

# **JPlatform 10.0**

## **Manuel d'installation et d'exploitation**

© 2022 JaliOS

v1.0.25





<b>1</b>	<b>INTRODUCTION .....</b>	<b>6</b>
<b>2</b>	<b>PRE-REQUIS .....</b>	<b>7</b>
<b>3</b>	<b>ARCHITECTURE DE JPLATFORM 10.0 .....</b>	<b>8</b>
3.1	ARCHITECTURE GENERALE.....	8
3.2	STOCKAGE DES DONNEES .....	8
<b>4</b>	<b>CONFIGURATION DE L'ENVIRONNEMENT D'INSTALLATION .....</b>	<b>10</b>
4.1	SYSTEME D'EXPLOITATION.....	10
4.2	MACHINE VIRTUELLE JAVA (JVM) .....	10
4.3	CONFIGURATION DE LA BASE DE DONNEES .....	10
4.4	ENCODAGE DES CARACTERES .....	10
4.5	FAQ .....	10
<b>5</b>	<b>INSTALLATION DE JPLATFORM SUR LE SERVEUR D'APPLICATION .....</b>	<b>11</b>
5.1	PRINCIPES GENERAUX.....	11
5.2	INSTALLATION SUR APACHE TOMCAT 8.5 OU APACHE TOMCAT 9.0 .....	11
5.3	INSTALLATION SUR IBM WEBSHERE APPLICATION SERVER 9.0 .....	15
5.4	INSTALLATION SUR JBOSS EAP 7.0, 7.1, 7.2 OU 7.3 .....	28
<b>6</b>	<b>DEMARRAGE ET PARAMETRAGE DE JPLATFORM .....</b>	<b>31</b>
6.1	DEMARRAGE DE L'APPLICATION.....	31
6.2	MODIFICATION DU MOT DE PASSE ADMINISTRATEUR.....	32
6.3	CONFIGURATION INITIALE DE L'APPLICATION .....	32
6.4	CONFIGURATION DE LA BASE DE DONNEES .....	33
6.5	CONFIGURATION E-MAIL.....	34
6.6	CONFIGURATION LDAP .....	36
6.7	CONFIGURATION DU PROXY DE SORTIE.....	38
6.8	CONFIGURATION DU REVERSE PROXY.....	38
6.9	CONFIGURATION DU FORMAT DES URLS D'AFFICHAGE.....	39
6.10	CONFIGURATION DU POOL DE CONNEXIONS .....	40
6.11	CONFIGURATION DE LA REPLICATION PAR JSYNC .....	40
6.12	AJOUT D'UN ADD-PACK .....	41
6.13	CONFIGURATION DU SERVEUR WEB FRONTAL .....	42
6.14	INSTALLATION DE JALIOS OFFICE LAUNCHER .....	42
6.15	AJOUT DE MODULES.....	43

<b>7</b>	<b>PROCEDURES D’EXPLOITATION .....</b>	<b>44</b>
7.1	ARRET/DEMARRAGE DE JPLATFORM .....	44
7.2	MONITORING .....	45
7.3	CONTROLE D’INTEGRITE DES DONNEES .....	46
7.4	JOURNAL DES EVENEMENTS (LOGS) .....	47
7.5	ARRET DES ECRITURES .....	48
7.6	GESTION DES INDEX DE RECHERCHE .....	49
7.7	GESTION DE CACHES .....	49
7.8	NETTOYAGE DU STORE .....	50
7.9	PROCEDURES DE SAUVEGARDE/RESTAURATION .....	51
<b>8</b>	<b>MISE A JOUR ET DEPLOIEMENT .....</b>	<b>53</b>
8.1	PROCEDURE GENERALE .....	53
8.2	DEVELOPPEMENT ET PREPARATION DU FICHIER WAR .....	53
8.3	DEPLOIEMENT DU WAR SUR LE SERVEUR .....	54
8.4	REMARQUES .....	55
<b>9</b>	<b>EQUIPEMENT ET CONFIGURATION .....</b>	<b>56</b>
9.1	CONFIGURATION DU SERVEUR .....	56
9.2	CONFIGURATION DES POSTES .....	59



## **1 Introduction**

Ce document décrit l'architecture, l'installation et la configuration de JPlatform 10.0 (Jalios Digital Platform).

Les procédures d'installation détaillées sont décrites pour les différents serveurs d'applications certifiés.

Les principales procédures d'exploitation du produit sont décrites, notamment, les procédures de mises à jour et de déploiements.

## 2 Pré-requis

Les composants pré-requis décrits dans cette section sont nécessaires au bon fonctionnement de Jalios JPLATFORM 10.0. Dans la suite de ce document, ils seront considérés comme installés et testés. Reportez-vous à la section 9 pour la description détaillée des configurations matérielles et logicielles.

Composants obligatoires :

- Serveur d'applications Java EE (Apache Tomcat, IBM WebSphere AS, RedHat JBoss EAP)
- Serveur SMTP

Composants optionnels :

- SGBDR externe (MySQL, Oracle, PostgreSQL, Microsoft SQL Server, MariaDB)
- Serveur Web (Apache, IHS, ...)
- Serveur LDAP (ActiveDirectory, Novell eDirectory, OpenLDAP, Sun Directory Server, ...)
- Serveur SSO (CAS, RSA ClearTrust, ...)
- Serveur POP3 ou IMAP

## 3 Architecture de JPlatform 10.0

### 3.1 Architecture générale

JPlatform 10.0 est une webapp Java EE 6. Elle fonctionne au sein d'un serveur d'application. Plusieurs instances de JPlatform 10.0 peuvent être déployées sur le même serveur d'application. Le schéma ci-dessous illustre l'architecture générale d'une application JPlatform 10.0.

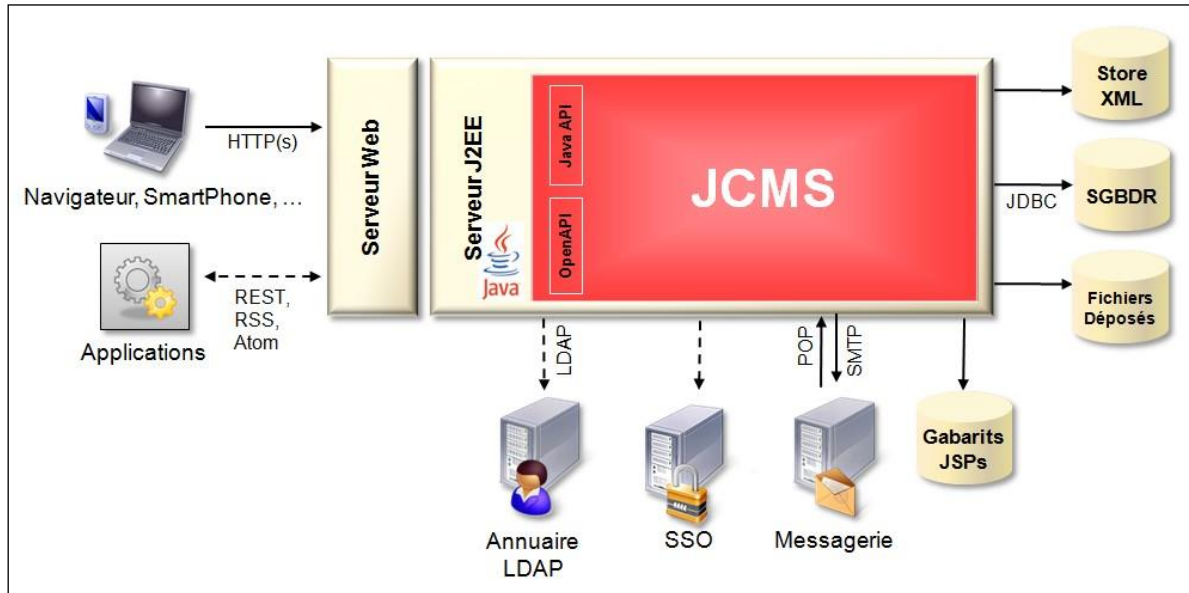


Figure 1 - Architecture générale de JPlatform 10.0

### 3.2 Stockage des données

JPlatform 10.0 intègre un double mécanisme de stockage. Les données sont réparties entre *JStore* et un *JcmsDB*. Les données de structure (espaces de travail, groupes, catégories, portlets), les membres et les contenus sont gérées dans *JStore*. Les données techniques, les archives, les préférences et les contenus utilisateurs (forum, avis, commentaire, sondage, ...) sont stockées dans la base de données relationnelle, *JcmsDB*.

#### 3.2.1 JStore

*JStore* est une base de données objets dont la persistance est assurée par la journalisation des opérations de modification. Au démarrage de l'application, l'ensemble des objets est chargé en mémoire en rejouant les opérations contenues dans le fichier `store.xml`. En cours d'exploitation, lorsqu'un objet est créé, modifié ou supprimé, l'opération décrivant cette écriture est ajoutée au journal. Seuls les attributs des objets sont chargés en mémoire et stockés dans `store.xml`. Cela ne concerne donc pas les fichiers déposés sur le serveur, les gabarits de présentation (JSP), les feuilles de styles, ... Deux métriques sont donc à considérer avec *JStore* : le volume mémoire et le temps de chargement.

Le volume mémoire dépend du nombre d'objets vivants à charger et de la taille moyenne de ces objets. La taille d'un objet est essentiellement liée au volume de texte stocké dans l'objet (typiquement, un objet JPlatform sans aucun champ texte pèse moins de 1 Ko). *JStore* repose sur les API Java dans lesquelles les chaînes de caractères sont stockées en Unicode sur 2 octets. Le volume mémoire occupé est donc un peu plus du double du volume de texte. Par exemple, une page A4 de texte représentant 3 Ko de texte brut sur disque occupera 7 Ko en mémoire dans *JStore* (index compris). Un site JPlatform gérant 50 000 contenus de ce type occupera donc environ 400 Mo.



En règle générale, on peut considérer les métriques suivantes :

- Les petits objets (fiche documentaire, catégorie, portlet, membre, ...) occupent en moyenne 3 Ko. 300 000 objets de cette sorte occupent environ 1 Go de mémoire
- Les gros objets (contenus avec des champs textes) occupent en moyenne 10 Ko. 100 000 objets de cette sorte occupent environ 1 Go de mémoire.

Le temps de chargement dépend du nombre d'opérations présentes dans le journal `store.xml`. Ce sont essentiellement les opérations de création qui sont coûteuses. Le temps de chargement est proportionnel au nombre et à la taille des objets. Par exemple, un journal de 50 000 objets représentant chacun une page A4 de texte brut (3 Ko sur disque) se charge en 1 minute sur un PC à 2 GHz avec 2 Go de RAM et des disques à 7200 rpm.

### 3.2.2 JcmsDB

JPlatform 10.0 stocke une partie de ses données dans une base de données relationnelle, *JcmsDB*, gérée par un SGBDR.

Le choix du SGBDR est fortement structurant et doit être fait lors de la mise en place de l'architecture. JaliOS ne fournit pas d'outils de conversion de base de données : il est très difficile de changer ce choix a posteriori.

JPlatform 10.0 est livré avec le SGBDR embarqué Derby. Ce SGBDR est bien adapté pour les environnements de développement ou d'expérimentation. Derby ne doit pas être utilisé sur un environnement de production.

La création de la base *JcmsDB* et les procédures d'exploitation (notamment la sauvegarde régulière) sont à la charge de l'administrateur du SGBDR. Dans les environnements utilisant Derby, la création de la base et les sauvegardes sont pris en charge par JPlatform.

Au premier démarrage, JPlatform génère dans *JcmsDB* l'ensemble des tables et des index nécessaires (quel que soit le SGBDR retenu). Cette structure peut évoluer selon les créations et modifications des types de publications et les ajouts de modules.

## 4 Configuration de l'environnement d'installation

### 4.1 Système d'exploitation

Sous Unix, il est recommandé de configurer la limite du nombre de fichiers ouverts simultanément à une valeur supérieure ou égale à 4096. Attention de ne pas dépasser la limite définie au niveau du kernel (`/proc/sys/fs/file-max`) ou de modifier cette limite également.

Vérifiez les limites dans un shell de l'utilisateur qui sert à lancé le serveur d'application avec la commande `ulimit`.

Configurez les limites de votre système (`/etc/limits` ou `/etc/security/limits.conf`)

### 4.2 Machine Virtuelle Java (JVM)

Le serveur d'application doit être configuré pour fonctionner avec l'une des JVM suivantes :

- Oracle JDK 11 -- à partir de JPlatform 10 SP2
- OpenJDK 11 (HotSpot) -- à partir de JPlatform 10 SP2
- Oracle Java SE 8
- OpenJDK 8 (HotSpot)
- IBM Java SE 8 pour IBM WebSphere AS.

Il est nécessaire de configurer la mémoire allouée à la JVM. L'allocation peut être paramétrée en spécifiant les paramètres : `-Xmx`(taille maximum du tas) et `-Xms`(taille initiale du tas). Pour JPlatform il est recommandé de mettre ces deux paramètres à la même valeur (afin d'éviter des réallocations durant le redémarrage de la webapp).

Le détail du paramétrage est indiqué à la section 5 lors l'installation de JPlatform sur le serveur d'application.

### 4.3 Configuration de la base de données

Si vous avez choisi d'utiliser le SGBDR embarqué Derby, vous n'avez aucun paramétrage à effectuer. Si vous utilisez un SGBDR externe vous devez tout d'abord installer JPlatform avec Derby puis suivre la procédure décrite à la section 6.3.

Pour JPlatform10, la base doit être créée avec un encodage supportant tous les caractères UTF-8. Consultez la fiche du module de base de données retenu pour plus de détails.

### 4.4 Encodage des caractères

JPlatform 10.0 utilise le mode d'encodage des caractères UTF-8. Il convient donc d'utiliser un éditeur texte supportant UTF-8 pour éditer les fichiers de JPlatform.

Configurer votre éditeur texte pour utiliser l'encodage UTF-8, sans BOM (Byte Order Mark).

### 4.5 FAQ

Pour les autres questions sur la configuration de l'environnement, reportez-vous à la FAQ sur Jalios Community : <https://community.jalios.com/faq>.

## 5 Installation de JPlatform sur le serveur d'application

### 5.1 Principes généraux

JPlatform 10.0 est une *webapp* fournie sous la forme d'un fichier *war* : `jcms-10.0.0.war`.

L'installation de JPlatform consiste à déployer ce fichier sur le serveur d'application. Il est indispensable que lors du déploiement, le fichier *war* soit décompacté car l'application lit et écrit des données dans son répertoire de déploiement.

Remarques : si le fichier *war* a été téléchargé avec Internet Explorer, une extension `.zip` a pu être ajoutée. Il faut *supprimer* l'extension `.zip` avant de procéder à l'installation.

Lors du déploiement, il faut choisir un nom de contexte associé à l'application. Ce nom est utilisé pour la construction de l'URL d'accès au site. Dans les procédures d'installation décrites ci-dessous, nous utiliserons le nom de contexte `jcms10`.

Afin de garantir la sécurité de votre plateforme et de vous permettre de toujours bénéficier des derniers correctifs, nous vous recommandons d'utiliser systématiquement la dernière version de maintenance du serveur d'application (en conservant la même version majeure indiquée).

### 5.2 Installation sur Apache Tomcat 8.5 ou Apache Tomcat 9.0

Téléchargez la dernière version de maintenance de Tomcat 9.0 sur le site suivant :

<https://tomcat.apache.org/>

#### 5.2.1 Configuration du serveur d'application

Si JPlatform est installé en tant que *webapp* racine (répertoire `ROOT/`), il peut y avoir des conflits d'accès entre des ressources situées dans les répertoires de JPlatform 10.0 et des *webapps* portant le même nom. Dans ce cas, il faut veiller à renommer les *webapps* voisines afin qu'elles disposent d'un accès non conflictuel avec JPlatform. C'est par exemple le cas avec la *webapp* d'administration de Tomcat et le répertoire `admin/` de JPlatform. Il faut éditer le fichier `TOMCAT_DIR/conf/Catalina/.../admin.xml` et modifier le chemin d'accès de la *webapp* en modifiant l'attribut `path` de la balise `Context` à une autre valeur, par exemple `/adminTomcat`.

