

Exécution de JSPWiki sur le grand système (z/OS)

Table of Contents

- [Exécution de JSPWiki sur le grand système \(z/OS\)](#)
 - [Quel est le problème?](#)
 - [Installation de Tomcat](#)
 - [Installation de JSPWiki](#)
 - [Authentification \(et autorisation\) auprès de SAF](#)
 - [Realm Tomcat](#)
 - [JAAS Login module](#)
 - [Pièges](#)
 - [En supposant que ASCII](#)
 - [Fichiers de script shell au format DOS](#)
 - [Numéro de contrôle du programme MVS](#)

Ceci est un résumé des astuces et conseils pour exécuter **JSPWiki** dans Tomcat sur le mainframe.

Quel est le problème?

Le mainframe IBM exécute toute une série de systèmes d'exploitation:

- z/OS (anciennement appelé OS/390 et MVS)
- z/Linux
- VSE
- TPF
- z/VM (hyperviseur pouvant exécuter tous les systèmes d'exploitation mentionnés ci-dessus)
- Code CF
- Plus?

Le "système d'exploitation phare" est z/OS, principalement utilisé par les institutions financières et les multinationales. Z/OS s'exécute sur du matériel propriétaire IBM, le modèle actuel étant IBM System z10.



On peut également l'exécuter sous un émulateur Hercules.

z/OS propose des services système Unix. On peut donc exécuter des applications de type Unix sur le grand système, du moins c'est la théorie. En pratique, la plupart des applications ne peuvent pas être facilement portées.

Pour Java, c'est différent, Java offre une réelle indépendance par rapport à la plate-forme, donc exécuter des applications Java sur le grand système n'est pas une grosse affaire.

La principale différence majeure par rapport aux autres systèmes d'exploitation réside dans le codage par défaut, **EBCDIC** (CP037 ou Cp1047). Il en va de même pour les applications Java, bien que vous puissiez démarrer une machine virtuelle avec `-Dfile.encoding = ISO8859-1` (IBM WebSphere Application Le serveur le fait)

Pour la plupart des programmeurs systèmes, exécuter JSPWiki sur l'ordinateur central est un choix naturel:

- Connaître l'environnement
- Pouvoir réutiliser vos procédures de sauvegarde existantes
- Réutiliser les procédures de reprise après sinistre existantes
- Réutiliser les produits de gestion de système existants tels que WLM et SMF
- Intégration au package de sécurité RACF/ACF2 existant

Installation de Tomcat

- Télécharger Tomcat
- Décompresser
- Binaire le télécharger sur le mainframe
- Ascii→ebcdic convertit vos scripts shell
- Cela doit aller dans un HFS monté en lecture seule, vous pouvez ensuite créer des instances dans (par exemple) `/var/tomcatxx` (en anticipant environ $36 \wedge 2$ instances):

```
metskem@xat1:/var/tomcat00>l
total 120
lrwxrwxrwx   1 $$BPXRT  FUSPOS           22 Sep 26  2006 bin ->
/usr/local/tomcat/bin/
drwxrwx---   3 WIKI00U  USIZOS05        8192 Feb  9  09:25 conf
lrwxrwxrwx   1 $$BPXRT  USIZOS05        21 Jan 28  10:21 lib ->
/usr/local/tomcat/lib
drwxrwx---   2 WIKI00U  USIZOS05       28672 Feb 11  06:00 logs
drwxrwx---   2 WIKI00U  USIZOS05        8192 Jan 12  06:58 temp
drwxrwxr-x   5 WIKI00U  USIZOS05        8192 Feb 10  08:50 webapps
drwxrwx---   3 WIKI00U  USIZOS05        8192 Jan 12  07:07 work
metskem@xat1:/var/tomcat00/webapps>cd webapps
metskem@xat1:/var/tomcat00/webapps>l
total 10304
drwxrwxr-x  10 WIKI00U  USIZOS05        8192 Jan 12  13:44 ROOT
lrwxrwxrwx   1 $$BPXRT  USIZOS05         30 Jan 28  10:47 docs ->
/usr/local/tomcat/webapps/docs
lrwxrwxrwx   1 $$BPXRT  USIZOS05         38 Jan 28  10:47 host-manager ->
/usr/local/tomcat/webapps/host-manager
lrwxrwxrwx   1 $$BPXRT  USIZOS05         33 Jan 28  10:47 manager ->
/usr/local/tomcat/webapps/manager
```

- L'exécuter en tant que **batchjob** ou, mieux, en tant que tâche démarrée (STC)
- Un exemple de procédure STC:

```
//TOMCAT PROC ID='00',ACTION='run',JMXPORT='5001'
//*-----
//TOMCAT EXEC PGM=BPXBATCH,REGION=1000M,TIME=NOLIMIT,MEMLIMIT=2000M,
// PARM='SH /usr/local/sys2/bin/tomcat.sh &ID &ACTION &JMXPORT'
//STDOUT DD SYSOUT=*,RECFM=VB,LRECL=1024,HOLD=YES
//STDERR DD SYSOUT=*,RECFM=VB,LRECL=1024,HOLD=YES
```

