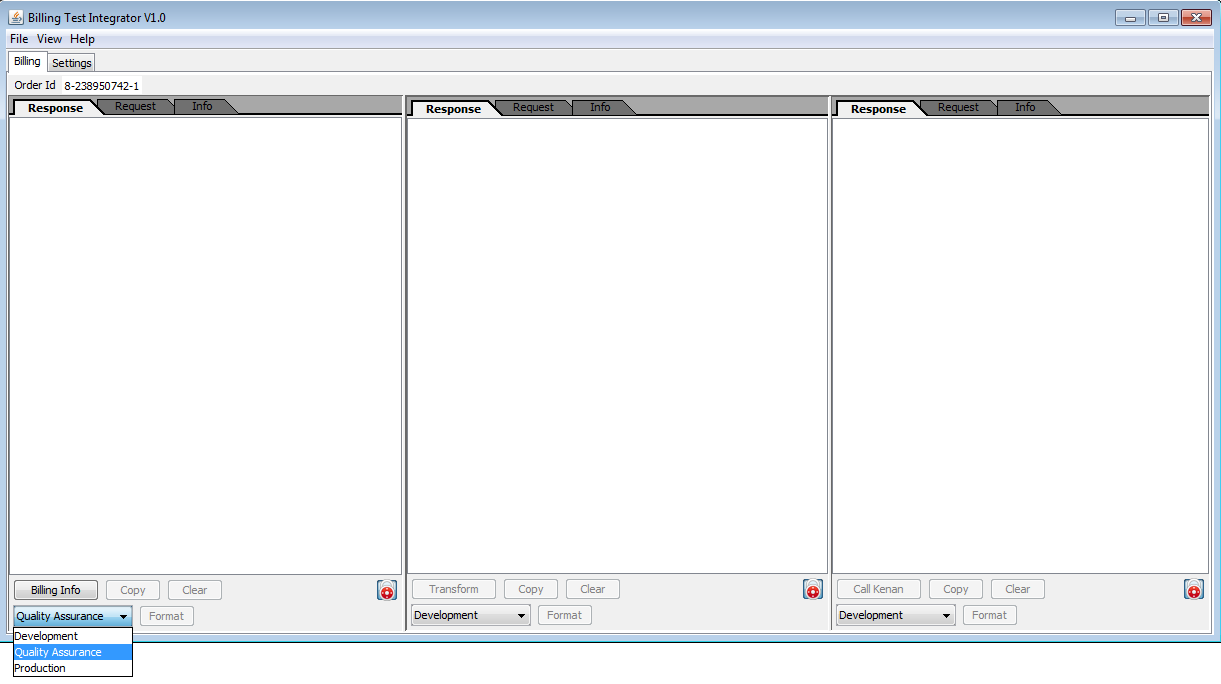


Figura – Alteração do ambiente.

Ao efetuar o pedido, a informação da resposta é mostrada na respectiva aba, e os botões de edição são habilitados para o primeiro quadro, como mostra a figura seguinte:

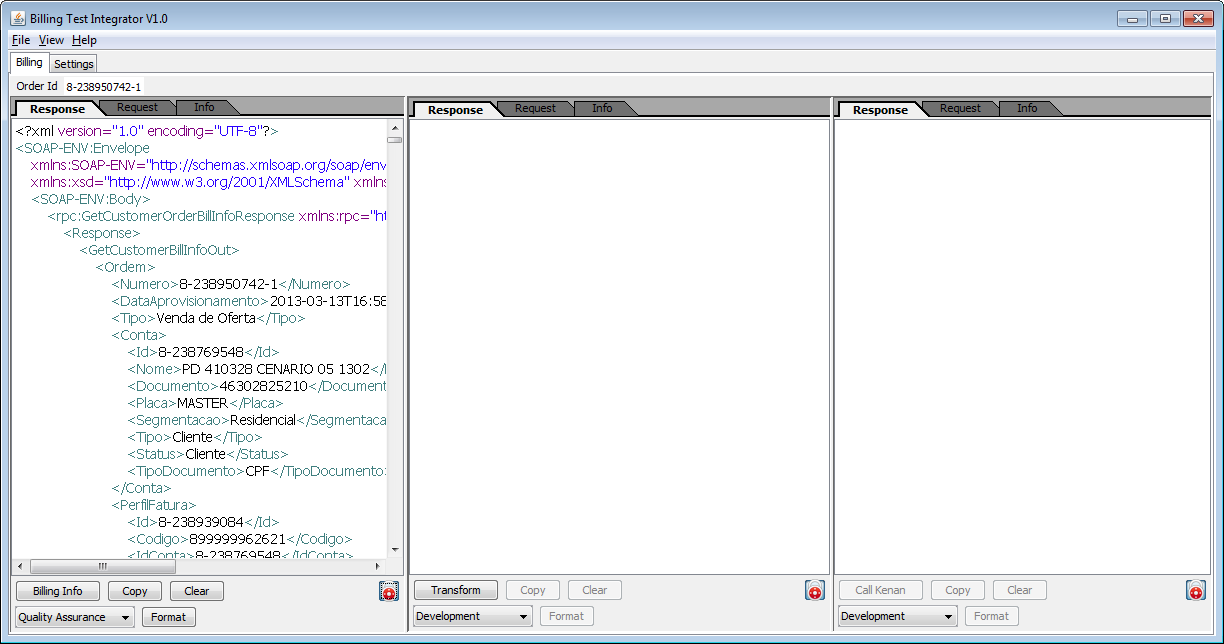


Figura – Resposta do pedido SOAP.

Só podem ser feitas edições nas abas de resposta dos serviços, já que estas serão transformadas nos pedidos subsequentes. Logo, o botão de formatação e o botão de “lock/unlock” (cadeado) só funcionarão para este painel. O botão de cópia funciona para qualquer aba que esteja ativa e o de limpeza limpa os pedidos e respostas para cada painel.

O mesmo acontece para os outros painéis:

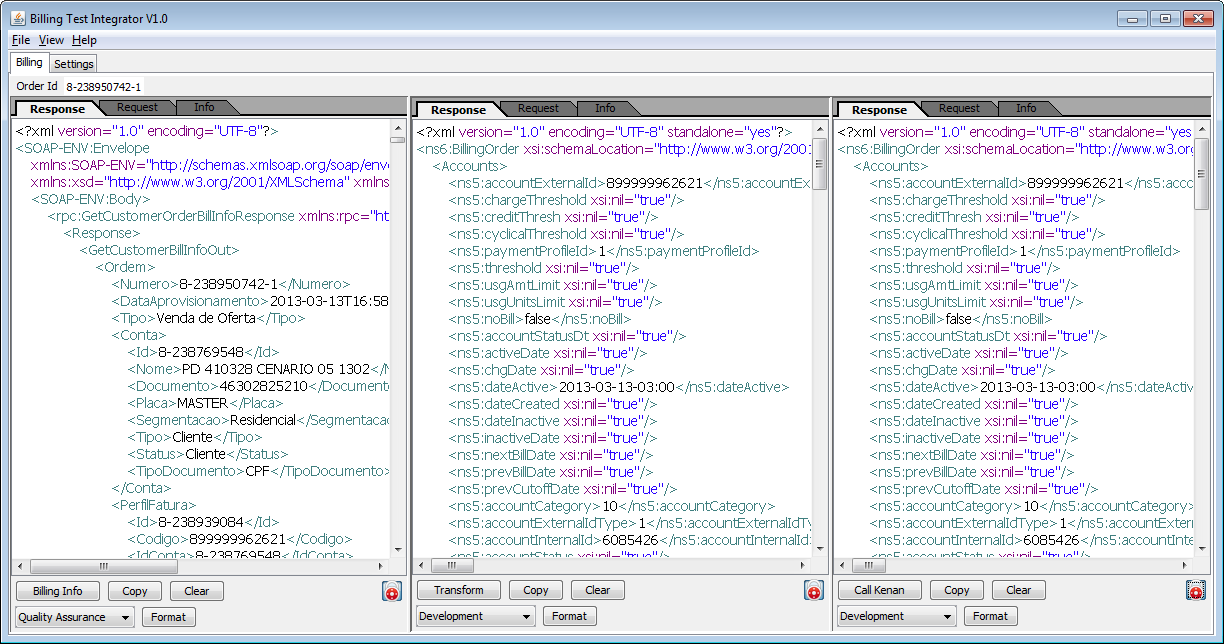


Figura – Demonstração da aplicação com as respostas para todos os pedidos.