#### Ajout de `tools.jar` (Java 8 uniquement)

Pour permettre à JPlatform 10.0 de générer et compiler les classes des types, la bibliothèque `tools.jar` (contenant le compilateur `javac`) doit être ajoutée à Tomcat. Cette action n'est pas nécessaire avec Java 11.

Sous Unix :

Ouvrez fichier `TOMCAT_DIR/bin/setclasspath.sh` et placez-vous en toute fin de fichier. Ajoutez les lignes ci-dessous :

```
# Jalios : tools.jar required to run JPlatform
# Set standard CLASSPATH
CLASSPATH="$JAVA_HOME/lib/tools.jar"
```

Sous Windows:

Lorsque Tomcat est lancé en service Windows, il faut modifier le *classpath* utilisé par le service. Pour cela, utilisez l'utilitaire graphique disponible dans la barre des tâches de Windows (installé lorsque le mode service est sélectionné). Modifiez le *classpath* (dans l'onglet Java) pour y ajouter le chemin vers le fichier `C:\{jdk_path}\lib\tools.jar`, précédé d'un point-virgule (;).

Si vous lancez Tomcat sous Windows par le script de lancement, ouvrez le fichier `TOMCAT_DIR\bin\setclasspath.bat` et placez-vous ligne 81 avant la ligne :

```
goto end
```

Ajoutez les lignes ci-dessous :

```
rem Jalios : tools.jar required to run JPlatform
rem Set standard CLASSPATH
set "CLASSPATH=%JAVA_HOME%\lib\tools.jar"
```

### **Configuration de l'allocation mémoire**

Il est nécessaire de configurer l'allocation mémoire de la JVM. Ceci se fait en positionnant des attributs via la variable d'environnement `JAVA_OPTS`.

Soit en ajoutant la ligne suivante dans les scripts de lancement de Tomcat (`catalina.sh` et `catalina.bat`), soit en positionnant la variable d'environnement avant l'exécution des scripts de lancement.

#### Sous Unix :

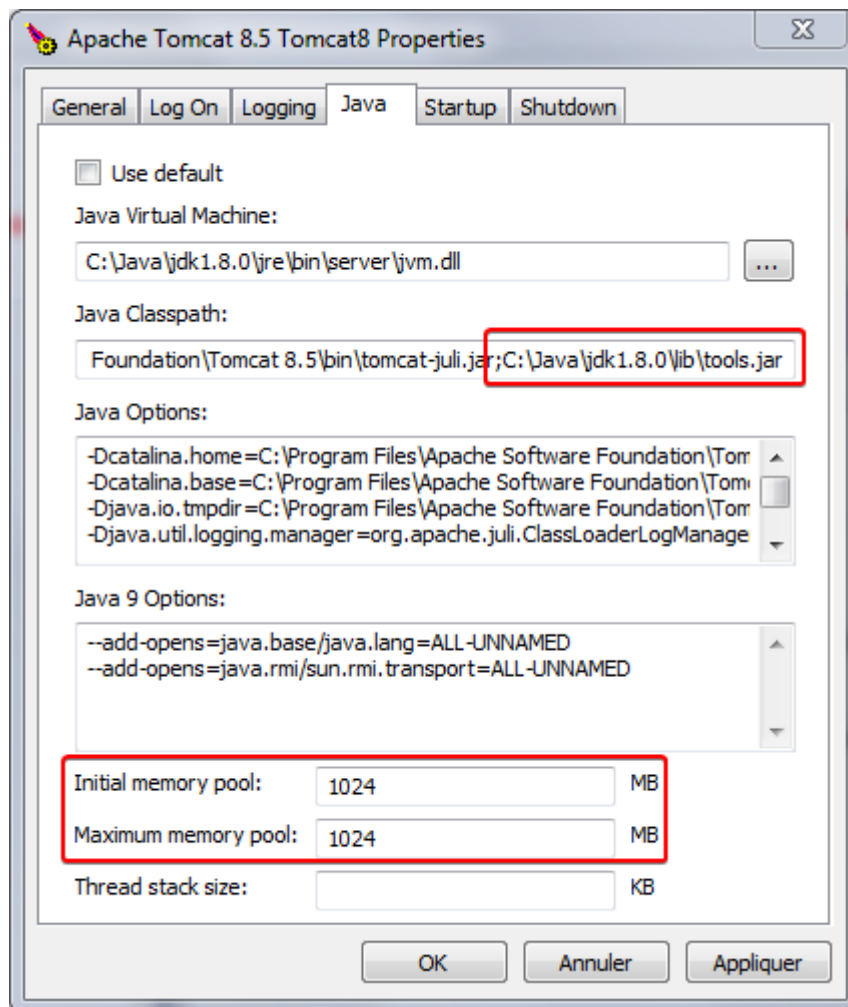
```
JAVA_OPTS="-Xmx1024m -Xms1024m"
```

#### Sous Windows :

```
set JAVA_OPTS=-Xmx1024m -Xms1024m
```

Si Tomcat est exécuté en tant que service Windows, éditer les propriétés via le menu Programmes > Apache Tomcat > Configure Tomcat, puis dans l'onglet Java :

- Spécifiez la valeur 1024 dans les champs "Initial memory pool" et "Maximum memory pool"



### Configuration des cookies de session en HttpOnly

Afin de se prémunir contre le vol de cookie via JavaScript, il faut configurer Tomcat pour que le cookie de session Java EE (JSESSIONID) positionne l'attribut `HttpOnly`.

Pour cela, modifiez la configuration du contexte par défaut de Tomcat et ajouter l'attribut `useHttpOnly="true"` dans balise `Context` du fichier `conf/context.xml`:

```
<Context [...] useHttpOnly="true">
  [...]
</Context>
```

### Configuration en mode production

Tomcat est livré en standard avec le mode développement activé. Afin d'améliorer les performances, modifier le fichier `conf/web.xml` de tomcat pour positionner le paramètre `development` à `false` sur la `JspServlet` :

Il est également nécessaire de positionner le paramètre `genStringAsCharArray` à `true` sur cette même servlet pour assurer une compilation correcte de l'ensemble des JSP fourni par JPlatform.

```
<servlet>
  <servlet-name>jsp</servlet-name>
  <servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
  [...]
</servlet>
```

```
<init-param>
  <param-name>development</param-name>
  <param-value>>false</param-value>
</init-param>

<init-param>
  <param-name>genStringAsCharArray</param-name>
  <param-value>>true</param-value>
</init-param>

  [...]
</servlet>
```

### Configuration de la compilation des JSP pour Tomcat 8.5

Tomcat 8.5 est livré en standard avec une configuration qui n'autorise pas la syntaxe Java 8 dans les JSP. Pour garantir la compilation correcte de tous les JSP livrés dans l'application, modifier le fichier `conf/web.xml` de tomcat pour positionner les paramètres `compilerSourceVM` et `compilerTargetVM` à 1.8 sur la `JspServlet` :

```
<servlet>
  <servlet-name>jsp</servlet-name>
  <servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
  [...]

  <init-param>
    <param-name>compilerSourceVM</param-name>
    <param-value>1.8</param-value>
  </init-param>
  <init-param>
    <param-name>compilerTargetVM</param-name>
    <param-value>1.8</param-value>
  </init-param>

  [...]
</servlet>
```

Cette configuration n'est pas nécessaire avec Tomcat 9.0.

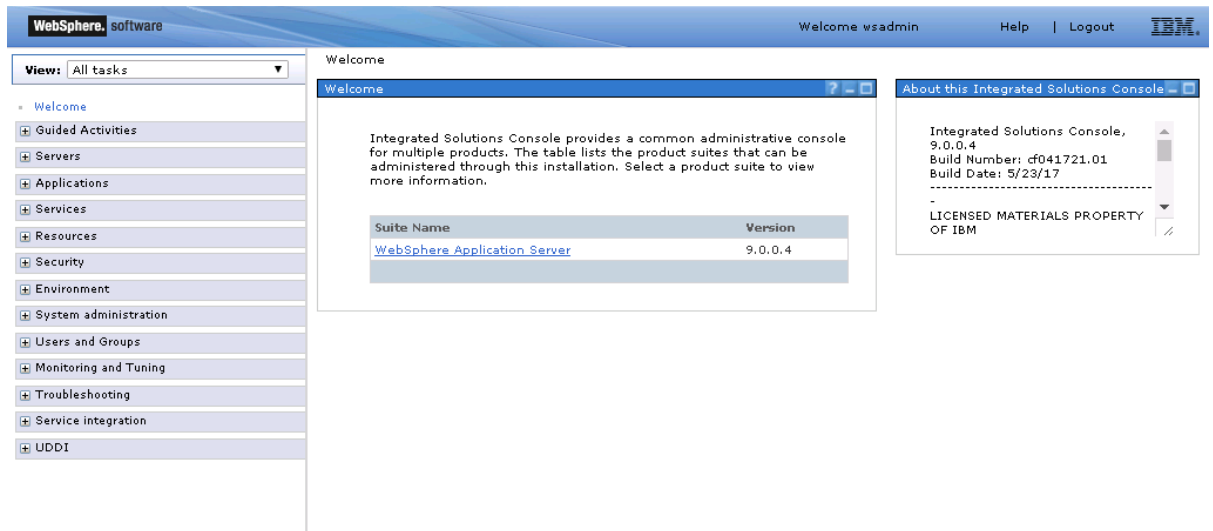
#### 5.2.2 Déploiement de l'application

1. Arrêtez Tomcat si nécessaire,  
Sous Windows en mode service : *Programmes > Apache Tomcat> Stop Tomcat* ;
2. Allez dans le répertoire `<TOMCAT_DIR>/webapps` ;
3. Créez un répertoire `jcms10` ;
4. Décompressez fichier `jcms-10.0.0.war` dans ce répertoire ;
5. Démarrez Tomcat,  
Sous Windows en mode service : *Programmes > Apache Tomcat> Start Tomcat* ;
6. JPlatform est maintenant prêt à fonctionner.

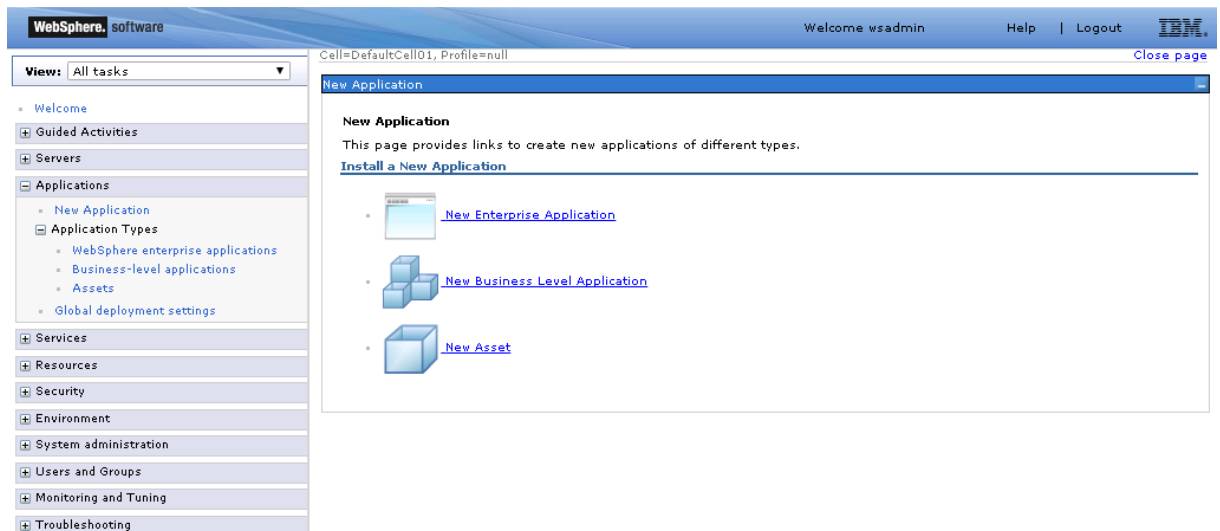
## 5.3 Installation sur IBM WebSphere Application Server 9.0

### 5.3.1 Déploiement de l'application

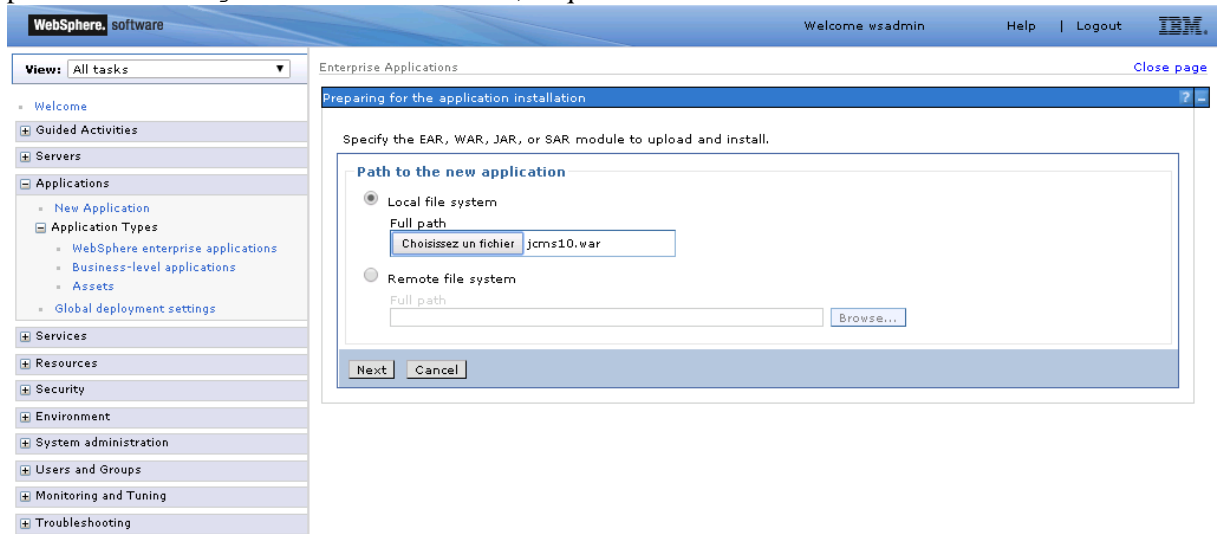
1. Démarrez WebSphere
2. Ouvrez la console d'administration



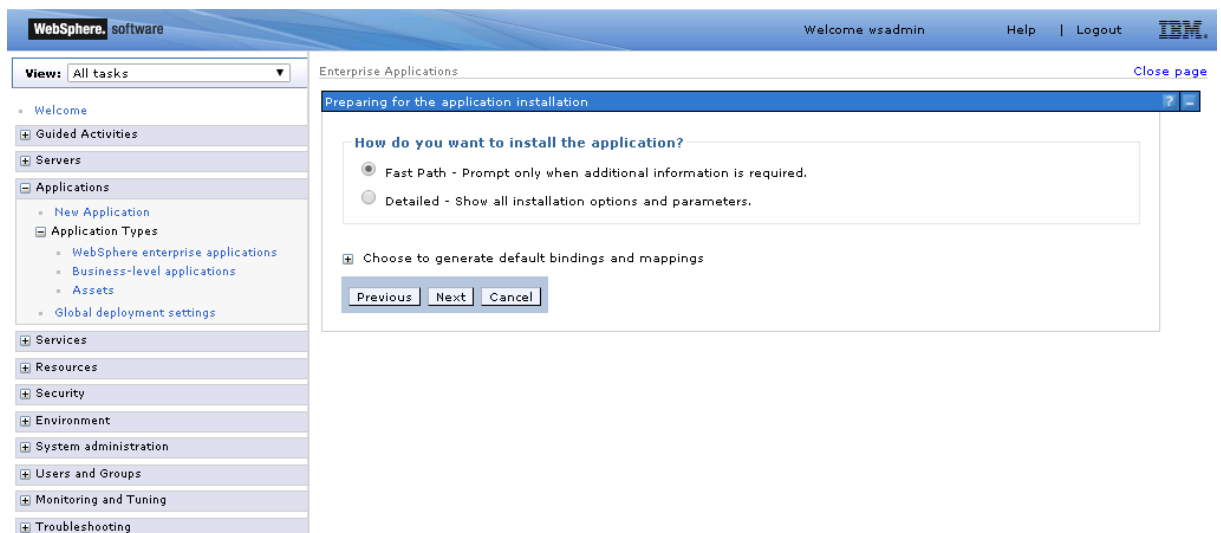
3. Allez dans le menu *Applications*, et cliquez sur *Nouvelle application*.
4. Cliquez sur *Nouvelle application d'entreprise*



5. Sélectionnez *Système de fichier local* et cliquez sur *Parcourir....* Aller dans le bon répertoire pour sélectionner `jcms-10.0.0.war`. Puis, cliquez sur *Suivant*.