Cela peut être démarré avec les commandes de l'opérateur suivantes:

```
S tomct,jobname=tomcat00,ID=00,ACTION='run'
S tomct,jobname=tomcat00,ID=00,ACTION='stop'
```

Avec un exemple de script de démarrage:

```
#!/bin/sh
#
# setup DB2 JDBC stuff:
#
export LIBPATH=/usr/lpp/db2810/jcc/lib:$LIBPATH
#-----
export ID=$1
export ACTION=$2
export JMXPORT=$3
#
export CATALINA_HOME=/var/tomcat$ID
export CATALINA_BASE=$CATALINA_HOME
export JAVA_HOME="/usr/lpp/java/J6.0_64"
#export JAVA_OPTS=" -Xmx256M -
Xshareclasses:name=RaboDefaultCache,verbose,groupAccess -Xscmx100M "
export JAVA_OPTS=" -Xmx256M "
export CATALINA_OPTS="$CATALINA_OPTS -
Djava.security.auth.login.config=/var/tomcat$ID/conf/jaas.config \
-Ddb2.jcc.propertiesFile=/var/tomcat$ID/conf/db2.jcc.properties \
-Dcom.sun.management.jmxremote.port=$JMXPORT \
-Dcom.sun.management.jmxremote.ssl=false \
-Dcom.sun.management.jmxremote.authenticate=true \
-
Dcom.sun.management.jmxremote.password.file=/var/tomcat$ID/conf/jmxremote.pa
ssword \
-
Dcom.sun.management.jmxremote.access.file=/var/tomcat$ID/conf/jmxremote.acce
ss"
#
umask 007
echo "Starting tomcat.sh with CATALINA_BASE=$CATALINA_BASE and
ACTION=$ACTION"
. $CATALINA_HOME/bin/catalina.sh $ACTION
```

Installation de JSPWiki

Fondamentalement la même procédure

- Télécharger **JSPWiki**
- Décompresser
- Téléchargement binaire dans le répertoire webapp de tomcat (ne vous contenter pas p le fichier de guerre!)
- Personnaliser votre jspwiki.properties (et éventuellement web.xml et jspwiki.policy)

Authentification (et autorisation) auprès de SAF

Pour ne pas exécuter la production avec le nom d'utilisateur par défaut userdatabase.xml/grouppdatabase.xml.

Voici donc quelques options pour utiliser votre système de sécurité SAF (RACF, ACF2, TopSecret).

Realm Tomcat

Vous trouverez ci-joint une implémentation de RACFRealm (info). Cela peut être utilisé lorsque vous avez configuré l'authentification gérée par le conteneur.

- Déposer ce fichier jar dans le répertoire lib de tomcat
- Configurer les éléments suivants (exemple) dans votre fichier server.xml:

```
<Resource name="RACFDatabase" auth="Container"
    type="nl.rabobank.hdw.tomcat.realm.RACFRealm"
    factory="nl.rabobank.hdw.tomcat.realm.RACFRealmFactory"
    description="RACF database"/>
<!-- Realm className="org.apache.catalina.realm.UserDatabaseRealm"
    resourceName="UserDatabase"/ -->

<Realm className="nl.rabobank.hdw.tomcat.realm.RACFRealm"/>
```

Donc, il faut remplacer le UserDatabaseRealm par le RACFRealm.

- Créer un profil de classe EJBROLE portant le nom <IDutilisateur>. <NomRôle> et autoriser les utilisateurs/groupes.
- Donc, si votre tomcat fonctionne avec l'ID utilisateur TOMCAT75, vous avez besoin du profil TOMCAT75.manager pour l'application de gestion.

Recharger **tomcat** et lancer l'application **tomcat manager**. On doit être invité (authentification de base) à saisir l'ID utilisateur/mot de passe RACF.

Pour utiliser ceci pour **JSPWiki**, activer simplement l'authentification gérée par conteneur dans le fichier web.xml, puis créer des profils pour les noms de rôle mentionnés dans le fichier web.xml de **JSPWiki** (Admin/Authentifié).

JAAS Login module

Le module de connexion JAAS est exactement ce qu'il dit: c'est une implémentation de `javax.security.auth.spi.LoginModule`.

Pour l'utiliser dans JSPWiki:

- Déposer le module **racflogin** dans le répertoire lib de tomcat
- Le configurer dans **jspwiki.properties**:

```
jspwiki.loginModule.class = nl.rabobank.hdw.auth.login.RACFLoginModule
```

- N'utiliser pas l'authentification gérée par conteneur dans ce cas.

Voilà, recycler votre tomcat et connecter-vous

On peut également utiliser le module de connexion JAAS IBM
com.ibm.security.auth.module.OS390LoginModule.

Pour plus d'informations à ce sujet consulter un RedBook SG24-7610: Java Security on z/OS - The Complete View.

Ce module ne transmet toutefois pas les messages en cas d'échec de la connexion.

Pièges

En supposant que ASCII

Certaines applications Java supposent faussement que le codage par défaut est ASCII et que ceux-ci ne fonctionnent pas correctement sur le grand système.

