

Idiosyncrasies

Table of contents

1 IBM Rational Software Development Platform Trial 6.0.0 (RSA)/UML Profile for Software Services, RSA Plug-In on Linux 2.6.12-1.1381_FC3.....	2
2 Eclipse 3.0.2/Ant 1.6/Axis 1.2RC3 on Linux 2.6.10-1.770_FC3.....	2
3 Debugging a process in BPWS4J.....	3
4 Calling an ANT target repeatedly or parameterising a call to a target.....	3
5 Setting your classpath in ANT.....	5
6 BPWS4J - One BPEL instance invoking another.....	5
7 Fedora Core - No sound when playing flash animations.....	8
8 ActiveBPEL - BPEL to BPEL invokation.....	8
9 ActiveBPEL - BPEL to BPEL invokation (2).....	10
10 Xerces 2.6.2 & Xalan Issue 2.6 - Missing org.w3c.dom.xpath interfaces.....	11
11 Blazix 1.2.5 & Java 1.5 incompatibility.....	11

During the course of programming I often come across idiosyncrasies of tools/frameworks/languages or even worse combinations of these! This page is a list of some of the things i've noted lately. There in no particular order, there just here to help out people if they had the same issues.

1. IBM Rational Software Development Platform Trial 6.0.0 (RSA)/UML Profile for Software Services, RSA Plug-In on Linux 2.6.12-1.1381_FC3

When attempting to install the UML Profile for Software Services in IBM RSA you may get an error like below. This error is actually related to a lack of permissions. You must be root or have installed the program to a writable directory to install plugins.

```

        java.io.IOException: Cannot allocate memory
        at java.lang.UNIXProcess.forkAndExec(Native Method)
        at java.lang.UNIXProcess.<init>(UNIXProcess.java:157)
        at java.lang.ProcHelper.run(ProcHelper.java:60)
csite null
csite null
csite null
csite null
csite null

```

To avoid this error start the program as root, install the plugin and then restart the program as a normal user.

2. Eclipse 3.0.2/Ant 1.6/Axis 1.2RC3 on Linux 2.6.10-1.770_FC3

When running an ant script from eclipse an error such as this may occur.

```

[ java] org.eclipse.ant.internal.ui.antsupport.AntSecurityException
[ java]         at
org.eclipse.ant.internal.ui.antsupport.AntSecurityManager.checkExit(AntSecurityManager.
[ java]         at java.lang.Runtime.exit(Runtime.java:88)
[ java]         at java.lang.System.exit(System.java:868)
[ java]         at
org.apache.axis.wsdl.Java2WSDL.main(Java2WSDL.java:617)
[ java]         at sun.reflect.NativeMethodAccessorImpl.invoke0(Native
Method)
[ java]         at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
[ java]         at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
[ java]         at java.lang.reflect.Method.invoke(Method.java:585)
[ java]         at
org.apache.tools.ant.taskdefs.ExecuteJava.run(ExecuteJava.java:193)

```

```
[ java]          at
org.apache.tools.ant.taskdefs.ExecuteJava.execute(ExecuteJava.java:130)
[ java]          at
org.apache.tools.ant.taskdefs.Java.run(Java.java:705)
[ java]          at
org.apache.tools.ant.taskdefs.Java.executeJava(Java.java:177)
[ java]          at
org.apache.tools.ant.taskdefs.Java.execute(Java.java:83)
[ java]          at
org.apache.tools.ant.UnknownElement.execute(UnknownElement.java:275)
[ java]          at org.apache.tools.ant.Task.perform(Task.java:364)
[ java]          at org.apache.tools.ant.Target.execute(Target.java:341)
[ java]          at
org.apache.tools.ant.Target.performTasks(Target.java:369)
[ java]          at
org.apache.tools.ant.Project.executeTarget(Project.java:1214)
[ java]          at
org.eclipse.ant.internal.ui.antsupport.InternalAntRunner.run(InternalAntRunner.java:385)
[ java]          at
org.eclipse.ant.internal.ui.antsupport.InternalAntRunner.main(InternalAntRunner.java:13
```

To avoid this error simply set fork="true" for any targets that cause such an error. For example..

```
<java classname="org.apache.axis.wsdl.Java2WSDL" fork="true"/>
```

3. Debugging a process in BPWS4J

Debugging a composition in [IBM's WS-BPEL](#) process engine is pretty tough. It's made a lot easier by turning on the log4j logging utility. Here is how to do it, you might have missed the step as it's hidden a bit in the documentation.