6. Sélectionnez *Raccourci* - Ne demander que si des informations supplémentaires sont requises. Puis, cliquez sur *Suivant*.





7. Cochez le choix *Activation du rechargement de la classe* et entrez 2 dans *Intervalle de rechargement en secondes*.  
(Optionnel : renseignez le champ *Nom de l'application*, e.g : JPlatform10)  
Puis, cliquez sur *Suivant*.

The screenshot shows the 'Install New Application' wizard in the WebSphere Administration Console. The left sidebar contains a navigation tree with 'Applications' > 'New Application' selected. The main panel is titled 'Specify options for installing enterprise applications and modules.' and shows 'Step 1: Select installation options'. The 'Select installation options' section includes several checkboxes and input fields:

- ☐ Precompile JavaServer Pages files
- Directory to install application: [empty text box]
- ☒ Distribute application
- ☐ Use Binary Configuration
- ☐ Deploy enterprise beans
- Application name: JCMS10
- ☒ Create MBeans for resources
- ☐ Override class reloading settings for Web and EJB modules
- Reload interval in seconds: 2
- ☐ Deploy Web services
- Validate Input off/warn/fail: warn
- ☐ Process embedded configuration
- File Permission: Allow all files to be read but not written to, Allow executables to execute, Allow HTML and image files to be read by everyone
- Application Build ID: Unknown
- ☐ Allow dispatching includes to remote resources
- ☐ Allow servicing includes from remote resources
- Business level application name: Create New BLA

8. Cliquez sur *Suivant*.

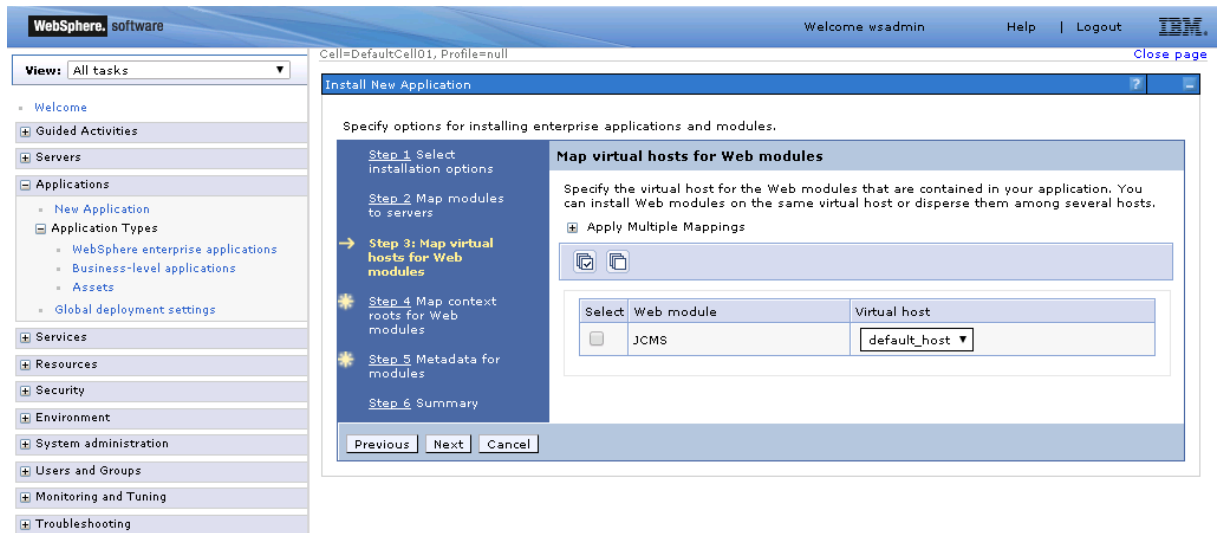
The screenshot shows the 'Install New Application' wizard in the WebSphere Administration Console, now at 'Step 2: Map modules to servers'. The main panel is titled 'Specify options for installing enterprise applications and modules.' and shows 'Step 2: Map modules to servers'. The 'Map modules to servers' section includes a text box for 'Clusters and servers' and a table for mapping modules to servers.

Clusters and servers: WebSphere:cell=DefaultCell01,node=DefaultNode01,server=server1 [Apply]

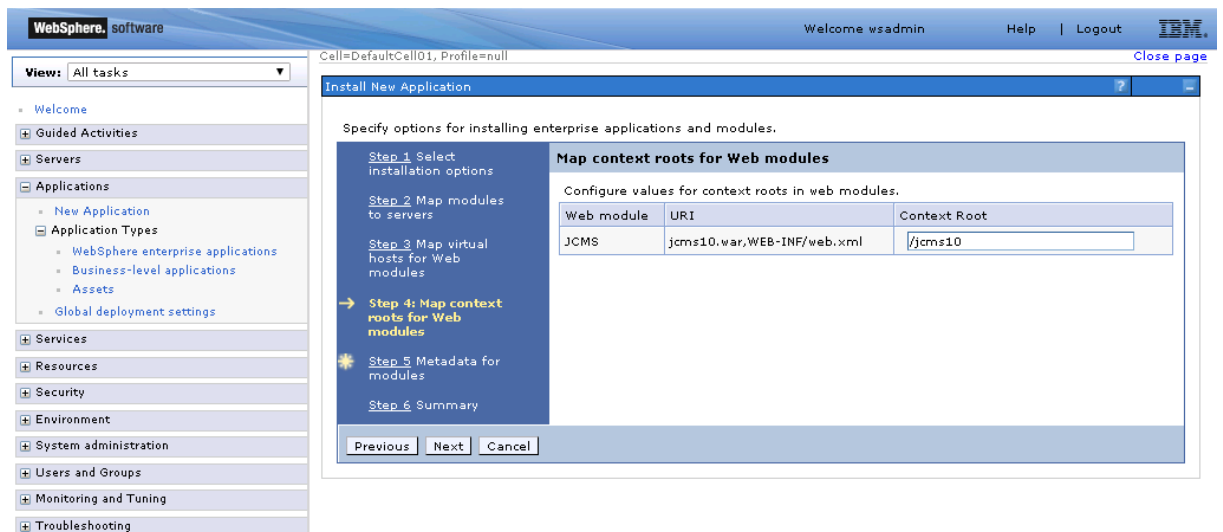
Select	Module	URI	Server
<input type="checkbox"/>	JCMS	jcms10.war,WEB-INF/web.xml	WebSphere:cell=DefaultCell01,node=DefaultNode01,server=server1

Navigation buttons: Previous, Next, Cancel

9. Cliquez sur *Suivant*.



10. Dans le champ *Racine du contexte*, taper l'URI d'accès de la webapp (p. ex. : /jcms10).  
Puis, cliquez sur *Suivant*.



11. A l'étape 5, cliquez sur *Suivant*

The screenshot shows the 'Install New Application' wizard in the WebSphere Administration Console. The left sidebar contains a tree view with 'Applications' expanded, showing 'New Application' and 'Application Types'. The main panel is titled 'Install New Application' and contains a progress bar with steps 1 through 6. Step 5, 'Metadata for modules', is the current step. It includes a description of the 'metadata-complete' attribute and a table for module metadata.

Module	URI	metadata-complete attribute
JCMS	jcms10.war,WEB-INF/web.xml	<input type="checkbox"/>

12. Vérifiez le récapitulatif et cliquez sur *Terminer*.

The screenshot shows the 'Install New Application' wizard in the WebSphere Administration Console, Step 6: Summary. The left sidebar is the same as in the previous screenshot. The main panel is titled 'Install New Application' and contains a progress bar with steps 1 through 6. Step 6, 'Summary', is the current step. It includes a description of the summary and a table of installation options.

Options	Values
Precompile JavaServer Pages files	No
Directory to install application	
Distribute application	Yes
Use Binary Configuration	No
Deploy enterprise beans	No
Application name	JCMS10
Create MBeans for resources	Yes
Override class reloading settings for Web and EJB modules	No
Reload interval in seconds	2
Deploy Web services	No
Validate Input off/warn/fail	warn
Process embedded configuration	No
File Permission	.*\,dl=755#.*\,so=755#.*\,a=755#.*\,sl=755
Application Build ID	Unknown
Allow dispatching includes to remote resources	No
Allow servicing includes from remote resources	No
Business level application name	
Asynchronous Request Dispatch Type	Disabled
Allow EJB reference targets to resolve automatically	No
Deploy client modules	No
Client deployment mode	Isolated
Validate schema	No
Cell/Node/Server	<a href="#">Click here</a>

13. Après l'installation, cliquez sur *Sauvegarder*.

The screenshot displays the WebSphere software administration console interface. The top navigation bar includes the 'WebSphere software' logo, a 'Welcome wsadmin' message, and links for 'Help' and 'Logout'. The left sidebar contains a tree view of administrative tasks, with 'Applications' expanded to show 'New Application' and 'Application Types'. The main content area is titled 'Installing...' and displays a series of status messages from the deployment manager, including resource validation, configuration updates, and the successful installation of JCMS10. A blue banner at the bottom of the messages states 'Application JCMS10 installed successfully.' Below this, instructions are provided on how to start the application and save changes to the master configuration. A 'Manage Applications' link is visible at the bottom of the console.

Views: All tasks

Installing...

If there are enterprise beans in the application, the EJB deployment process can take several minutes. Do not save the configuration until the process completes.

Check the SystemOut.log on the deployment manager or server where the application is deployed for specific information about the EJB deployment process as it occurs.

ADMA5018: Installation of JCMS10 started.

ADMA5067: Resource validation for application JCMS10 completed successfully.

ADMA5058: Application and module versions are validated with versions of deployment targets.

ADMA5005: The application JCMS10 is configured in the WebSphere Application Server repository.

ADMA5005: The application JCMS10 is configured in the WebSphere Application Server repository.

ADMA5081: The bootstrap address for client module is configured in the WebSphere Application Server repository.

ADMA5053: The library references for the installed optional package are created.

ADMA5005: The application JCMS10 is configured in the WebSphere Application Server repository.

ADMA5001: The application binaries are saved in /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/workspace/cells/DefaultCell01/applications/JCMS10.ear/JCMS10.ear

ADMA5005: The application JCMS10 is configured in the WebSphere Application Server repository.

SECJ0400: Successfully updated the application JCMS10 with the appContextIDForSecurity information.

ADMA5005: The application JCMS10 is configured in the WebSphere Application Server repository.

ADMA5005: The application JCMS10 is configured in the WebSphere Application Server repository.

ADMA5113: Activation plan created successfully.

ADMA5011: The cleanup of the temp directory for application JCMS10 is complete.

ADMA5013: Application JCMS10 installed successfully.

Application JCMS10 installed successfully.

To start the application, first save changes to the master configuration.

Changes have been made to your local configuration. You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

To work with installed applications, click the "Manage Applications" link.

[Manage Applications](#)

### 5.3.2 Configuration des bibliothèques partagées

Il faut ensuite configurer WebSphere AS pour qu'il utilise la version des librairies Derby et Jackson livrée avec JPlatform 10.0.

1. Pour cela, dans le menu *Environnement* > *Bibliothèque partagée*, choisissez la portée *Cellule* puis cliquez sur *Nouveau*.
2. Saisissez *JcmsLibs* dans *Nom* et dans *Chemin d'accès aux classes* saisissez les chemins d'accès aux librairies suivantes fournies avec JPlatform ;

- WEB-INF/lib/derby-10.13.1.1.jar (*derby-10.14.2.0.jar* à partir de *JPlatform 10 SP2*)
- WEB-INF/lib/derbyclient-10.13.1.1.jar (*derbyclient-10.14.2.0.jar*)
- WEB-INF/lib/derbynet-10.13.1.1.jar (*derbynet-10.14.2.0.jar*)
- WEB-INF/lib/jackson-core-lgpl-1.9.12.jar
- WEB-INF/lib/jackson-mapper-lgpl-1.9.12.jar

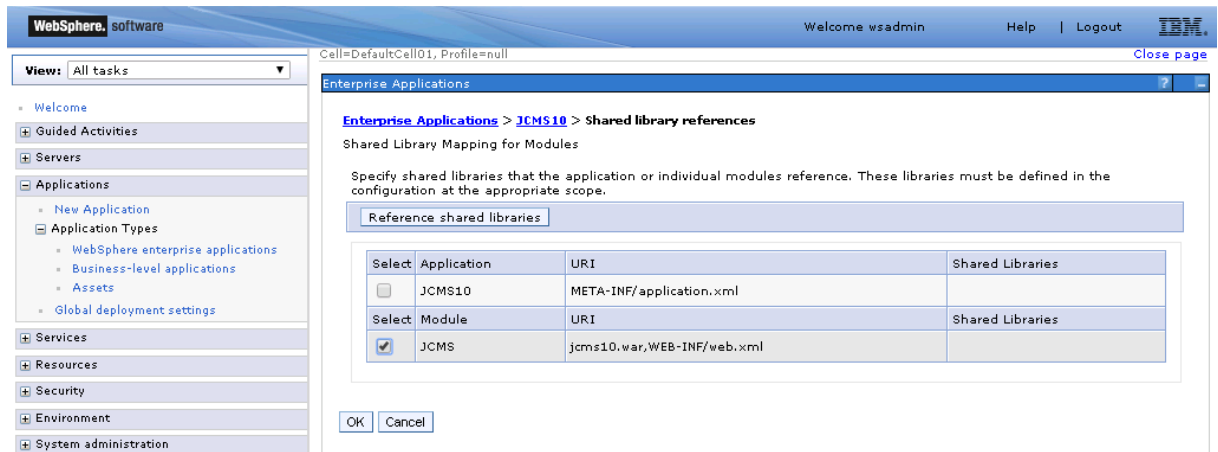
**Attention !** les chemins doivent être absolus, ne doivent pas contenir d'espaces et, sous Windows, les caractères "\" (backslash) doivent être doublés. Saisissez un chemin par ligne. Cochez la case *Utilisez un chargeur de classe isolé pour cette bibliothèque partagée*. Cliquez sur *OK*

*Note : si JPlatform a déjà été démarré une première fois dans votre serveur d'application WebSphere AS, sans cette configuration, la prise en compte des bibliothèques partagées nécessite un redémarrage complet du serveur WebSphere AS (le redémarrage de l'applicatif JPlatform ne suffit pas)*