Ainsi, lorsque vous effectuer des E/S (fichiers, sockets), réfléchir à l'encodage des données que vous écrire/liser:

```
String encoding = "ISO8859-1";
String fileName = "/tmp/myWonderfulFile";
//
// lecture:
BufferedReader in = new BufferedReader(new InputStreamReader(new
FileInputStream(new File(fileName)), encoding));
//
// écrire:
BufferedWriter out = new BufferedWriter(new OutputStreamWriter(new
FileOutputStream(new File(fileName)), encoding));
// Spécifie un codage lors de l'utilisation de la méthode getBytes pour
convertir une chaîne en un ByteArray:
```

```
byte myBytes = myString.getBytes("8859_1");

// Spécifie également un codage lors de la création d'une chaîne à partir
d'un objet ByteArray:

String myString = new String(myBytes, "8859_1");
```

Fichiers de script shell au format DOS

Si vos scripts de shell sont au format DOS (au lieu du format Unix), ils ne s'exécutent pas. Le moyen le plus simple de vérifier ceci est votre éditeur préféré (UltraEdit ou autre chose) et de les convertir au format Unix. Vous pouvez également le vérifier avec la commande hexdump:

```
metskem@xat1:/usr/local/tomcat/bin>hexdump version.sh | grep '0d 15'
```

On ne doit pas voir autant d'occurrences '0d 15'.

On peut supprimer ces caractères hex'15 'avec (exemple):

```
cat version.sh | tr -d '\r' > versionNew.sh
```

Numéro de contrôle du programme MVS

Pour valider un ID utilisateur/mot de passe, vous devez appeler une fonction autorisée MVS (RACINIT).

Pour que cela réussisse, l'environnement doit être "contrôlé par le programme", voir le chapitre consacré à BPX.DAEMON FACILITY dans le guide de planification des services système UNIX:

Si la ressource BPX.DAEMON de la classe FACILITY est définie, votre système est doté de la sécurité z/OS UNIX.

Votre système peut exercer plus de contrôle sur vos superutilisateurs.

Si vous avez choisi ce niveau de sécurité, vous devez préalablement configurer les éléments suivants:

- Toutes les bibliothèques de chargement à partir desquelles les modules sont chargés dans l'espace d'adressage doivent être protégées par RACF PADS.
- Toutes les dll chargées à partir du système de fichiers doivent avoir leur bit de contrôle de programme activé (utiliser shellcmd extattr + p blabla.so)
- Tous les systèmes de fichiers dont les dll sont chargées doivent être montés avec l'attribut `setuid mount -s setuid ...`, c'est le BTW par défaut.
- L'utilisation Le disque qui exécute le wiki doit avoir un permis READ sur la classe FACILITY BPX.DAEMON

L'une de ces exigences peut échouer, rechercher les symptômes possibles suivants dans MVS SYSLOG:

```
ICH420I PROGRAM CELHV003 FROM LIBRARY CEE.SCEERUN2 CAUSED THE ENVIRONMENT TO  
BECOME UNCONTROLLED.
```

```
BPXP014I ENVIRONMENT MUST BE CONTROLLED FOR DAEMON (BPX.DAEMON) PROCESSING.
```

```
BPXP015I HFS PROGRAM /A0/usr/lpp/java/J6.0_64/lib/s390x/libwrappers.so  
IS NOT MARKED PROGRAM CONTROLLED.  
BPXP014I ENVIRONMENT MUST BE CONTROLLED FOR DAEMON (BPX.DAEMON)  
PROCESSING.
```

Vous pouvez vérifier avec:

- Utiliser ls -E, le "p" (contrôle du programme) doit être présent dans les bits de mode

```
metskem@xat1:/usr/lpp/java/J6.0_64>ls -E ./bin/j9vm/libjvm.so  
-rwxr-xr-x  aps-  1 $$BPXRT  OMVS      225280 Nov  8 06:21  
./bin/j9vm/libjvm.so
```

- Utiliser df -v pour vérifier les attributs de montage (le deuxième affichage indique le non SUID):

```
metskem@xat1:/usr/lpp/java/J6.0_64>df -v .  
Mounted on      Filesystem                Avail/Total      Files      Status  
/A0/usr/lpp/java/J6.0_64 (SYS8.OMVS.JAVA16.A0)  11216/868320    4294962897  
Available  
HFS, Read Only, Device:14437, ACLS=Y  
File System Owner : XAT1      Automove=Y      Client=N  
Filetag : T=off  codeset=0  
  
metskem@xat1:/usr/lpp/java/J6.0_64>df -v /var/tomcat00/  
Mounted on      Filesystem                Avail/Total      Files      Status  
/XAT1/var/tomcat00 (SYSTEM.XAT1.HFS.VAR.TOMCAT00) 736312/1792800 4294966301  
Available  
HFS, Read/Write, Device:14533, ACLS=Y, No SUID  
File System Owner : XAT1      Automove=U      Client=N  
Filetag : T=off  codeset=0
```

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=ibm:jspwiki-zos>

Last update: **2025/02/19 10:59**