- 1. Edit the log4j.properties file in the \$TOMCAT_HOME/webapps/bpws4j/WEB-INF/classes directory
- 2. Uncomment or add log4j.rootLogger=DEBUG, fileout
- 3. Uncomment or add log4j.logger.bpws.runtime=ALL
- 4. Save and close the log4j.properties file
- 5. Lookout for a file called bpws4j.out in \$TOMCAT_HOME/bin

4. Calling an ANT target repeatedly or parameterising a call to a target

Sometimes you will want to call a target numerous times. You can achieve this in ANT by using the antcall task. In this example I am undeploying a few AXIS web services. Note how the antcalls are parameterised using param and retrieved using \${}. I've used a few ANT properties as well which I think are rather sharp ;)

The code below is a snippet to show how its done.

```

    <!-- set some values for tomcat the Web Service server -->
    <property name="tomcat.server" value="localhost"/>
    <property name="tomcat.port" value="8080"/>
    <property name="tomcat.url"
value="http://${tomcat.server}:${tomcat.port}/manager/" />
    <property name="tomcat.username" value="username_value"/>
    <property name="tomcat.password" value="password_value"/>
    <property name="tomcat.application" value="/axis"/> <!--the prefix
slash if needed! -->

    <!-- Some custom properties-->
    <property name="axisAdminURI"
value="http://${tomcat.server}:${tomcat.port}/axis/services/AdminService"/>

    <target name="undeploy_service" description="UnDeploy a Web
Services in AXIS">
        <java classname="org.apache.axis.client.AdminClient"
fork="true">
            <classpath refid="base.path"/>
            <arg value="-l${axisAdminURI}"/>
            <arg value="${wsdd.param}"/>
        </java>
        <echo>undeploy ran</echo>
    </target>

    <target name="all" description="clean, build and run">
        <description>
            This target does all the work of calling all the
sub-targets.
        </description>
        <!-- Undeploy all the Web Service's whose wsdd files we
have-->
        <echo>We will now call the "undeploy_service" target for
each *undeploy.wsdd file</echo>
        <antcall target="undeploy_service">
            <param name="name.param" value="CoreBanking"/>
            <param name="wsdd.param"
value="${conf}/CoreBanking_undeploy.wsdd"/>
        </antcall>
        <antcall target="undeploy_service">
            <param name="name.param" value="RiskManagement"/>
            <param name="wsdd.param"
value="${conf}/RiskManagement_undeploy.wsdd"/>
        </antcall>
        <antcall target="undeploy_service">
            <param name="name.param" value="CreditCard"/>
            <param name="wsdd.param"
value="${conf}/CreditCard_undeploy.wsdd"/>
        </antcall>
    </target>

```

```

        <!-- Setup the project structure -->
        <antcall target="init"/>
        ....etc
    </target>

```

5. Setting your classpath in ANT

Setting up your Java classpath is always a pain when you first start using Java or ANT so here's how you do it with ANT. By defining the classpath as a path you can reuse it again and again within an ANT build! The code snippet below is for compiling some AXIS Web Services.

```

<!-- Set up some properties -->
<property name="src" location="src"/>
<property name="build" location="build"/>
<property name="build.classes" location="build/classes"/>

<!-- Set up the classpath -->
<path id="base.path">
    <pathelement path="{build}" />
    <pathelement path="{build.classes}" />
    <fileset dir="/home/ronan/axis-1_2RC3/lib">
        <include name="**/*.jar"/><!--All the AXIS jars-->
    </fileset>
    <fileset dir="/home/ronan/javamail-1.3.2">
        <include name="**/*.jar"/><!--All the Javamail
jars-->
    </fileset>
    <fileset dir="/home/ronan/jaf-1.0.2">
        <include name="**/*.jar"/><!--All the Activation
jars-->
    </fileset>
    <fileset dir="/home/ronan/jakarta-tomcat-5.5.8/server/lib">
        <include name="**/catalina-ant.jar"/><!--All the
Tomcat ANT jars-->
    </fileset>
</path>

<!--Compile something with our classpath-->
<javac srcdir="{src}" destdir="{build.classes}" fork="true">
    <classpath refid="base.path"/>
    <include name="**/*.java"/>
</javac>

```

6. BPWS4J - One BPEL instance invoking another

Getting the BPWS4J process engine to invoke another BPEL process from within a running BPEL process can be a real pain to figure out due to the poor or non-existent documentation.

However after much pain and effort I have got this working. The following is what you need to do.