No caso do “Transformer”, é possível ver informações específicas relativas à “Billing Order” do cliente.

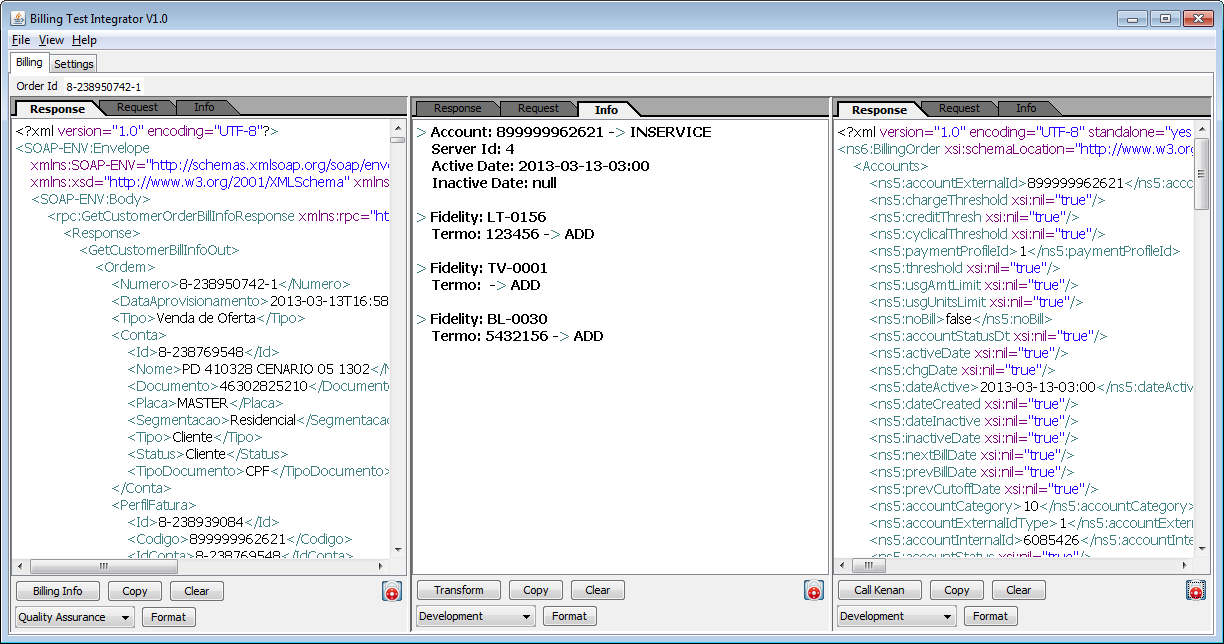


Figura – Detalhes da “Billing Order” após chamada ao serviço “Transformer”.

Na aba “Settings” é possível configurar os URL para os serviços mencionados em cada um dos ambientes.

