

ZOS: Installer Apache Tomcat

Table of Contents

- [ZOS: Installer Apache Tomcat](#)
 - [Installation de Tomcat](#)
 - [Exécution en tant que tâche démarrée \(STC\)](#)
 - [Installation de modules web](#)
 - [Authentification \(et autorisation\) gérée par SAF](#)
 - [Tomcat Realm](#)
 - [Connexion avec le module de connexion JAAS](#)

Installation de Tomcat

- Télécharger le package binaire Core apache-tomcat-x.x.x.zip sur tomcat.apache.org.
- Télécharger les fichiers zip dans le répertoire d'installation de Tomcat au sein des services système UNIX sur le système z/OS cible. Le répertoire d'installation de Tomcat doit avoir une taille d'au moins 36 Mo. Par conséquent, le jeu de données de montage d'hébergement doit avoir le nombre correspondant de cylindres.
- Se connecter aux services système UNIX et décompresser le fichier zip (Cela créera un sous-répertoire "apache-tomcat-x.x.x"):

```
jar -xvf apache-tomcat-x.x.x.zip
```

- S'assurer que ces variables d'environnements CATALINA_HOME=tomcat_directory/apache-tomcat-x.x.x et _BPXK_AUTOCVT=ON sont définies. (Remplacer tomcat_directory par le répertoire d'installation Tomcat actuel et x.x.x par la version actuelle de Tomcat.) On peut définir ces paramètres à l'aide des instructions "export" appropriées dans les répertoires \$HOME/.profile ou /etc/profile. Par exemple:

```
export CATALINA_HOME=$HOME/tomcat/apache-tomcat-5.5.x
export _BPXK_AUTOCVT=ON
```

- Vérifier également que la variable \$JAVA_HOME soient définies sur le répertoire de base Java actuel et que la variable \$PATH contient \$JAVA_HOME/bin. \$LIBPATH doit contenir \$JAVA_HOME/bin et \$JAVA_HOME/bin/classic.
- Pour que les fichiers de script Tomcat soient exécutables, les marquer comme suit:

```
chmod -t -c IS08859-1 $ CATALINA_HOME/bin/*.sh
```



Le service Web DFSMSrmm utilise une DLL C++ à partir de la liste de liens z/OS. L'objet programme s'appelle EDGXHCLL. Pour rendre cette DLL disponible pour le service Web DFSMSrmm, installer un lien dans le chemin de la bibliothèque de systèmes \$LIB_PATH.



Accéder au chemin de la bibliothèque dans le système de fichiers, par exemple: aller dans /usr/lib, puis taper `ln -e EDGXHCLL libEDGXHCLL.so`. Cette étape établit un lien externe vers la DLL dans LINKLST.

Exécution en tant que tâche démarrée (STC)

Exemple STC proc:

```
//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'
```

Exemple de script de démarrage:

```
#!/bin/sh
#
# configuration des éléments DB2 JDBC:
#
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$ID/conf/jmxremote.password \
-
Dcom.sun.management.jmxremote.access.file=/var/tomcat$ID/conf/jmxremote.access"
#
umask 007
echo "Démarrer de tomcat.sh avec CATALINA_BASE=$CATALINA_BASE and ACTION=$ACTION"
. $CATALINA_HOME/bin/catalina.sh $ACTION
```

Installation de modules web

Par exemple pour installer JSPWiki le procédure est fondamentalement la même:

- télécharger JSPWiki
- décompresser l'archive
- télécharger le binaire dans le répertoire webapp de tomcat (pas uniquement le fichier warfile!)
- personnaliser jspwiki.properties (et éventuellement web.xml et jspwiki.policy)



Le nom d'utilisateur par défaut est géré dans userdatabase.xml/grouppdatabase.xml.

Authentification (et autorisation) gérée par SAF

Pour utiliser le système de sécurité SAF (RACF, ACF2, TopSecret) voici quelques options.

Tomcat Realm

Cela peut être utilisé lorsqu'on a configuré l'authentification gérée par le conteneur.

- Télécharger le fichier RACFRealm.jar
- déposer ce fichier jar dans le répertoire lib de tomcat
- configurer ce qui suit (exemple) dans server.xml (**RACFRealm** remplace **UserDatabaseRealm**):

```
<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"/>
```

- créer un profil de classe EJBROLE portant le nom <userid>.<roleName> et autoriser les utilisateurs/roupes (ex: si tomcat fonctionne avec l'ID utilisateur TOMCAT75, il faut créer un profil TOMCAT75.manager pour l'application de gestion).
- Rafraichir tomcat et lancer l'application tomcat manager. On doit être invité (authentification de base) à saisir ID utilisateur/mot de passe RACF.



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

Cette méthode d'authentification présente un inconvénient majeur: elle ne renvoie pas de messages d'erreur utiles.

Connexion avec le module de connexion JAAS

Le module de connexion **JAAS** est une implémentation de `javax.security.auth.spi.LoginModule`.

Pour l'utiliser dans **JSPWiki**, par exemple :

- télécharger le fichier racfloginmodule
- déposer le dans le répertoire lib de tomcat
- configurer jspwiki.properties:

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

- ne pas utiliser l'authentification gérée par conteneur dans ce cas
- rafraichir tomcat et se reconnecter

From:

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

Permanent link:

<http://www.ouarte.garden/doku.php?id=ibm:zos-os390-tomcat>

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