- 1. Write the BPEL and WSDL for the outermost workflow, we will call this Workflow #1
- 2. Write the BPEL and WSDL for the inner workflow (the one which the outermost workflow will call), we will call this Workflow #2
- 3. Edit the WSDL of Workflow #2 by adding in a WSDL binding section and associated service port binding section. You should note that normally a Workflow exposed by BPEL4WS does not need these sections as a coded client will call BPWS4J directly. However as we need to call the engine again we need to tell Workflow #1 how to do this by giving it a binding and an address to find Workflow #2. This is achieved within the WSDL of Workflow #2. note: The binding namespaces of the input/output sections must match the "Method Namespace URI's of the deployed BPEL services.
- 4. Save and deploy the two Workflows separately.
- 5. Invoke Workflow #1 as usual and be impressed ;)

Below is the WSDL for Workflow #2 with the alterations made to make it identifiable to Workflow #1

```
<?xml version="1.0"?>
<definitions name="RiskManagementP2PService"
  targetNamespace="http://acme.com/wsd1/RiskManagementP2P"
  xmlns:plt="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.xmlsoap.org/wsd1/"
  xmlns:RiskManagementP2P="http://acme.com/wsd1/RiskManagementP2P"
  xmlns:CoreBankingP2P="http://acme.com/wsd1/CoreBankingP2P"
    xmlns:RiskManagement="http://RiskManagement"
    xmlns:CreditCard="http://CreditCard"
  xmlns:wsd1="http://schemas.xmlsoap.org/wsd1/"
  xmlns:wsd1soap="http://schemas.xmlsoap.org/wsd1/soap/">

  <message name="ServiceRequestType">
    <part name="accountName" type="xsd:string"/>
  </message>
  <message name="ServiceResponseType">
    <part name="getRiskAssessmentReturn" type="xsd:boolean"/>
  </message>

  <portType name="RiskManagementP2PPortType">
    <operation name="RiskManagementP2P">
      <input
message="RiskManagementP2P:ServiceRequestType" name="ServiceRequestType"/>
      <output
```

```
message="RiskManagementP2P:ServiceResponseType"
name="ServiceResponseType" />
  </operation>
</portType>

  <binding name="RiskManagementP2PSoapBinding"
type="RiskManagementP2P:RiskManagementP2PPortType">
  <wsdlsoap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http" />
  <operation name="RiskManagementP2P">
    <wsdlsoap:operation soapAction="" />
    <input name="ServiceRequestType">
      <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://acme.com/wsdl/RiskManagementP2P#RiskManagementP2PService#CoreBankingP
#http://acme.com/wsdl/RiskManagementP2P#RiskManagementP2PPortType"
use="encoded" />
      </input>
      <output name="ServiceResponseType">
        <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://acme.com/wsdl/RiskManagementP2P#RiskManagementP2PService#CoreBankingP
#http://acme.com/wsdl/RiskManagementP2P#RiskManagementP2PPortType"
use="encoded" />
        </output>
      </operation>
    </binding>

    <service name="RiskManagementP2PService">
      <port
binding="RiskManagementP2P:RiskManagementP2PSoapBinding"
name="RiskManagementP2PPortType">
        <wsdlsoap:address
location="http://localhost:8080/bpws4j/services/" />
      </port>
    </service>

    <plt:partnerLinkType name="RiskManagementP2PPLT">
      <plt:role name="RiskManagementP2P">
        <plt:portType
name="RiskManagementP2P:RiskManagementP2PPortType" />
      </plt:role>
    </plt:partnerLinkType>

    <plt:partnerLinkType name="CoreBankingP2PPLT">
      <plt:role name="CoreBankingP2P">
        <plt:portType
name="CoreBankingP2P:CoreBankingP2PPortType" />
      </plt:role>
    </plt:partnerLinkType>

    <plt:partnerLinkType name="RiskManagementPLT">
      <plt:role name="RiskManagement">
        <plt:portType
```

```

name="RiskManagement:RiskManagementPortType"/>
    </plt:role>
  </plt:partnerLinkType>

  <plt:partnerLinkType name="CreditCardPLT">
    <plt:role name="CreditCard">
      <plt:portType
name="CreditCard:CreditCardPortType"/>
    </plt:role>
  </plt:partnerLinkType>

</definitions>

```

7. Fedora Core - No sound when playing flash animations

This is a strange error i've had since FC1 (i'm using FC3 now). Whenever a flash file is run inside firefox it will not play the audio. After much reading into this issue it seems the [ALSA](#) (Advanced Linux Sound Architecture) is the root of the problem. To fix the problem do the following.

- 1. Ensure your yum.conf is set like [this](#).
- 2. As root run yum install alsa-tools.i386
- 3. Restart firefox.
- 4. Try out a flash audio file.