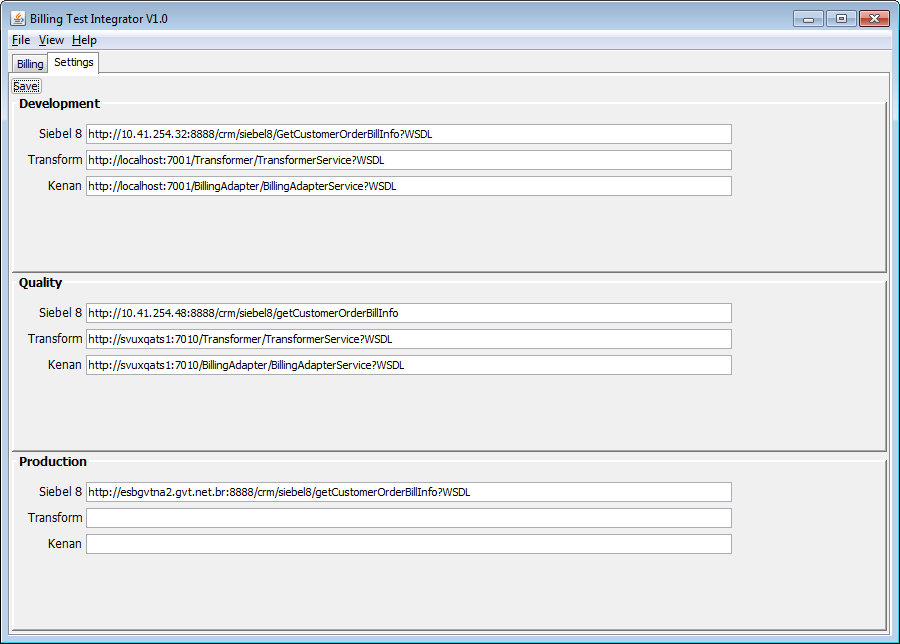


Figura – Configurações dos URL dos serviços nos respectivos ambientes.

Ao inserir ou alterar algum ambiente, é imponte guardar as configurações com o botão “Save” deste mesmo painel. Isso irá gerar o ficheiro de configurações que será lido da próxima vez que se iniciar a aplicação.

Ao correr a aplicação é criada a pasta “log” com os as informações dos acontecimentos mais importantes para que, no caso de erro, seja possível compreender o que pode ter corrido mal.

Libs usadas neste projeto:

* glassfish.jaxb\_1.0.0.0\_2-1-12.jar
* glassfish.jaxws.rt\_1.1.0.0\_2-1-5.jar
* log4j-1.2.13.jar
* stax-ex-1.2.jar
* streambuffer-0.8.jar
* stubs-stubs-1.1.jar
* xercesImpl-2.8.1.jar
* xml-text-editor-0.0.3.jar

Na raiz do projeto existe um “build.xml” para se correr com o Ant que faz toda a limpeza e build/rebuild do projeto na pasta build.

Dúvidas, sugestões e contribuições monetárias: [jose.junior16@gvt.com.br](mailto:jose.junior16@gvt.com.br)

Erros e soluções de problemas

Se obtiver o seguinte erro:

br.com.gvt.stubs.Component$JaxbAccessorF\_accountExternalId cannot be cast to com.sun.xml.bind.v2.runtime.reflect.Accessor

Isso quer dizer que existe um conflito do JAXB, explicado no seguinte parágrafo:

“The crux of this issue is there is a change in the JAXB **API**, the runtime implementation you are attempting to use does not match the version of the JAXB API bundled with the JDK.

In order to use a different version, you should copy coresponding versions of [jaxb-api.jar](http://www.java2s.com/Code/JarDownload/jaxb-api/jaxb-api-2.2.3.jar.zip) and [jaxws-api.jar](http://www.java2s.com/Code/JarDownload/jaxws/jaxws-api-2.2.1.jar.zip) into an endorsed lib (e.g. %JAVA\_HOME%\lib\endorsed).

A complete list of options is given in section [7.1.2 of the Unofficial JAXB Guide](http://jaxb.java.net/guide/Migrating_JAXB_2_0_applications_to_JavaSE_6.html#Using_JAXB_2_1_or_JAXB_2_2_with_JavaSE_6)

It is a mistake to copy the implementation jars (e.g. jaxb-impl.jar) into endorsed lib, these should simply be on your classpath.

Also note that you can get into trouble if you attempt to use a newer version of jaxb without also including a compatible version of jaxws. This is because the old jaxws attempts to reference the old jaxb, so if you're changing one make sure you do both. (Stack-trace in package com.sun.xml.internal.wsimplicates an old jax-ws implementation. Even the latest release of Java still ships with old version 1 jaxb and jaxws apis).”

Fonte: <http://stackoverflow.com/questions/14162159/supplying-a-different-version-of-jaxb-for-jax-ws-in-java-1-6>