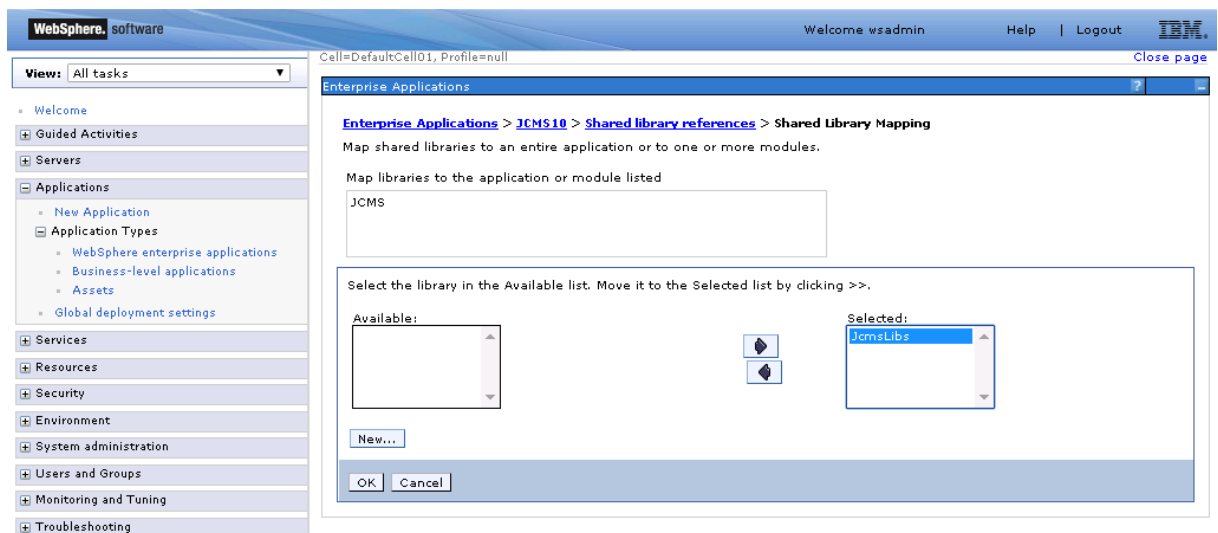
The screenshot shows the WebSphere AS Administration Console interface. On the left is a navigation tree with categories like Welcome, Guided Activities, Servers, Applications, Services, Resources, Security, Environment, System administration, Users and Groups, Monitoring and Tuning, Troubleshooting, Service integration, and UDDI. The 'Applications' category is expanded, showing 'New Application', 'Application Types', 'WebSphere enterprise applications', 'Business-level applications', 'Assets', and 'Global deployment settings'. The 'Shared Libraries' link is highlighted. The main panel displays the configuration for a shared library named 'JcmsLibs'. The 'Scope' is set to 'cells:DefaultCell01:nodes:DefaultNode01:servers:server1'. The 'Name' is 'JcmsLibs'. The 'Description' field is empty. The 'Classpath' field contains a list of absolute paths to the required JAR files: /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/installedApps/DefaultCell01/JCMS10.ear/jcms10.war/WEB-INF/lib/derby-10.13.1.1.jar, /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/installedApps/DefaultCell01/JCMS10.ear/jcms10.war/WEB-INF/lib/derbyclient-10.13.1.1.jar, /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/installedApps/DefaultCell01/JCMS10.ear/jcms10.war/WEB-INF/lib/derbynet-10.13.1.1.jar, /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/installedApps/DefaultCell01/JCMS10.ear/jcms10.war/WEB-INF/lib/jackson-core-lgpl-1.9.12.jar, and /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/installedApps/DefaultCell01/JCMS10.ear/jcms10.war/WEB-INF/lib/jackson-mapper-lgpl-1.9.12.jar. The 'Native Library Path' field is empty. Under the 'Class Loading' section, the checkbox 'Use an isolated class loader for this shared library' is checked. At the bottom are buttons for 'Apply', 'OK', 'Reset', and 'Cancel'.

## Installation de JPlatform sur le serveur d'application

3. Sauvegardez la configuration.
4. Ouvrez le menu *Applications > Types d'application > Applications d'entreprise WebSphere* et cliquez sur l'application JPlatform.
5. Dans la section *Référence*, choisissez l'option *Références de bibliothèque partagée*.
6. Cochez la case de sélection pour l'application JPlatform et cliquez sur le bouton *Bibliothèque partagées de référence* et cliquez sur *OK*.



7. Sélectionnez la librairie partagée *JcmsLibs* créée à l'étape précédente dans la colonne de gauche, et ajoutez-la aux bibliothèques sélectionnées avec le bouton >> (flèche droite).



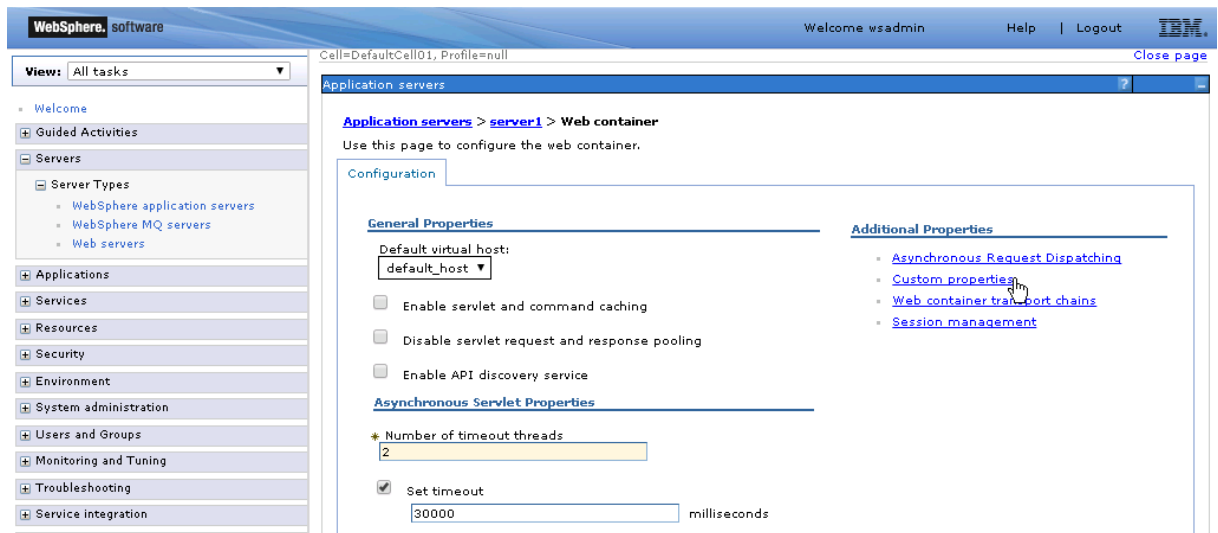
8. Cliquez sur OK sur les 2 écrans suivants.
9. Sauvegardez la configuration

## 5.3.3 Configuration du serveur d'application

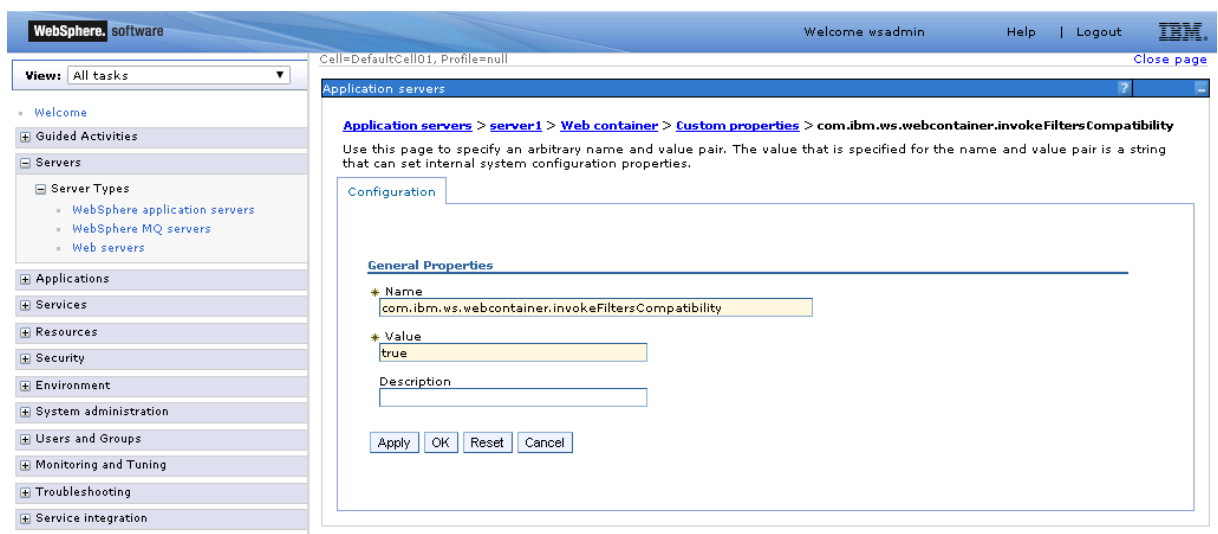
### Configuration des Servlets Filter

Il faut ensuite configurer WebSphere AS pour qu'il invoque les *Servlets Filter* de JPlatform dans toutes les circonstances.

1. Ouvrez le menu *Serveurs > Types de serveurs > Serveurs d'applications WebSphere* et sélectionnez votre serveur.
2. Dans la section *Paramètres du conteneur Web > Paramètres du conteneur Web*, choisissez l'option *Conteneur Web*.



3. Cliquez sur *Propriétés personnalisées*, puis sur *Nouveau*. Saisissez `com.ibm.ws.webcontainer.invokeFiltersCompatibility` dans *Nom* et `true` dans *Valeur*. Cliquez sur *OK* puis *Sauvegarder*.

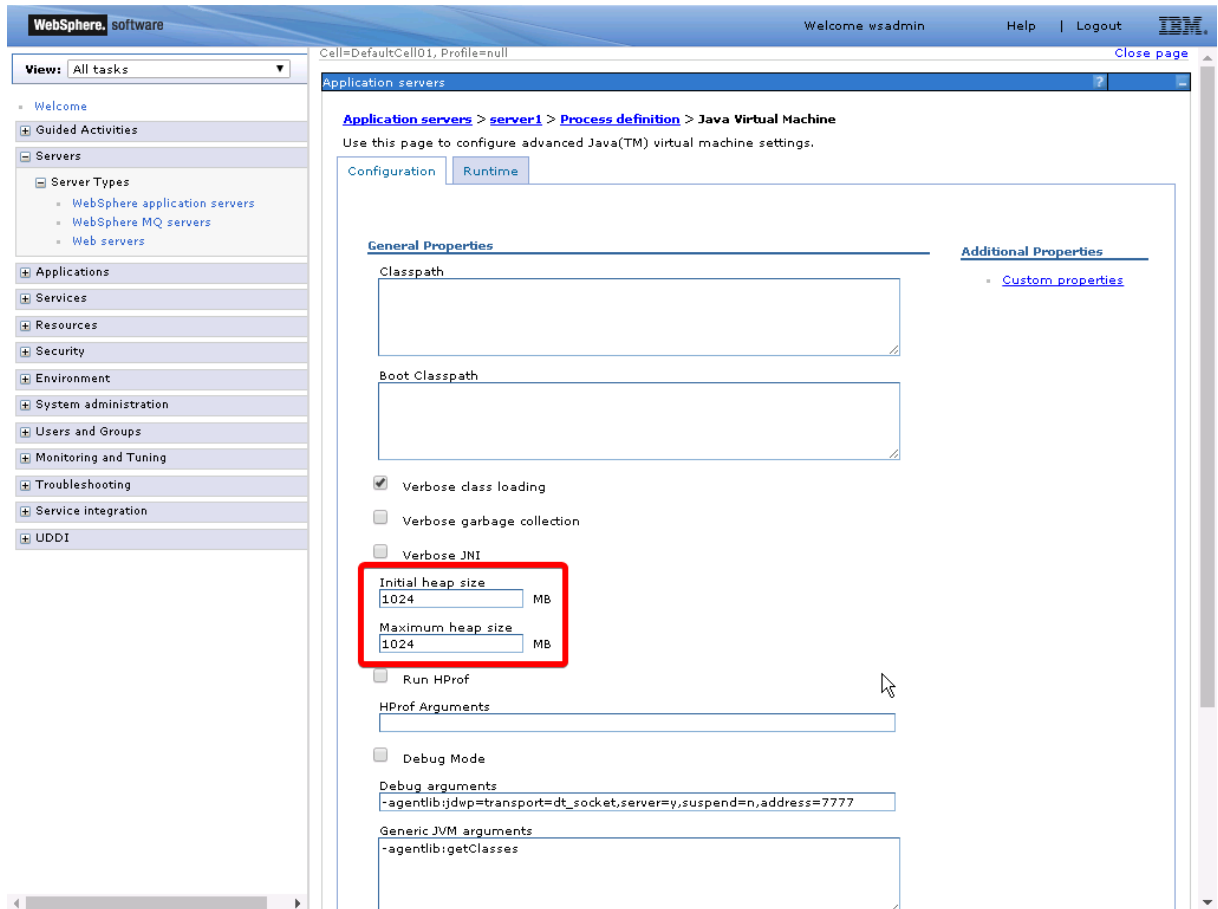


## Configuration de l'allocation mémoire

## Installation de JPlatform sur le serveur d'application

Il est nécessaire de configurer l'allocation mémoire de la JVM (1024MB recommandé).

1. Dans la console d'administration accédez à *Serveurs > Types de serveurs > Serveurs d'applications WebSphere*  
Cliquez sur le serveur à configurer,
2. Dans la section *Infrastructure du serveur*, Ouvrez *Gestion des processus et Java > Définition des processus*
3. Enfin, dans la section *Propriétés supplémentaires* cliquez sur *Machine virtuelle Java*  
Saisissez 1024 dans les champs *Taille de tas initiale* et *Taille de tas maximale*



4. Cliquez sur OK et sauvegardez la configuration

Pour plus de détails référez-vous à la documentation de WebSphere AS :  
[http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/topic/com.ibm.websphere.express.doc/ae/tprf\\_tunejvm\\_v61.html](http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/topic/com.ibm.websphere.express.doc/ae/tprf_tunejvm_v61.html)



### Configuration de la sécurité des cookies de sessions

Afin de se prémunir contre le vol de cookie via JavaScript, configurez WebSphere AS pour que le cookie de session Java EE (JSESSIONID) positionne l'attribut `HttpOnly`. Pour cela, positionnez la propriété suivante pour protéger tous les cookies émis par l'application :

```
com.ibm.ws.webcontainer.httpOnlyCookies=*
```

La procédure de configuration et les version de WebSphere AS requise sont indiqués dans cet avertissement de sécurité :

<http://www-01.ibm.com/support/docview.wss?uid=swg1PK98436>

La documentation détaillée de cette propriété est disponible à l'adresse suivante (recherchez `httpOnlyCookies`) :

[https://www.ibm.com/support/knowledgecenter/en/SSAW57\\_9.0.0/com.ibm.websphere.nd.multiplatform.doc/ae/rweb\\_custom\\_props.html](https://www.ibm.com/support/knowledgecenter/en/SSAW57_9.0.0/com.ibm.websphere.nd.multiplatform.doc/ae/rweb_custom_props.html)