8. ActiveBPEL - BPEL to BPEL invokation

ActiveBPEL allows you to invoke another BPEL process from within a running BPEL process. However there is something which may catch you out. The partnerLink definitions in the .pdd file you must specify the PortName of the BPEL process to be invoked. However this is not the portType as I initially assumed, instead it is the name of the wsdl:port specified in the WSDL of the deployed BPEL process. Below I clarify this some more.

Here is a section of the BPEL process WSDL file. Note RiskManagementP2PServicePort is the on the wsdl:service.

```

    <wsdl:service name="RiskManagementP2PService">
      <wsdl:port
binding="RiskManagementP2P:RiskManagementP2PServiceBinding"
name="RiskManagementP2PServicePort">
        <soap:address
location="http://localhost:8080/active-bpel/services/RiskManagementP2PService"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
      </wsdl:port>

```

```
</wsdl:service>
```

Here is a section of the .pdd file. Note RiskManagementP2PServicePort is the PortName.

```
    <partnerRole endpointReference="static">
      <wsa:EndpointReference
xmlns="http://schemas.xmlsoap.org/ws/2003/03/addressing"
xmlns:RiskManagementP2P="http://acme.com/wsdl/RiskManagementP2P">
<wsa:Address>http://localhost:8080/active-bpel/services/RiskManagementP2PService</wsa:A
      <wsa:ServiceName
PortName="RiskManagementP2PServicePort">RiskManagementP2P:RiskManagementP2PService</wsa:
      </wsa:EndpointReference>
    </partnerRole>
```

If these two names do not match you will get the following obscure exception.

```
AxisFault
  faultCode:
  {http://www.active-endpoints.com/2004/06/bpel/extensions/}systemError
  faultSubcode:
  faultString: Error invoking operation.
  faultActor:
  faultNode:
  faultDetail:
    {}string: java.lang.NullPointerException
    at
org.activebpel.rt.axis.bpel.AeInvokeHandler.handleInvoke(AeInvokeHandler.java:125)
    at
org.activebpel.rt.bpel.server.engine.AeInMemoryQueueManager$AeInvokeWork.run(AeInMemory
    at org.activebpel.work.AeWorkerThread.run(AeWorkerThread.java:154)

Error invoking operation.
    at
org.apache.axis.message.SOAPFaultBuilder.createFault(SOAPFaultBuilder.java:221)
    at
org.apache.axis.message.SOAPFaultBuilder.endElement(SOAPFaultBuilder.java:128)
    at
org.apache.axis.encoding.DeserializationContext.endElement(DeserializationContext.java:
    at org.apache.xerces.parsers.AbstractSAXParser.endElement(Unknown
Source)
    at
org.apache.xerces.impl.XMLNSDocumentScannerImpl.scanEndElement(Unknown
Source)
etc.....
```

9. ActiveBPEL - BPEL to BPEL invokation (2)

If you wish to invoke a BPEL process from another BPEL process you must include the binding and related service elements in the WSDL for the BPEL process. Normally this is not necessary when publishing a BPEL process as a client will directly invoke it, this is not the case for BPEL to BPEL invokations though

A common error indicating you have not including the binding section in the WSDL is as follows..

```
06/24/2005 06:14:19:63 [risk_management.bpr] RiskManagementP2P.pdd
No port found for service
http://acme.com/wsdl/CreditCardP2P:CreditCardP2PService.
```

To correct the error look at the published WSDL for the target BPEL process e.g. <http://localhost:8080/active-bpel/services/CreditCardP2PService?WSDL> and cut and paste the `wSDL:binding` and `wSDL:service` sections into the locally stored WSDL which also declares your partner link types.

The import bits to copy and paste should look like below.

```
<wSDL:binding name="CreditCardP2PServiceBinding"
type="CreditCardP2P:CreditCardP2PPortType">
    <soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
    <wSDL:operation name="CreditCardP2P">
        <soap:operation soapAction="" style="rpc"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
        <wSDL:input>
            <soap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
        </wSDL:input>
    </wSDL:operation>
</wSDL:binding>
```

```
<wsdl:service name="CreditCardP2PService">
  <wsdl:port binding="CreditCardP2P:CreditCardP2PServiceBinding"
name="CreditCardP2PServicePort">
    <soap:address
location="http://localhost:8080/active-bpel/services/CreditCardP2PService"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
  </wsdl:port>
</wsdl:service>
```

10. Xerces 2.6.2 & Xalan Issue 2.6 - Missing org.w3c.dom.xpath interfaces

I haven't used Xalan/Xerces in a while so I downloaded the latest versions. However Xerces 2.6.2 & Xalan Issue 2.6 combined do not have an implementation of some important XPATH interfaces. This apparenly is caused by a reordering of where the xpath interfaces should live.

