

**INTEGRATION OF SPEEDO
INTO J2EE ENVIRONMENT**

Stéphane Rodiere
Sebastien Chassande-Barrioz

SOMMAIRE

1.INTRODUCTION.....	3
1.1.Objective.....	3
2.DEPLOYMENT INTO APPLICATIONS SERVERS.....	4
2.1.Introduction.....	4
2.2.Common configuration	4
2.2.1.Prerequisite.....	4
2.2.2.Configuration.....	4
2.2.2.1.Deployment.....	4
2.2.2.2.build.properties configuration.....	4
1.1.1.1.1.Configure J2ee.jar and tools.jar classloader.....	4
1.1.1.1.2.Access to external resources	4
1.1.1.1.3.Define Database librairie in order to compile.....	4
1.1.1.1.4.Define Database name.....	4
1.1.1.1.5.Application Server.....	5
1.1.1.1.6.Application Name to deploy.....	5
1.1.1.1.7.Application directory name.....	5
1.1.1.1.8.JMeter directory.....	5
2.2.2.3.Resource connector.....	5
2.3.Weblogic 8.1 Deployment.....	5
2.3.1.Introduction.....	5
2.3.2.Ant tasks.....	5
2.3.3.speedo.properties file.....	6
2.3.4.Specific JDBC deployment.....	6
2.3.5.JCA connector.....	6
2.3.6.J2eedo Deployment.....	7
2.3.7.Http Execution.....	7
2.3.8.Recordable scenario execution	7
2.4.Websphere 5.0 Deployment.....	7
2.4.1.Ant tasks.....	7
2.4.2.speedo.properties file.....	7
2.4.3.Specific JDBC deployment.....	8
2.4.4.JCA connector.....	10
2.4.5.J2eedo Deployment.....	11
2.4.6.Http Execution.....	12
2.4.7.Recordable scenario execution	12

1. Introduction

1.1. Objective

The objective of this document is to integrate Speedo in a J2ee environment.

2. Deployment into Applications Servers

2.1. Introduction

This tutorial assumes that we have tested our application (J2eedo application) into Weblogic, Jonas and Websphere application Servers.

Every application server has a different installation process that you have to configure in purpose of your configuration.

This tutorial uses Ant in order to deploy applications. We recommend following our Ant instruction rather than deploying without using Ant module.

2.2. Common configuration

2.2.1. Prerequisite

On purpose to deploy J2eedo, you must:

- Having Weblogic 8.1 and/or Jonas 3.3.2 and/or Websphere 5.0.2 installed
- Having Jmeter 1.9.1 installed in order to run recorded tests.

2.2.2. Configuration

2.2.2.1. *Deployment*

Deployment is realized by an Ant task.

In order to deploy J2eedo into Application server, it is necessary to:

- configure test/j2ee/build.properties file
- Run install target from test/j2ee/build.xml

2.2.2.2. *build.properties configuration*

1.1.1.1.1. *Configure J2ee.jar and tools.jar classloader*

```
# Required
j2ee.lib = ...../j2ee.jar
tools.lib = ...../tools.jar
```

1.1.1.1.2. *Access to external resources*

```
# Required
external.lib = .....speedo_tronc/externals
```

1.1.1.1.3. *Define Database librairie in order to compile*

```
# Required
database.lib = ...../speedosrc/output/dist/lib/jdbc/postgres_jdbc2.jar
```

1.1.1.1.4. *Define Database name*

```
# Database name (possible values: 'oracle' or 'postgres')
# Required
#database.name = oracle
database.name = postgres
```

1.1.1.1.5. Application Server

```
# the name of the J2EE server to use
# possible values: jonas, weblogic or websphere
j2ee.server = weblogic
```

1.1.1.1.6. Application Name to deploy

This name will be used to create ejb, database, web and ear names.

```
# The application name is used as base name in order to build archive
# (jar,war,ear)
app.name = j2eedo
```

1.1.1.1.7. Application directory name

It correspond to the name of the application package name

```
# The application directory is the path relative to the source directory
('src')
# where the J2EE application will be expected
app.dir = j2eedo
```

1.1.1.1.8. JMeter directory

```
# the jmeter install directory
# don't forget to copy jmeter-ant.jar into $ANT_HOME/lib
jmeterhome = C:/jakarta-jmeter-1.9.1
```

2.2.2.3. Resource connector

Speedo follows JDO's specification deploying into application servers. Speedo uses JCA connector in order to integrate Speedo into a J2ee environment.

You will find 'rar' target into speedo's 'build.xml'.