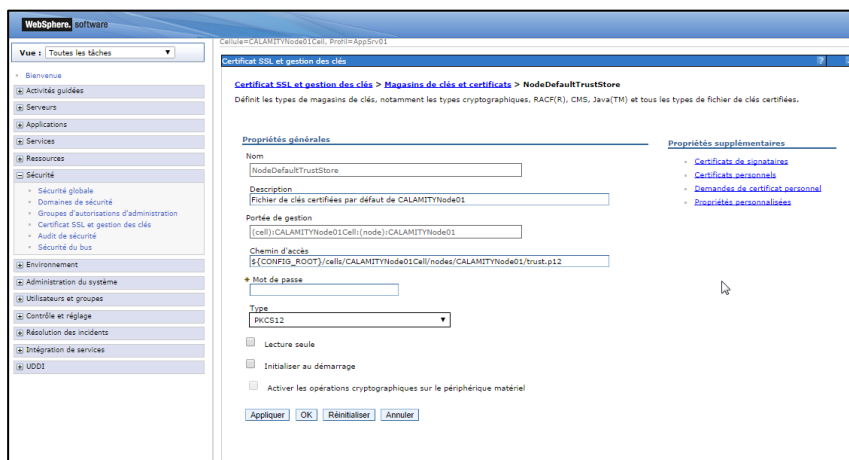
### Import des certificats

Si votre application JPlatform doit effectuer des requêtes HTTPS sur un serveur web distant, il est nécessaire de réaliser l'import des certificats intermédiaires du serveur distant. C'est le cas de la webapp JPlatform 10 par défaut qui contient une Portlet RSS vers le flux des actualités Jalios sur <https://www.jalios.com/>

Sans cette manipulation, le message suivant peut apparaître dans les logs de JPlatform et de WebSphere AS :

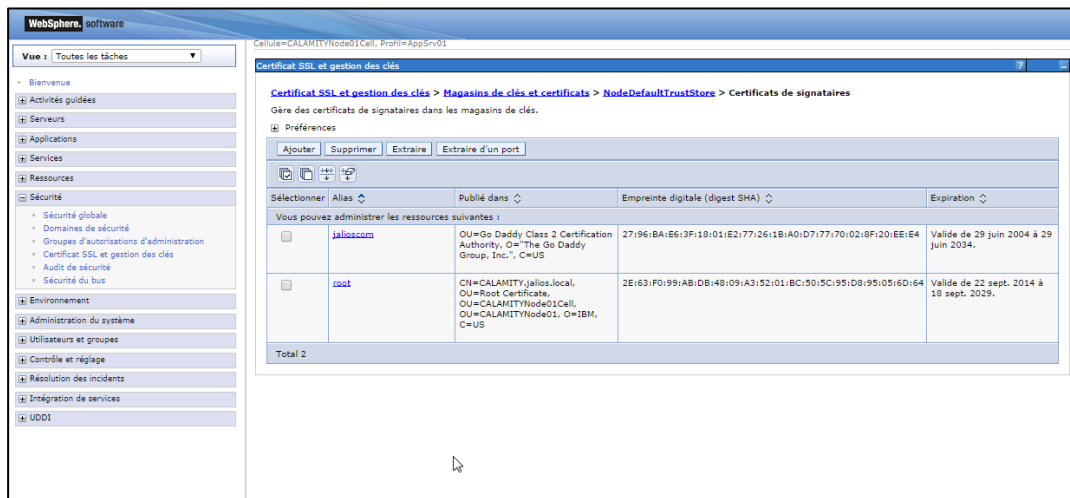
```
javax.net.ssl.SSLHandshakeException: com.ibm.jsse2.util.j: PKIX path building failed:  
java.security.cert.CertPathBuilderException: PKIXCertPathBuilderImpl could not build a valid CertPath.;
```

1. Dans la console d'administration accéder à *Sécurité -> Certificat SSL et gestion des clés -> Magasins de clés et certificats -> NodeDefaultTrustStore*

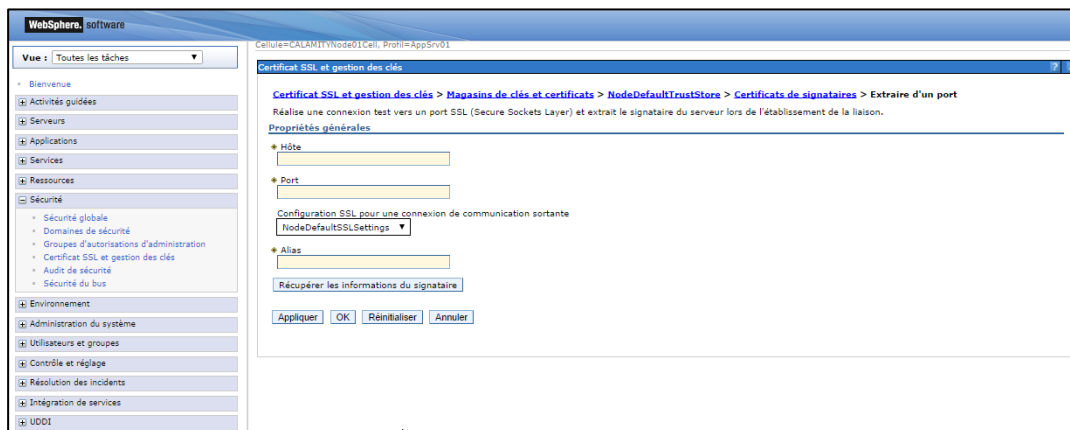


2. Dans la section *Propriétés supplémentaires*, cliquez sur le lien *Certificats de signataires*

Cliquez sur le bouton *Extraire d'un port*,

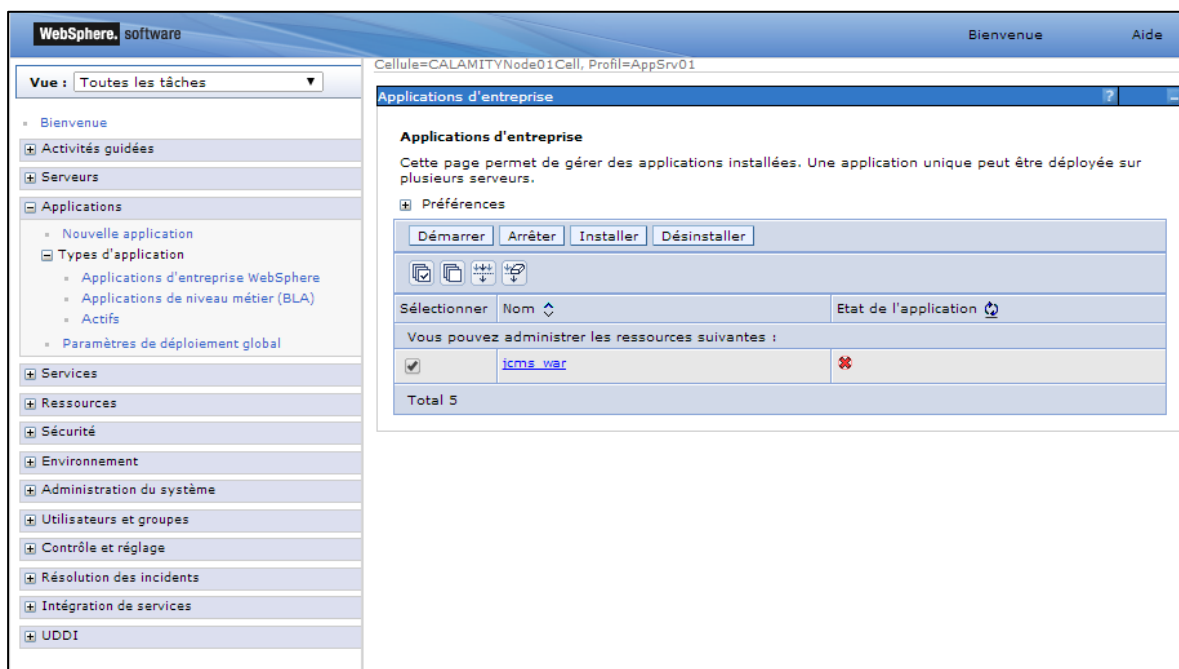


- Entrez les informations sur le site distant :  
Le nom de domaine du site distant, par exemple `www.jalios.com`  
Le port, généralement le port 443  
Choisissez un alias, par exemple 'jalioscom'  
Cliquez sur le bouton *Appliquer*



### 5.3.4 Démarrage de l'application

Pour démarrer la webapp, aller dans le menu *Applications > Types d'application > Applications d'entreprise WebSphere*. Cochez la case à côté de `jcms.war`, puis cliquez sur *Démarrer*.



JPlatform est maintenant prêt à fonctionner.

### 5.4 Installation sur JBoss EAP 7.0, 7.1, 7.2 ou 7.3

#### Déploiement :

- Décompressez le war dans un nouveau répertoire  
EAP-7.3\standalone\deployments\jcms10.war\
  - Créez un fichier vide avec le nom  
EAP-7.3\standalone\deployments\jcms10.war.dodeploy  
Ce fichier est utilisé par JBoss EAP au premier déploiement
- Configuration du deployment-timeout :  
Pour permettre le déploiement d'une application JPlatform, il est nécessaire d'augmenter le paramètre deployment-timeout de JBoss EAP.
  - Editez le fichier EAP-7.3\standalone\configuration\standalone.xml et ajoutez l'attribut deployment-timeout à 1200 secondes

```
<deployment-scanner scan-interval="5000" relative-to="jboss.server.base.dir" path="deployments" deployment-timeout="1200" />
```

- Configuration de tools.jar :  
JPlatform requiert la librairie tools.jar fournie dans la JDK.  
Il est nécessaire de créer un module JBoss EAP pour permettre à l'applicatif JPlatform de l'utiliser.
  - Créez le fichier jboss-deployment-structure.xml dans jcms10.war\WEB-INF\ :

```
<?xml version="1.0" encoding="UTF-8"?>
<jboss-deployment-structure>
  <deployment>
    <exclusions>
      <module name="com.fasterxml.jackson.core.jackson-core" />
      <module name="com.fasterxml.jackson.core.jackson-annotations" />
      <module name="com.fasterxml.jackson.core.jackson-databind" />
      <module name="com.fasterxml.jackson.jaxrs.jackson-jaxrs-json-provider" />
      <module name="org.jboss.resteasy.resteasy-jackson2-provider" />
      <module name="org.jboss.resteasy.resteasy-jackson-provider" />
    </exclusions>
    <dependencies>
      <module name="com.sun.tools.javac" />
    </dependencies>
  </deployment>
</jboss-deployment-structure>
```

- Créez le répertoire EAP-7.3\modules\com\sun\tools\javac\main
- Recopiez le fichier JDK\lib\tools.jar de votre JDK dans ce répertoire
- Créez le fichier module.xml dans ce répertoire

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="com.sun.tools.javac">
  <resources>
    <resource-root path="tools.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <system export="true">
      <paths>
        <path name="com/sun/tools/javac"/>
      </paths>
    </system>
  </dependencies>
</module>
```

- **Configuration UTF-8**

Appliquer la configuration indiquée dans la documentation suivante :

How to globally set the uri-encoding to UTF-8 in JBoss EAP 7 ?

<https://access.redhat.com/solutions/3325351>

- **Configuration de derby :**

JPlatform est livré avec le SGBDR embarqué Derby. Les jars de Derby livrés dans la webapp déclenchent automatiquement l'instanciation des drivers JDBC 4 Derby au démarrage de JBoss, avant même le démarrage de JPlatform. Ainsi sans configuration spécifique, le répertoire de démarrage de JBoss est utilisé comme répertoire de stockage des bases de données derby.

Si vous souhaitez utiliser Derby, il est nécessaire d'ajouter une propriété système dans le fichier EAP-7.3\standalone\configuration\standalone.xml pour forcer le répertoire home de derby à celui attendu par JPlatform (répertoire WEB-INF\data\derby dans la webapp).

```
<extensions>
    ....
</extensions>
<system-properties>
    <property name="derby.system.home"
        value="C:\Java\jboss-eap-7.3\standalone\deployments\jcms10.war\WEB-INF\data\derby"/>
</system-properties>
```

### Démarrage et premier accès :

- Démarrez JBoss EAP :
  - Sur Windows avec EAP-7.3\bin\standalone.bat
  - Sur Unix avec EAP-7.3\bin\standalone.sh
- Patientez pendant le démarrage de JPlatform  
Si vous le souhaitez, consultez les logs de démarrage dans EAP-7.3\standalone\deployments\jcms10.war\WEB-INF\data\logs\jcms.log  
En raison des générations de classes Java effectuées par JPlatform, il peut être nécessaire de procéder à 2 démarrages successif de JBoss avant de pouvoir accéder à l'application.
- JPlatform est accessible à l'adresse <http://localhost:8080/jcms10/>

### Configuration de l'allocation mémoire

Il est nécessaire de configurer l'allocation mémoire de la JVM (1024MB recommandé). Pour cela référez-vous à la documentation de JBoss.

- [https://access.redhat.com/documentation/en-us/red\\_hat\\_jboss\\_enterprise\\_application\\_platform/7.3/html/configuration\\_guide/configuring\\_jvm\\_settings](https://access.redhat.com/documentation/en-us/red_hat_jboss_enterprise_application_platform/7.3/html/configuration_guide/configuring_jvm_settings)
- [https://access.redhat.com/documentation/en-US/JBoss\\_Enterprise\\_Application\\_Platform/5/html/Performance\\_Tuning\\_Guide/chap-Performance\\_Tuning\\_Guide-Java\\_Virtual\\_Machine\\_Tuning.html](https://access.redhat.com/documentation/en-US/JBoss_Enterprise_Application_Platform/5/html/Performance_Tuning_Guide/chap-Performance_Tuning_Guide-Java_Virtual_Machine_Tuning.html)

### Configurations avancées

- Nombre de paramètres autorisés :  
Pour le traitement de certaines opérations utilisant de très nombreux paramètres (comme l'ajout de données dans la panier), il est nécessaire d'ajouter une propriété système dans le fichier EAP-7.3\standalone\configuration\standalone.xml

```
<extensions>
....
</extensions>
<system-properties>
  <property name="org.apache.tomcat.util.http.Parameters.MAX_COUNT"
    value="5000"/>
</system-properties>
```
- Dépôt de fichier :  
Pour autoriser le dépôt de fichier de taille conséquente, il est nécessaire de configurer l'attribut max-post-size dans le même fichier de configuration standalone.xml