After considerable reading up on this topic I discovered the newer 2.7 version of Xerces has the org.w3c.dom.xpath interfaces so downloading and placing it on your classpath will solve any problems.

The following imports (and others from the same tree) will cause a classpath error.

```
import org.w3c.dom.xpath.XPathEvaluator;
import org.w3c.dom.xpath.XPathNSResolver;
import org.w3c.dom.xpath.XPathResult;
```

11. Blazix 1.2.5 & Java 1.5 incompatibility

When running "The Blazix EJB Compiler", java desisoft.ejbtools.EjbCompiler or blxejbc: you may encounter to the following error. This is a very common error.

```
java.io.FileNotFoundException:
ejbtemp/stocks/StockQuotesBeanCtx_Skel.class (No such file or directory)
at java.io.FileInputStream.open(Native Method)
at java.io.FileInputStream.<init>(FileInputStream.java:106)
at desisoft.ejbtools.EjbCompiler.addZipEntry(EjbCompiler.java:79)
at
desisoft.ejbtools.EjbCompiler.compileAndAddToJar(EjbCompiler.java:1338)
at desisoft.ejbtools.EjbCompiler.compileBeans(EjbCompiler.java:449)
at desisoft.ejbtools.EjbCompiler.compile(EjbCompiler.java:224)
at desisoft.ejbtools.EjbCompiler.main(EjbCompiler.java:1513)
```

This error is caused by an incompatibility between blazix and Java 1.5 To avoid this error use Java 1.4 with blazix or use the following script with Java 1.5

The script creates the RMI skeletons not produced by default on Java 1.5 installation. It's written for Bash/Linux but should be easy enough to alter for Dos/Windows.

```
#####
# Abstract: This script overcomes a bug in the autogenerated build.cmd
#
#         created by Blazix 1.2.5 This error seems to be caused by
#
#         Java 1.5+ not creating skeltons for RMI communication.      #
# Alternatives : Use the Java 1.4 SDK to avoid this problem. Of course this
#
#         is not always possible thus this script.                  #
# Usage : Use this script instead of the autogenerated build.cmd    #
# Platform : Linux/Bash
#
# Author: Ronan Barrett
#
# Date: 27/09/05                                                    #
#####

#Set these variables to determine how the script parameters - TODO:Might
want to make these comamnd parameters
package_name="stocks"
ejb_name="StockQuotes"
temp_dir="ejbtemp"
home_dir=`pwd`/" #store the current directory and append a trailing slash

#compile the Home, the Bean and the Interface
javac -d "$home_dir" "$home_dir$ejb_name"Home.java
"$home_dir$ejb_name".java "$home_dir$ejb_name"Bean.java

#roll up all the compiled files to a jar
jar -cvf "$ejb_name".jar -C "$home_dir" META-INF/ejb-jar.xml
"$package_name"/"$ejb_name"Home.class "$package_name"/"$ejb_name".class
"$package_name"/"$ejb_name"Bean.class

#prepare a temporary directory
rm -ifr "$temp_dir" > /dev/null
mkdir "$temp_dir"

#perform the Blazix specific EJB work - This causes an error which we
ignore
echo "***** Ignore errors *****"
java desisoft.ejbtools.EjbCompiler -keepgenerated "$ejb_name".jar
"$ejb_name"Ejb.jar
echo "***** Stop ignoring *****"
```

```
#compile the source files created by desisoft.ejbtools.EjbCompiler
cd "$temp_dir"
javac -classpath "$CLASSPATH":../../"$ejb_name".jar *.java

#clean up and move the compiled EjbCompiler files so we can generate some
Skeletons
rm -ifr "$package_name" > /dev/null
mkdir "$package_name"
mv *.class "$package_name"

#Java 1.5 doesn't be default create Skeletons so we force RMIC todo it for
us
rmic -v1.1 -classpath $CLASSPATH:../../"$ejb_name".jar
"$package_name"."$ejb_name"HomeCtx
rmic -v1.1 -classpath $CLASSPATH:../../"$ejb_name".jar
"$package_name"."$ejb_name"BeanCtx
cd ..

#finally roll up all the required plumbing into a jar file
jar cf "$ejb_name"Ejb.jar "$package_name"/*.class META-INF/*

#finally, finally, clean up
rm -ifr "$temp_dir" > /dev/null

#you might want to try out a simple client using your new jar, try the
commands below
#javac -classpath StockQuotes.jar:../blazix.jar SessionClientSample.java
#java SessionClientSample
```