Execute this target or use specific speedo.rar :

Server	Connector
Weblogic	Specific rar : speedo_for_weblogic_ra.rar
Jonas	speedo_for_jonas_ra.rar
Websphere	speedo_ra.rar

2.3. Weblogic 8.1 Deployment

2.3.1. Introduction

In this version 8.1.0, there are a number of issues with EAR and RAR's files classloaders. These issues may no longer exist in later releases of Weblogic.

On purpose to avoid theses issues, we configure our server in adding speedo.jar and speedo.properties classpath references.

2.3.2. Ant tasks

We have to configure Weblogic specific parameters in order to deploy j2eedo into Weblogic.

Open build-Weblogic.xml, and configure these two properties:

- weblogic.home : Weblogic installation directory
- deploy.dir : ear install directory.

Notice : Be sure that `j2ee.server = weblogic`

2.3.3. speedo.properties file

Speedo.properties file must be update according to environment parameters. :

- JDBC parameters
- ConnectionFactoryName : datasource JNDI's name. This name must be set too into Application server datasource properties. This is the preferable way to integrate Speedo into a J2EE environment,.

In order to avoid Weblogic 8.1 classpath issues, you have to define speedo.properties classpath to your server.

Example :

```
set CLASSPATH=C:\bea\weblogic81\common\lib;%CLASSPATH%
```

where :

- C:\bea\weblogic81\common\lib: represent speedo.properties directory.

2.3.4. Specific JDBC deployment

In order to configure XA database access, you have to deploy database's jar (**pg73jdbc3.jar for PostgreSQL**) into Weblogic classpath (configure *startWeblogic.cmd*)

You have to be sure that JDBC driver is in your server classpath.

Example :

```
set CLASSPATH=C:\bea\weblogic81\common\lib\pg73jdbc3.jar;  
C:\bea\weblogic81\common\lib\speedo.properties; %CLASSPATH%
```

First, you need to create a datasource and a connexion pool into your server.

This datasource must have the JNDI reference that you configure into *connectionFactoryName*'s parameter.

2.3.5. JCA connector

You have to configure the server classloader by adding your rar reference (speedo_for_weblogic.rar). It must be done by editing *startWeblogic.cmd* file.

Example :

```
set CLASSPATH=C:\bea\weblogic81\common\lib\pg73jdbc3.jar;  
C:\bea\weblogic81\common\lib\speedo.properties;C:\bea\weblogic81\common\lib\speedo.jar; %  
CLASSPATH%
```

where :

- C:\bea\weblogic81\common\lib\speedo.jar is the speedo distribution reference.

You can configure Speedo JCA by editing META-INF/ra.xml and/or META-INF/weblogic-ra.xml substituting parameters with your own values.

Notice: you can edit META-INF/weblogic-ra.xml to configure the JNDI location to which you want speedo to be bound.

2.3.6. J2eedo Deployment

In order to deploy an j2eedo application, execute install's target from j2eedo build.xml from test/J2ee directory.

It generates and deploy an Ear file composed by:

- 1 EJB module (jar)
- 1 WEB module (war)
- 1 Database module (jar)
- 1 Rar module

You can start administration console in order to verify J2eedo installation, and browse to your WebLogic admin port to ensure that Speedo and J2eedo are accessible to the proper servers.

2.3.7. Http Execution

J2eedo must be accessible from :

<http://server:7001/j2eedo/index.html>

2.3.8. Recordable scenario execution

You can execute Jmeter scenario by executing ant task into J2eedo build.xml

2.4. Websphere 5.0 Deployment

2.4.1. Ant tasks

We have to configure Websphere specific parameters in order to deploy j2eedo into Websphere.

Open build-Websphere.xml, and configure these two properties:

- Was5.home : Websphere Appserver directory
- deploy.dir : ear install directory.

Notice : Be sure that `j2ee.server = websphere`

2.4.2. speedo.properties file

Speedo.properties file must be update according to environment parameters. :

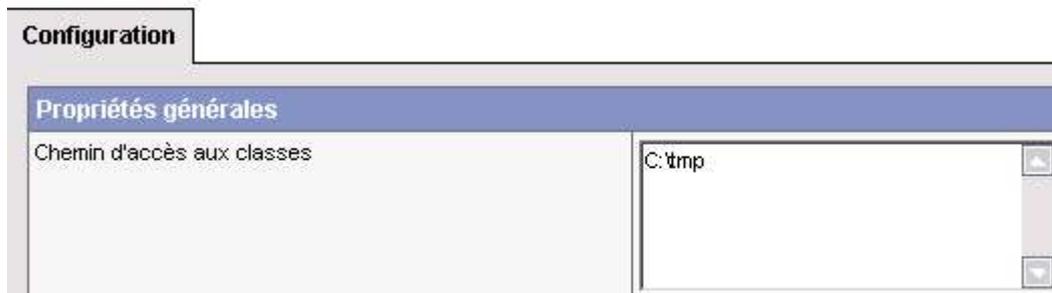
- JDBC parameters
- `ConnectionFactoryName` : datasource JNDI's name. This name must be set too into Application server datasource properties. This is the preferable way to integrate Speedo into a J2EE environment,.