```
<http-listener ... max-post-size="2147483647" />
```

[https://access.redhat.com/documentation/en-us/red\\_hat\\_jboss\\_enterprise\\_application\\_platform/7.3/html/configuration\\_guide/configuring\\_the\\_web\\_server undertow](https://access.redhat.com/documentation/en-us/red_hat_jboss_enterprise_application_platform/7.3/html/configuration_guide/configuring_the_web_server undertow)

### Configuration

Pour toute autre configuration, référez-vous à la documentation officielle de JBoss EAP 7 :

- [https://access.redhat.com/documentation/en-us/red\\_hat\\_jboss\\_enterprise\\_application\\_platform/7.3/](https://access.redhat.com/documentation/en-us/red_hat_jboss_enterprise_application_platform/7.3/)
- [https://access.redhat.com/documentation/en-us/red\\_hat\\_jboss\\_enterprise\\_application\\_platform/7.3/html-single/configuration\\_guide/index](https://access.redhat.com/documentation/en-us/red_hat_jboss_enterprise_application_platform/7.3/html-single/configuration_guide/index)

## 6 Démarrage et paramétrage de JPlatform

### 6.1 Démarrage de l'application

Une fois JPlatform 10.0 installé et le serveur d'application démarré, allez dans un navigateur et consultez la page d'état de JPlatform :

`http://localhost:<port>/jcms10/admin/status.jsp`

La liste ci-dessous récapitule les ports par défaut des différents serveurs d'application

- Apache Tomcat : 8080
- IBM WebSphere AS : 9080
- JBoss EAP : 8080

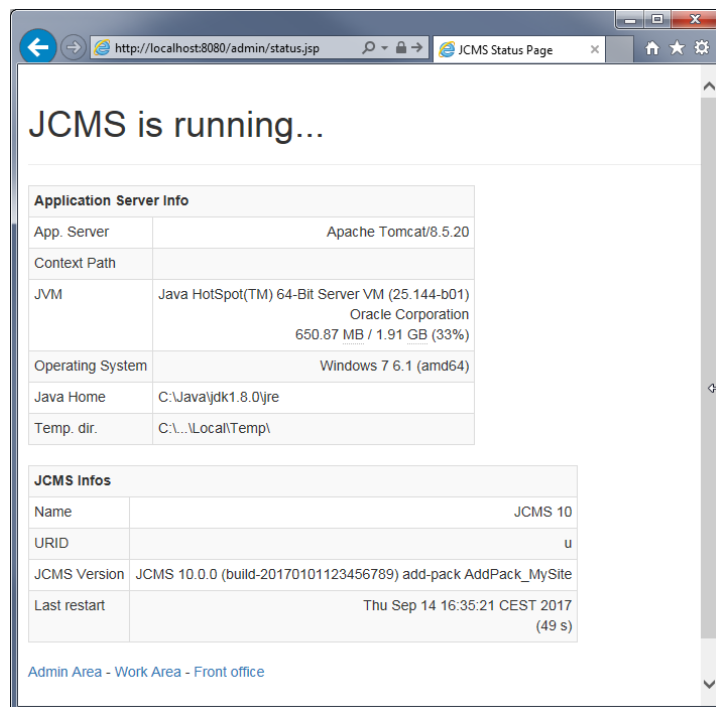


Figure 2 - Page d'état de JPlatform 10.0

Vérifiez que le site est bien démarré (le tableau *JCMS Infos* doit apparaître sous le tableau *Application Server Info*).

Si aucune erreur n'est affichée, cliquez sur le lien *Front office* pour accéder à la page d'accueil par défaut de JPlatform.





Figure 3 - Page d'accueil par défaut de JPlatform 10.0

Cette page comporte une zone d'identification.

Identifiez-vous comme administrateur (identifiant : *admin*, mot de passe : *admin*).

### 6.2 Modification du mot de passe administrateur

Une fois authentifié en tant qu'administrateur, cliquez sur l'icône  puis sur le lien  *Espace d'administration*.

Dans la section *Utilisateurs*, cliquez ensuite sur le lien *Membres*.



Dans la liste des membres, cliquez sur l'icône d'édition situé à côté du membre « Admin » pour ouvrir le menu contextuel et choisissez *Modifier*.

Saisissez le mot de passe actuel (*admin*), et entrez un nouveau mot de passe sécurisé.  
Cliquez sur Enregistrer.

Appliquez les recommandations de l'ANSSI relatives à la sécurité des mots de passe :  
<https://www.ssi.gouv.fr/guide/mot-de-passe/>

### 6.3 Configuration initiale de l'application

Cette section décrit les principales propriétés à modifier pour une configuration initiale. Pour plus de détails, consultez le Manuel utilisateur de JPlatform sur Jalios Community (<https://community.jalios.com/documentations/manuels>).

Une fois authentifié en tant qu'administrateur, cliquez sur l'icône  puis sur le lien  *Espace d'administration*.

Dans la section *Exploitation*, cliquez ensuite sur le lien *Propriétés*.



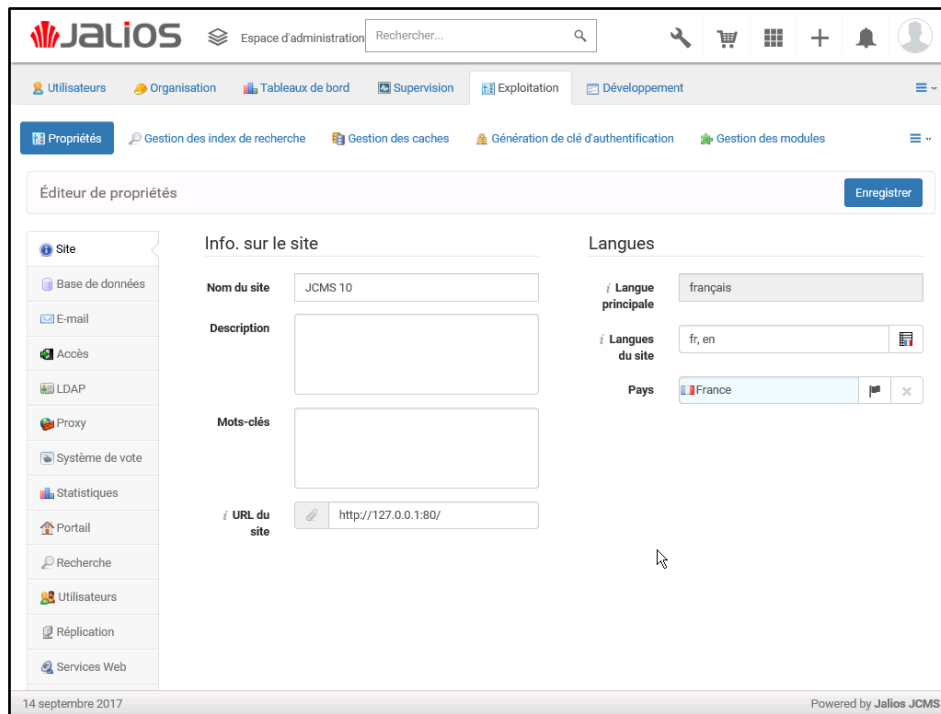


Figure 4 - Editeur de propriétés de JPlatform 10.0

#### Onglet Site > Info sur le site

- *Nom du site* : entrez le nom du site ;
- *URL de base*: entrez l'URL d'accès de la webapp (p. ex., <http://localhost:8080/jcms10/> pour un site local ) ;

#### Onglet Site >Langues

- *Langue principale* : choisissez la langue principale du site. Attention : JPlatform n'assure pas la conversion des contenus lorsque la langue principale change. Veillez donc à choisir la langue principale du site avant la saisie des contenus et ne plus la changer par la suite.
- *Langue du site* : sélectionnez la liste des langues du site. Les options de multilinguisme s'activent dès qu'il existe au moins deux langues. L'ordre des langues sera respecté dans les interfaces du front-office.

## 6.4 Configuration de la base de données

JPlatform est pré-paramétré pour utiliser le SGBDR embarqué Derby.  
Pour utiliser un SGBDR externe suivez cette procédure :

- Téléchargez sur le site support le module correspondant à votre SGBDR
  - <https://community.jalios.com/plugin>
- Installez dans JPlatform ce module et redémarrez JPlatform

- Sur le SGBDR :
  - Créez une base de données nommée, par exemple, *jcmsdb*.

La base doit être créée avec un encodage supportant tous les caractères UTF-8. Consultez la fiche du module de base de données retenu pour plus de détails.
  - Créez un utilisateur nommé, par exemple, *jcms*. Cet utilisateur doit avoir l'ensemble des droits sur la base *jcmsdb*.
- Sur JPlatform :
  - Déclaration via une URL JDBC
    - Allez dans l'onglet Base de données, sélectionnez votre SGBDR dans la liste déroulante.
    - Indiquez l'URL JDBC de connexion ainsi que le compte et le mot de passe de l'utilisateur *jcms*.
    - Cliquez sur *Vérifier la connexion*
  - Déclaration via une DataSource
    - Indiquez la DataSource
    - Cliquez sur *Vérifier la connexion*
    - Attention ! Si la DataSource est utilisée, le pool de connexion (généralement c3p0) est systématiquement désactivé. La définition du pool doit se faire dans la déclaration de la DataSource.
  - Redémarrez JPlatform.
  - Au premier démarrage, JPlatform crée dans la base *jcmsdb* l'ensemble des tables et des indexes nécessaire à son fonctionnement.
  - Vérifiez la bonne configuration en allant dans *Espace d'administration > Supervision > Etat de la base de données*.

## 6.5 Configuration E-mail

### 6.5.1 Configuration du mail sortant

JPlatform communique avec le serveur de messagerie e-mail pour envoyer des mails aux utilisateurs. Ces mails sont émis par différentes fonctions de JPlatform, telles que le workflow, la notification des nouveautés, l'envoi de publication par mail, les alerte aux administrateurs, ainsi que par certains modules (Newsletter, Forum, ESN, ...)

#### Onglet E-mail > E-mail sortant

- *Activé* : détermine si l'envoi d'e-mail est activé ;
- *Serveur SMTP* : adresse du serveur SMTP ;
- *E-mail par défaut* : e-mail par défaut du site.

### 6.5.2 Configuration du mail entrant

JPlatform peut aussi utiliser recevoir des mails. Ce vecteur de communication avec JPlatform peut être utilisé soit pour archiver des mails soit pour déclencher des actions consécutives à la réception d'un mail (p. ex. réponse à une discussion de forum, changement de son statut ESN, ...) La configuration du mail entrant nécessite de disposer d'une ou plusieurs boîtes aux lettres dédiées à JPlatform.

JPlatform est compatible avec les serveurs POP3 et IMAP. Ces serveurs doivent être configurés pour rediriger tous les mails `*@example.com` dans une boîte mail dédié. Cette redirection est nécessaire pour supporter les adresses emails dynamiques.

Exemple:

- `dbforum@example.com => jcms@example.com`
- `archive+124df6s5@example.com => jcms@example.com`
- `espace1@example.com => jcms@example.com`

JPlatform analyse :

- l'attribut `from` pour déterminer le membre émetteur du mail
- et les attributs `to`, `cc`, `bcc` pour déterminer la tâche à effectuer.

Les préfixe et suffixe d'adresse sont utilisés lors de l'envoi de mail nécessitant une réponse par JPlatform. Pour une plus grande homogénéité, ces valeurs devraient être cohérentes avec les autres adresses déclarées dans JPlatform (administrateur, espace de travail, adresse par défaut)

Les mails entrants dont l'adresse correspond à l'expression régulière des archives seront archivés dans JcmsDB. Afin d'éviter que n'importe quel mail soit archivé, il est possible de définir des droits d'archivage:

- L'expéditeur doit avoir un compte de Membre dans JPlatform (p. ex. `from:membre@example.com`).
- L'adresse d'expédition doit contenir une clé d'authentification (p. ex. `to:archive+54f78g1@jcms.example.com`).

#### Onglet E-mail > E-mail entrant

- *Activé* : détermine si le mail entrant est activé ;
- *Libellé en préfixe* : libellé mis en préfixe des adresses e-mail pour les réponses ;
- *Suffixe d'adresse* : suffixe ajouté aux adresses e-mail entrant pour les réponses ;
- *Compte e-mail* : nom du compte de messagerie ;
  - *Activé* : détermine si le compte est actif ;
  - *Serveur* : adresse du serveur de messagerie ;
  - *Protocole* : protocole de consultation des mails POP3 ou IMAP ;

- *Identifiant* : identifiant (login) du compte de messagerie ;
- *Mot de passe* : mot de passe du compte mail ;
- *Boîte mail* : le protocole POP3 utilise INBOX par défaut. Les comptes IMAP peuvent spécifier des sous-dossiers.
- *Planification* : détermine la fréquence de consultation du compte de messagerie. Peut-être spécifié soit par une période en minute soit par un ordonnancement au format CRON ;
- *Auteur* : auteur des mails stockés dans JPlatform
- *Espace de travail* : espace de travail d'accueil des mails stockés dans JPlatform;
- *Suppression des mails* : détermine si les mails sont retirés du serveur de mail après avoir été lus.

## 6.6 Configuration LDAP

### 6.6.1 Principes

JPlatform gère de façon autonome les utilisateurs, leurs groupes et leurs droits. Néanmoins, JPlatform peut être couplé avec un annuaire LDAP (ou LDAPS) pour valider les authentifications des membres et récupérer leurs informations (nom, email, ...). Lorsqu'un utilisateur s'authentifie, JPlatform interroge l'annuaire LDAP. Si l'authentification est valide, JPlatform crée un compte pour cette personne lors du premier accès puis effectue l'authentification. Si l'authentification LDAP échoue, JPlatform procède à une authentification avec sa propre base de membres.

JPlatform complète l'intégration avec les annuaires LDAP en synchronisant les groupes LDAP et les utilisateurs qui les composent. La gestion des groupes étant propre à chaque serveur LDAP, JPlatform s'adapte au schéma de l'annuaire. Des pré-configurations sont proposées pour les principaux serveurs LDAP du marché : ActiveDirectory, OpenLDAP et Sun Directory.

Chaque entreprise ayant un schéma LDAP spécifique, le paramétrage de la configuration LDAP nécessite de savoir :

- Quel est le modèle des DN (*Distinguish Name*) des utilisateurs et des groupes ?
- Quel est le modèle des groupes (`posixGroup`, `groupOfUniqueNames`, ...) ?
- Quels champs contiennent les informations sur la personne (compte utilisateur, nom, prénom, email, fonction, ...)

### 6.6.2 Création d'un utilisateur technique dans le LDAP

Il est nécessaire de créer un utilisateur technique permettant à JPlatform de se connecter à l'annuaire LDAP. Créez par exemple l'utilisateur suivant : `uid=jcms,ou=People,dc=example,dc=com`, avec le mot de passe de votre choix.

Cet utilisateur doit disposer des droits de recherche et de consultation dans l'annuaire LDAP. Les droits d'écritures ne sont pas nécessaires, JPlatform ne modifie jamais l'annuaire LDAP.

### 6.6.3 Configuration des propriétés JPlatform

Dans l'onglet *LDAP*, affectez les valeurs suivantes aux différents champs :

### Configuration

- *Actif* : cochez *Oui* ;
- *Serveur* : nom de la machine hébergeant le serveur LDAP ;
- *Port* : port d'écoute du serveur LDAP (389) ou LDAPS (636) ;
- *SSL* : cochez *Non* sauf si vous utilisez un serveur LDAPS ;
- *Identifiant (DN)* : identifiant de l'utilisateur technique (p. ex. : `uid=jcms,ou=People,dc=example,dc=com`)
- *Mot de passe JCMS* : mot de passe de l'utilisateur technique défini ci-dessus.

Cliquez sur *Vérifier la connexion...* pour vérifier que JPlatform arrive à atteindre le serveur LDAP et à s'authentifier avec l'utilisateur technique.

### Schéma LDAP

Choisissez un modèle de schéma LDAP pour pré-remplir les champs Utilisateurs et Groupes

#### Utilisateurs

- *DN de recherche* : DN de base utilisé pour rechercher les utilisateurs dans le LDAP. p. ex. : `ou=People,dc=example,dc=com`
- *Filtre de recherche* : Format de la requête utilisée pour rechercher un utilisateur. La chaîne `{0}` représente la valeur saisie par l'utilisateur dans le champ *Compte utilisateur*. Ce format est dépendant de votre schéma LDAP. Sur ActiveDirectory, le format est généralement `sAMAccountName={0}`. Sur Sun Directory ou OpenLDAP, le format est généralement `uid={0}`.
- *objectClass* : L'object class de votre schéma LDAP qui caractérise un utilisateur (p. ex. `person`, `posixAccount`, `inetOrgPerson`, `organizationalPerson`, ...)
- *Synchroniser les comptes* : Active la synchronisation pour automatiquement créer et mettre à jour les membres en utilisant les informations du LDAP. Désactivez cette propriété empêche la connexion des utilisateurs non existant dans JPlatform. La synchronisation est automatiquement désactivée lors de la récupération d'une copie de travail.
- *Compte utilisateur, Nom, Prénom, ...* : Indiquez pour chacun des champs d'un membre JPlatform, les champs correspondants dans l'annuaire LDAP (si ils existent).

#### Groupes

- *DN de recherche* : DN de base utilisé pour rechercher les groupes dans le LDAP (p. ex. `ou=Groups,dc=example,dc=com`)
- *Filtre de recherche* : requête utilisée pour rechercher un groupe. La chaîne `{0}` représente le DN de l'entrée LDAP (utilisateur ou sous groupe) dont on doit chercher les groupes. Cette requête est dépendante du schéma LDAP. Sur ActiveDirectory, Oracle Directory Server ou OpenLDAP, le format est généralement `(&(objectClass=group)(member={0}))`.

- *objectClass* : L'object classe de votre schéma LDAP qui caractérise un groupe (p. ex. `group`, `posixGroup`, `groupOfNames`, `groupOfUniqueNames`, ...)
- *Synchroniser les groupes* : Active la synchronisation pour automatiquement créer et mettre à jour les groupes d'un membre en utilisant les informations du LDAP. Désactivez cette propriété créera les nouveaux utilisateurs LDAP dans le groupe par défaut. La synchronisation est automatiquement désactivée si la synchronisation des utilisateurs est désactivée.
- *Nom* : indique le champ LDAP contenant le nom du groupe (p. ex. `cn`)
- *Member* : Indiquez le champ LDAP qui définit l'appartenance d'un membre ou d'un sous groupe (p. ex. `member`, `memberUid`, `memberOf`, ...).

Pour plus de détails sur la mise en œuvre, consultez l'article *Configuration de JPlatform avec Open-LDAP (LDAP/LDAPS)* sur Jalios Community (<https://community.jalios.com/howto/ldap>).

## 6.7 Configuration du proxy de sortie

### 6.7.1 Principes

Certaines fonctionnalités de JPlatform (syndication RSS, réplication, ...) effectuent des requêtes HTTP qui peuvent nécessiter de passer par le serveur proxy de l'entreprise. Si c'est le cas, renseignez les champs *Proxy de sortie* de l'onglet *Proxy*.

### 6.7.2 Configuration des propriétés JPlatform

- *Serveur* : indiquez l'adresse du serveur
- *Port* : indiquez le port d'écoute du serveur
- *Compte* : indiquez le compte utilisateur de connexion au serveur proxy si nécessaire
- *Mot de passe* : indiquez le mot de passe de connexion au serveur proxy si nécessaire
- *Sauf pour* : indiquez les adresses pour lesquelles il ne faut pas passer par le proxy. Indiquez notamment les adresses des réplicas afin que les requêtes de synchronisation JSync ne soient pas aiguillées sur le proxy.

## 6.8 Configuration du Reverse Proxy

### 6.8.1 Principes

Un Reverse Proxy est un serveur HTTP (HTTPS) qui reçoit des requêtes et les diffuse sur d'autres serveurs http. Ce type d'architecture permet de masquer les véritables serveurs aux clients http qui ne connaissent que le Reverse Proxy. Les architectures en Reverse Proxy sont typiquement utilisées pour sécuriser une grappe de serveurs http, fournir une URL d'accès homogène à des services http hétérogènes ou encore faire de la répartition de charge.

JPlatform doit être configuré explicitement pour fonctionner derrière un Reverse Proxy. En effet, il doit produire des liens hypertextes et des redirections http pour l'URL du Reverse Proxy au lieu de l'URL pour son propre serveur http.

Il est possible de déclarer plusieurs Reverse Proxy.

### 6.8.2 Configuration des propriétés JPlatform

Dans l'onglet *Site*, affecter les valeurs suivantes aux différents champs :

- *Actif* : cochez Oui ;
- *Adresse IP du Rev. Proxy* : l'adresse IP du Reverse Proxy (seules les requêtes en provenance de cette adresse seront traitées en mode Reverse Proxy)
- *Chemin de contexte externe* : chemin du site lorsqu'il est accédé de l'extérieur (par exemple /cms)
- *URL de base externe* : L'URL de base du site JPlatform lorsqu'il est accédé depuis l'extérieur.
- *URL de base SSL externe* : L'URL de base du site JPlatform lorsqu'il est accédé depuis l'extérieur en https.

### 6.9 Configuration du format des URLs d'affichage

Afin d'améliorer le référencement des sites JPlatform par les moteurs de recherche, la construction des URL d'affichage des données (publications, catégories, membres, ...) contient le titre de la donnée. Les moteurs de recherche augmentent en effet le classement d'une page en fonction des mots présents dans l'URL.

Dans JPlatform 10.0, les URLs du front-office sont de la forme :

`http://monsite.com/jcms/identifiant/le-titre-de-la-donnée.`

Le préfixe des URL (/jcms/) et le format de la partie textuelle (/le-titre-de-la-donnée) sont paramétrables. Il est par exemple possible d'ajouter le nom d'une catégorie, la date, etc. Ces paramètres se font via les propriétés `descriptive-url.*` dont les valeurs par défaut sont précisées dans le fichier `WEB-INF/jalios/jcms.prop` et qui peuvent être personnalisées dans le fichier `WEB-INF/data/custom.prop` ou dans le fichier `plugin.plugin.prop` de votre module principal.

Si le préfixe des URL doit être modifié, cela doit être fait avant toute production de publication car certaines publications possédant des champs de type *zone de texte* et *zone de texte riche* peuvent contenir des URL avec ce préfixe.

Pour changer le préfixe des URL, il faut effectuer les deux opérations suivantes :

1. Ajoutez la propriété `descriptive-urls.prefix` dans le fichier `WEB-INF/data/custom.prop` en utilisant le préfixe voulu. Exemple :

```
descriptive-urls.prefix: monprefixe
```

2. Editez le fichier `WEB-INF/web.xml` et remplacez toutes les occurrences de

```
<url-pattern>/jcms/*</url-pattern>
```

par :

```
<url-pattern>/monprefixe/*</url-pattern>
```

### 6.10 Configuration du pool de connexions

Il est recommandé d'activer le pool de connexion JDBC sur les sites JPlatform en production. Le pool de connexion augmente les performances, réduit la consommation de ressources sur la base et garantit plus de robustesse en cas de déconnexion avec la base.

Hibernate peut être configuré avec différents pool de connexion (C3P0, DBCP, ..). C3P0 est présent en standard dans JCMS. Il peut être configuré en modifiant les propriétés suivantes dans `WEB-INF/data/custom.prop`, utilisez des valeurs choisies précisément en fonction de la charge attendue sur le site :

```
# C3P0 properties
hibernate.cfg.common.prop.hibernate.c3p0.max_size:
hibernate.cfg.common.prop.hibernate.c3p0.min_size:
hibernate.cfg.common.prop.hibernate.c3p0.acquire_increment:
hibernate.cfg.common.prop.hibernate.c3p0.max_statements:
hibernate.cfg.common.prop.hibernate.c3p0.idle_test_period:
hibernate.cfg.common.prop.hibernate.c3p0.timeout:
```

En complément de ces propriétés, il est possible d'en définir d'autres pour une configuration plus avancée. Cependant, elles doivent être déclarées dans le fichier `WEB-INF/classes/c3p0.properties`. Attention ! Les propriétés définies dans `custom.prop` écrasent celles définies dans `c3p0.properties` et leur nom est différent.

Il est notamment recommandé de renseigner via la propriété `c3p0.checkoutTimeout` qui définit le timeout de prise d'une connexion en milliseconde (par défaut, il n'y a pas de timeout).

Par exemple, pour limiter la prise de connexion à 5 secondes :