In order to avoid Websphere classpath issues, you have to define `speedo.properties`' classpath to your server. This file must be set by configuring server JVM (define an classloader access by adding `speedo.properties` path)

[Serveurs d'applications](#) > [server1](#) > [Définition des processus](#) >

Machine virtuelle Java

Paramètres avancés de la machine virtuelle Java. 



2.4.3. Specific JDBC deployment

In order to configure XA database access, you have to deploy database's jar (**postgres_jdbc2.jar for PostgreSQL**)

First, you need to create a datasource and a connexion pool into your server.

This datasource must have the JNDI reference that you configure into `connectionFactoryName`'s parameter.

Configuration	
Propriétés générales	
Portée	* cells:g-necm5-100:nodes:g-necm5-100
Nom	* Datasource j2eedo
Nom JNDI	jdbc_1
Persistance gérée par conteneur	<input type="checkbox"/> Utiliser cette source de données dans la persistance CMP.
Description	<div style="border: 1px solid gray; height: 40px;"></div>
Catégorie	
Taille de la mémoire cache des instructions	10 instructions
Nom de classe auxiliaire de la source de données	com.ibm.websphere.rsadapter.Gene
Alias d'authentification géré par composant	g-necm5-100/PostGRES Authentication ▼
Alias d'authentification géré par conteneur	g-necm5-100/PostGRES Authentication ▼
<input type="button" value="Valider"/> <input type="button" value="OK"/> <input type="button" value="Restaurer"/> <input type="button" value="Annuler"/>	

Ps: you have to configure a J2C authentication

1) Access

☐ Sécurité

[Sécurité globale](#)

[SSL](#)

☐ Mécanismes d'authentification

[LTPA](#)

☑ Registres d'utilisateurs

☐ Configuration JAAS

[Connexions des applications](#)

[Données d'authentification J2C](#)

2) Configure

Configuration

Propriétés générales

Alias	* g-necml5-100/PostGRES Authentificat
ID utilisateur	* sureauso
Mot de passe	*
Description	

Valider OK Restaurer Annuler

2.4.4. JCA connector

The speedo_ra.rar connector contains a reference to speedo.jar.

You can configure Speedo JCA by editing META-INF/ra.xml substituting parameters with your own values.

When j2eedo.ear has been deployed, you have to configure JCA connector.

Logon to the WebSphere Administrative Console. Select the module Connector installation you provided. Click on the link marked J2C Connection Factories. This is here that you have to configure an instance of Speedo's JCA implementation. Create a new Factory by configuring the JNDI location to which you want speedo to be bound (This JNDI reference must be lookup in the Web.jar).

Configuration	
Propriétés générales	
Portée	* cells:g-necml5-100:applications:j2eedo.ear:deployments:j2eedo
Nom	* speedo
Nom JNDI	speedo
Description	
Catégorie	
Préférences d'authentification	Aucun

Notice: You can obtain a reference to the PMF by this way:

```
persistenceManagerFactory = (PersistenceManagerFactory) ictx.lookup("speedo");
```

2.4.5. J2eedo Deployment

In order to deploy an j2eedo application, execute install's target from j2eedo build.xml from test/J2ee directory.

Browse to the admin console in your web browser. Select Update or Install New Application from the navigation panel. Select the path to your j2eedo.ear file.

On the following screen, leave the options.

On the following screen, ensure that the Deploy EJBs option is checked.

On the Bean JNDI name screen, be sure that JNDI reference of the session bean is **ejb/storeService**.

Save the changes to the domain configuration.

It generates and deploy an j2eedo.ear file composed by:

- 1 EJB module (jar)
- 1 WEB module (war)
- 1 Database module (jar)
- 1 Rar module

You can start administration console in order to verify J2eedo installation, and browse to your Websphere admin port to ensure that Speedo and J2eedo are accessible to the proper servers.

2.4.6. Http Execution

J2eedo must be accessible from :

`http://server:9080/j2eedo/index.html`

2.4.7. Recordable scenario execution

You can execute Jmeter scenario by executing ant task into J2eedo build.xml

FICHE D'EVOLUTION

Vers.	Edit.	Date	§	Action	Nature de la modification
0	0	13/03/03		C	Création du document

§ = Chapitre(s) ou paragraphe(s) modifié(s) Action = C : Création, M : Modification, S : Suppression