```
c3p0.checkoutTimeout: 5000
```

Pour plus de détails sur la configuration de c3p0, consultez la documentation :

<http://www.mchange.com/projects/c3p0/index.html>

### 6.11 Configuration de la réplication par JSync

JPlatform intègre JSync, un protocole de réplication des données. Ce protocole assure la synchronisation des données entre plusieurs instances JPlatform. JSync permet de mettre en place à la fois des architectures à haute disponibilité (répartition de charge et redondance) et des architectures décentralisées (plusieurs instances JPlatform réparties sur un extranet). Ces différentes architectures sont décrites dans le livre blanc technique de JPlatform (<https://community.jalios.com/documentations/livresblancs>)

JSync est un protocole de réplication optimiste à propagation épidémique. JSync permet de mettre en place des topologies très souples. Dans une architecture répliquée, chaque instance de JPlatform est appelée un réplica. Les réplicas sont typiquement organisés en groupe avec un leader. Chaque réplica modifie ses données de façon autonome puis diffuse ses changements à son leader. Celui-ci les propage aux autres réplicas du groupe assurant ainsi la convergence globale du groupe. Un réplica peut se déconnecter, diverger puis se reconnecter pour diffuser ses nouveautés. JPlatform propose deux modes de propagation des changements :

- propagation automatique (diffusion dès que les nouvelles écritures se sont stabilisées) ;
- propagation manuelle (diffusion à la demande de l'administrateur)



Durant les phases de divergences, des conflits sur les données peuvent apparaître. Par exemple, une même donnée peut être modifiée sur un réplica et détruite sur un autre. Lors de la phase de diffusion des nouveautés, JSync résout automatiquement ce type de conflit. Les suppressions l'emportent et les mises à jour concurrentes sont réordonnées de façon homogène. Les conflits inter-objets ne sont pas traités (par exemple, suppression d'une catégorie sur un réplica et ajout d'un contenu référençant cette catégorie sur un autre réplica). Aussi, afin de prévenir tout conflit, il est recommandé de réduire les périodes de divergence. Pour cela, les contributeurs agissant sur des données liées doivent être regroupés sur un même réplica ou sur des réplicas configurés en propagation immédiate. Les lecteurs peuvent par contre être répartis sur l'ensemble des réplicas du groupe.

L'utilisation de JSync est incompatible avec le SGBDR interne de JPlatform 10.0, Derby. Vous devez donc mettre en place un SGBDR externe pour l'utiliser. Le leader et les réplicas doivent se connecter à la même base de données. Pour assurer la haute-disponibilité de la base, il est recommandé de mettre en place une base en cluster.

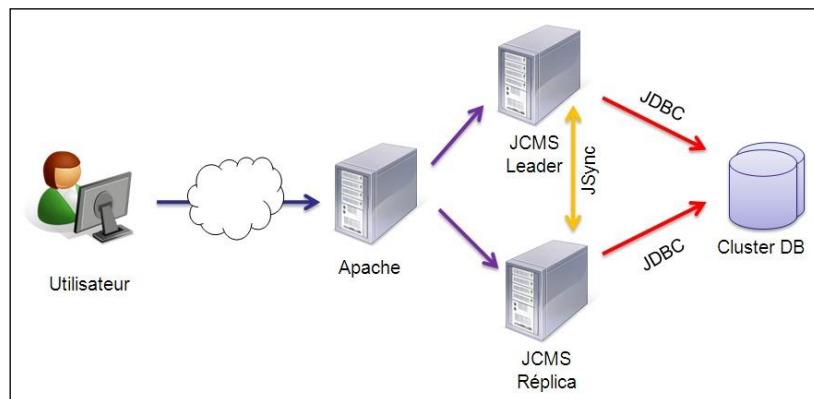


Figure 5 – Exemple de configuration haute-disponibilité pour JPlatform 10.0

Pour les détails sur la mise en œuvre de la réplication dans le cas de la répartition de charge, consultez l'article *Mise en œuvre d'un site JPlatform à haute disponibilité avec le protocole de réplication JSync* sur le site support de Jalios (<https://community.jalios.com/howto/jsync>).

### 6.12 Ajout d'un add-pack

JPlatform 10.0 est livré en standard avec une configuration permettant de gérer 1 espace de travail, 10 espaces collaboratifs<sup>1</sup>, 10 membres, 3 invités, 100 documents en base et 2 langues. Pour aller au-delà, vous devez acquérir et installer un *add-pack*.

Un add-pack se présente sous forme d'une clé cryptée qu'il faut déclarer avec la propriété `channel.add-pack`. Cette propriété doit être ajoutée dans le fichier `WEB-INF/data/custom.prop`.

Pour vérifier que le add-pack est bien pris en compte, consultez *Espace d'administration > Etat du site*. Le nom du add-pack figure dans la version de JPlatform.

---

<sup>1</sup> Nécessite le module Espaces Collaboratifs.

### 6.13 Configuration du serveur web frontal

JPlatform 10.0 ne nécessite pas l'utilisation d'un frontal web spécifique. Elle peut utiliser celui offert par le serveur d'application Java EE. Cependant la mise en place d'un frontal web, tel qu'Apache HTTPD ou IHS, permet :

- d'optimiser les performances,
  - coté serveur : en servant les fichiers statiques directement par le frontal web et en allégeant ainsi de manière significative la charge du serveur Java EE
  - coté client : en activant des en-têtes de cache HTTP sur les ressources statiques afin d'éviter aux clients de recharger les ressources à chaque requête
- d'assurer une haute disponibilité via la mise en place de mécanisme de répartition de charge (loadbalancing) et/ou de tolérance aux pannes (fail-over).

L'article « *Configurer Apache, mod\_jk et Tomcat pour JPlatform* » décrit la mise en place d'un frontal Apache HTTPD pour Tomcat et JPlatform :

<https://community.jalios.com/howto/apache>

L'article « *Mise en oeuvre d'un site JPlatform à haute disponibilité avec le protocole de réplication JSync* » décrit la mise en place de plusieurs application JPlatform synchronisé via le protocole JSync derrière un frontal Apache HTTPD :

<https://community.jalios.com/howto/jsync>

JPlatform 10.0 est certifié avec le serveur web Apache HTTPD 2.2 et 2.4 en utilisant le connecteur mod\_jk 1.2.

Veillez à respecter les contraintes suivantes :

- Les ressources dynamiques générées par JPlatform (jsp et servlet) ainsi que les ressources protégées par des droits d'accès dans JPlatform (répertoire upload/) doivent impérativement être traitées par le serveur Java EE. Référez-vous à l'article ci-dessus pour une liste exhaustive de ces ressources dynamique.
- Le `KeepAlive` doit être activé sur le serveur web frontal

### 6.14 Installation de Jalios Office Launcher

Si vous souhaitez pouvoir déclencher l'édition WebDAV depuis un autre navigateur qu'Internet Explorer, il faut installer le composant *Jalios Office Launcher* sur le poste des utilisateurs concernés.

Sur JPlatform, éditez le fichier `WEB-INF/data/custom.prop` et positionnez la propriété `channel.webdav.custom-protocol.enabled` à `true`. Redémarrez JPlatform.

Attention ! Une fois cette propriété positionnée, tous les contributeurs devront utiliser *Jalios Office Launcher*.

Téléchargez l'archive de *Jalios Office Launcher* sur Jalios Community :

<https://community.jalios.com/tools/jaliosofficelauncher>

Sur le poste utilisateur, quittez toutes les applications Microsoft Office (Word, PowerPoint, Excel, ...). Décompactez l'archive, et lancez le programme `setup.exe`.

Sur les postes Windows Vista, si vous accédez à votre site JPlatform en HTTP (et non HTTPS), il faut lancer le fichier `vista-webdav.reg` qui se trouve aussi dans l'archive, puis redémarrer le poste.

### 6.15 Ajout de modules

JPlatform intègre un mécanisme d'extension par modules (aussi appelés *plug-in*). Les modules JPlatform fournissent de nouvelles fonctionnalités (p. ex. nouvelles portlets, blog, podcast, connexion SSO, ...) Ces modules peuvent être proposés par Jalios ou par d'autres acteurs (éditeurs, intégrateurs, utilisateurs, ...). Chaque module possède sa propre licence d'utilisation.

Les modules JPlatform sont diffusés sur le site Jalios Community :

<https://community.jalios.com/plugin/>

Un module se présente sous forme d'une archive zip. L'ajout, le paramétrage et le retrait de module se fait via l'interface du Gestionnaire de modules de JPlatform.

Pour ajouter un module, dans la boîte *Déposer un module* sélectionnez le fichier zip du module avec le bouton *Parcourir...*, cochez la case *Déployer automatiquement* et cliquez sur le bouton *Déposer un module*. Une fois le module installé, JPlatform doit être redémarré.

Pour paramétrer un module, cliquez sur son nom dans la liste des modules et sélectionnez l'onglet *Administration*.

Pour supprimer un module, cliquez sur son nom dans la liste des modules et cliquez sur le bouton *Supprimer les fichiers déployés...*

## **7 Procédures d'exploitation**

### **7.1 Arrêt/démarrage de JPlatform**

Le démarrage et l'arrêt de JPlatform consistent simplement à démarrer et arrêter la webapp. Lorsque JSync est utilisé, il est recommandé d'arrêter la webapp du leader en dernier.

#### **7.1.1 ApacheTomcat**

Tomcat ne permet pas de démarrer ni d'arrêter une webapp en particulier, il faut démarrer ou arrêter l'ensemble du serveur.

#### **7.1.2 IBM WebSphere Application Server**

L'arrêt et le démarrage de JPlatform se font par la console d'administration de WebSphere Application Server.

##### **7.1.2.1 Démarrage**

Ouvrez la console de WebSphere Application Server : *Programmes > IBM WebSphere > Application Server v6 > Profiles > Default > Console d'administration.*

Allez dans le menu Applications > Applications d'entreprise. Cocher la case à côté de jcms10, puis cliquez sur *Démarrer*. La première exécution est assez longue, à cause de la compilation des fichiers.

##### **7.1.2.2 Arrêt**

Ouvrez la console de WebSphere Application Server : *Programmes > IBM WebSphere > Application Server v6 > Profiles > Default > Console d'administration.*

Aller dans le menu Applications > Applications d'entreprise. Cochez la case à côté de jcms10, puis cliquez sur *Arrêter*.

## 7.2 Monitoring

JPlatform 10.0 fournit une interface de monitoring de l'application. Outre la consommation mémoire, celle-ci présente, pour une période donnée, les redémarrages du site, les accès aux sites (requêtes et sessions) et le nombre d'objets présents dans le store. Toutes ces informations sont désormais persistantes et ne sont donc plus perdues lors des redémarrages. La fenêtre de persistance est paramétrable (via la propriété `monitoring.history-day`).

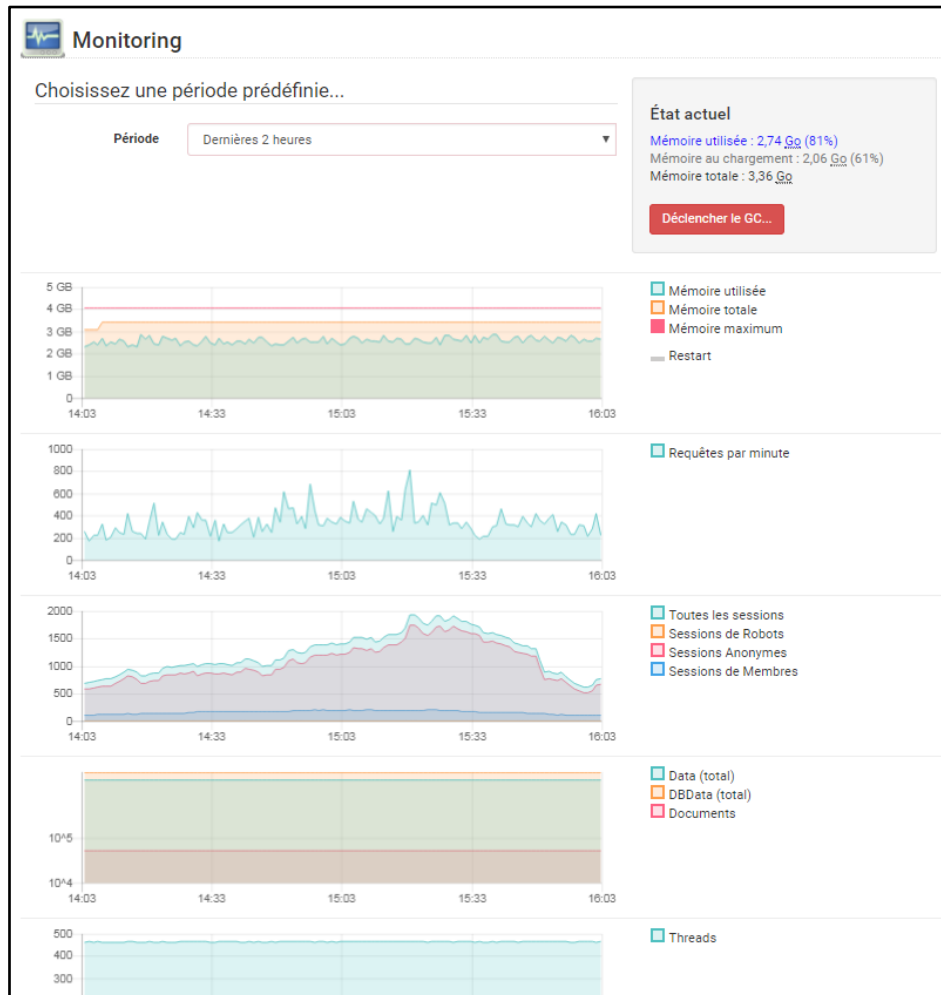


Figure 6 - Monitoring

Attention ! Les consommations mémoires indiquées sont issues des informations fournies par les API du JDK et sont généralement surestimées. Ces chiffres représentent la totalité de la mémoire actuellement consommée (une partie de cette mémoire sera relâchée plus tard). Par ailleurs, si plusieurs webapps fonctionnent sur la même JVM, la consommation affichée reflète les consommations cumulées de toutes les webapps.

Du fait de la gestion de la mémoire dans la JVM (basée sur un *garbagecollector*), il est tout à fait normal que la consommation augmente (jusqu'à 90%) et diminue régulièrement. Si par contre, la consommation ne redescend jamais en-dessous de 80%, il faut augmenter la mémoire allouée à la JVM.

### 7.3 Contrôle d'intégrité des données

Il est possible à tout moment de demander un contrôle de l'intégrité des données. Ce contrôle consiste à lister toute donnée ne respectant pas les contraintes d'intégrités qui lui sont assignées (p. ex. champs obligatoires, doublons, contrôles spécifiques, ...)

Pour lancer le contrôle d'intégrité, allez dans *Espace d'administration* > *Supervision* > *Contrôle d'intégrité des données*. Un rapport est généré lors du premier accès. Le calcul étant gourmand en ressources processeur, ce rapport est mis en cache (en mémoire). Pour l'actualiser, cliquez sur le bouton *Rafraîchir*.

#### Contrôle d'intégrité des données

**Attention**  
🚨 14 contraintes d'intégrité ne sont pas respectées.  
Date de l'analyse : 10/10/14 16:41

Rafraîchir Export tableau (CSV)

	ID	Type	Valeur	Contrainte d'intégrité		
📄	11.	c_8408	Workspace	Collaborative Space ...	Nom d'espace de travail déjà utilisé: Collaborative Space Model	🚨
📄	12.	c_8409	Group	Collaborative Space ...	Un groupe utilise déjà le nom 'Collaborative Space - Attendees'. Veuillez en choisir un autre.	🚨
📄	13.	c_8410	Group	Collaborative Space ...	Un groupe utilise déjà le nom 'Collaborative Space - Writers'. Veuillez en choisir un autre.	🚨
📄	14.	c_8443	Category	Virtual Desktop Serv...	Il existe déjà une catégorie nommée 'Virtual Desktop Services' dans la catégorie 'root'.	🚨

Page 2/2   1   2   10 par page ▼

Figure 7 - Contrôle de l'intégrité des données

## 7.4 Journal des événements (logs)

JPlatform 10.0 utilise la bibliothèque log4j pour la journalisation des d'évènements. Les journaux sont enregistrés dans le répertoire <WEBAPP\_DIR>/WEB-INF/data/logs/.

Les 50 derniers événements sont consultables depuis l'espace d'administration (*Supervision > Journal des événements*).

Pour plus de détails sur la configuration de la journalisation, consultez la fiche pratique sur Jalios Community : *La journalisation des événements avec log4j* (<https://community.jalios.com/howto/log4j>).












Journal des événements				
Tous les niveaux				
Niveau	Date	Heure	Message	Source
 Information	2014/10/10	15:58:32	Init the UploadManager	UploadManager
 Information	2014/10/10	15:57:19	StrongLockManager: init done	StrongLockManager
 Information	2014/10/10	15:57:19	StrongLockManager: cache enabled	StrongLockManager
 Information	2014/10/10	15:57:18	ESN Manager initialized.	ESNManager
 Information	2014/10/10	15:57:18	ESN: Agreement is required.	ESNManager
 Information	2014/10/10	15:57:18	ESN: Mail Notification is enabled.	ESNManager
 Information	2014/10/10	15:57:18	Graphviz is disabled.	MemberNetwork
 Information	2014/10/10	15:57:18	initNetwork	MemberNetwork
 Information	2014/10/10	15:57:17	Administrator "Admin" starts a new session	LoggingAuthenticationHandler
 Information	2014/10/10	15:57:17	Cleaning upload temp directory: C:\tomcat\work\Catalina\localhost\jcms\upload.tmp	MultipartFilter
Page 2/5   1   2   3   4   5   10 par page ▼				

Figure 8 - Journal des événements

### 7.5 Arrêt des écritures

Pour certaines opérations de maintenance, il peut être nécessaire d'empêcher momentanément les utilisateurs de contribuer. Pour cela, cliquez sur le bouton *Désactiver les écritures* situé dans l'Espace d'administration et entrez un message de maintenance. Durant l'arrêt des écritures :

- Toutes les écritures sont arrêtées dans le store et dans jcmsdb.
- Les écritures liées à la réplication par JSync restent actives (pour les arrêter, allez dans l'éditeur de propriété).
- Un message signalant l'arrêt des écritures défile dans le bandeau de l'Espace d'administration ;
- Les espaces de travail deviennent inaccessibles et le message de maintenance est affiché à la place ;
- L'espace d'administration est actif mais il n'est plus possible de modifier les membres, les groupes et les workflows ;
- Toutes les interfaces de contribution disponibles en front-office disparaissent ;
- Le suivi des lecteurs est désactivé.

Pour rétablir les écritures, cliquez sur le bouton *Activer les écritures*.



## 7.6 Gestion des index de recherche

En cours d'exécution aucune réindexation n'est nécessaire. Les index sont alimentés automatiquement lors des créations, mises à jour, suppressions ou synchronisations des données. De même, le module de suggestion d'orthographe de recherche est planifié pour se mettre à jour régulièrement. Cependant, après un déploiement, une migration ou une modification externe du store, il est nécessaire de lancer une réindexation.

Pour réindexer les publications, allez dans *Espace d'administration > Exploitation > Gestion des index de recherche* et cliquez sur le bouton *Tout réindexer*.

Attention, l'indexation et l'optimisation des index sont des opérations gourmandes en ressources processeur et en temps d'exécution. Néanmoins, le site et la recherche restent disponibles pendant la durée de ces opérations.

Gestion des index de recherche

En cours d'exécution aucune réindexation n'est nécessaire. Les index sont alimentés automatiquement lors des créations, modifications, suppressions ou synchronisations de publications et de catégories. De même, le module de suggestion d'orthographe de recherche est planifié pour se mettre à jour régulièrement. Cependant, après un déploiement, une migration ou une modification externe du store, il est nécessaire de lancer une réindexation.

Tout réindexer...

Publications

Catégories

Documents

Archives

Membre

Moteur de recherche des publications

Dernière réindexation	-
Dernière optimisation	-
<div> <div>Index anglais</div> <div> Chemin : C:\Users\lschetib\Documents\Dev\workspaces\jcms_trunk\UCMS\webapps\en\WEB-INF\data\lucene\PublicationsIndices\en  Dernière modification: il y a 40 min  Version : 1399986684706  Verrouillé : Non  Contient des documents supprimés : Oui  Nombre maximum de documents : 461  Nombre de documents : 456 </div> </div>	
<div> <div>Index français</div> <div> Chemin : C:\Users\lschetib\Documents\Dev\workspaces\jcms_trunk\UCMS\webapps\en\WEB-INF\data\lucene\PublicationsIndices\fr  Dernière modification: il y a 40 min  Version : 1399986684713  Verrouillé : Non  Contient des documents supprimés : Oui  Nombre maximum de documents : 461  Nombre de documents : 456 </div> </div>	

Réindexer toutes les Publications...

Optimiser les index...

⚠ Attention, l'indexation et l'optimisation sont des opérations gourmandes en ressources processeur et en temps d'exécution. La recherche est toujours disponible pendant ces opérations.

Figure 9 - Gestion des index de recherche

## 7.7 Gestion de caches

Les pages portail de JPlatform 10.0 peuvent contenir des caches (définis pour chaque portlet). Le gestionnaire des caches (*Espace d'administration > Exploitation > Gestion des caches*) permet de consulter et de vider ces caches.

### 7.8 Nettoyage du store

Le stockage des données de structure (catégorie, groupes, membre, portlet, ...) et des contenus repose sur JStore. A chaque écriture (création, mise à jour, suppression), JStore enregistre une ligne représentant cette opération dans le fichier `store.xml`. Ce fichier grossit au rythme des écritures. Le fichier `store.xml` n'est lu que dans trois cas :

1. au démarrage de JPlatform
2. lors de l'accès à l'historique des versions d'une publication (lecture partielle et mise en cache)
3. et lors des phases de synchronisation par JSync (lecture partielle).

Si la taille du fichier devient importante, p. ex. plus de 100 Mo, et que le nombre de mise à jour et de suppression est élevé (cf. *Espace d'administration > Supervision > Etat du store*), il peut être nécessaire de nettoyer le store.

**Attention ! Le nettoyage du store ne doit pas se faire sur un site en production.** Par ailleurs, si la réplication JSync est activée, il faut suivre la procédure suivante :

1. Arrêtez les écritures sur le(s) webapp(s) de production et synchroniser les réplicas le cas échéant.
2. Faites une copie complète du site (via le DeployManager)
3. Lancez le nettoyage du store sur la copie
4. Redéployez le store nettoyé sur toutes les webapps de production

Pour lancer le nettoyage, il faut préciser la portée du nettoyage (portion du fichier `store.xml` qui sera nettoyée) et les règles de nettoyage à appliquer. Une fois ces choix faits, cliquez sur le bouton *Nettoyer et redémarrer*. Le fichier `store.xml` actuel est copié (dans le répertoire `WEB-INF/data`) et remplacé par le store nettoyé, puis le site redémarre. Il convient de vérifier dans les traces, que le démarrage avec ce nouveau store s'est correctement déroulé.

Nettoyage du store

Le nettoyage du store est destiné à améliorer le temps de chargement du store, l'accès aux versions et les échanges par réplication. Le nettoyage peut aussi porter sur un store de développement avant sa diffusion en production. Le principe du nettoyage consiste à fusionner et à supprimer certaines opérations du store (création, modification et suppression). Le nettoyage ne supprime aucune donnée vivante.

Portée du nettoyage

Début	Fin
<input type="radio"/> Du début du store	<input type="radio"/> A la fin du store
<input type="radio"/> De la date <input type="text"/>	<input type="radio"/> A la date <input type="text"/>
<input type="radio"/> De la marque <input type="text"/>	<input type="radio"/> A la marque <input type="text"/>
<input type="radio"/> De l'estampille <input type="text"/>	<input type="radio"/> A l'estampille <input type="text"/>
<input type="radio"/> De l'estampille de synchronisation ( ) <input type="text"/>	<input type="radio"/> A l'estampille de synchronisation ( ) <input type="text"/>

Espaces de travail

URID

Règles de nettoyage

Nettoyer et redémarrer...

Figure 10 - Nettoyage du store

## 7.9 Procédures de sauvegarde/restauration

### 7.9.1 Sauvegarde automatique du Store

JPlatform 10.0 intègre un système de sauvegarde régulière du store. Celui-ci ne dispense pas de mettre en place un véritable système de sauvegarde régulière d'un site JPlatform (store, types, workflows, documents, propriétés, ...). L'objectif est de fournir une sauvegarde minimum du store pour pallier aux cas extrêmes (p. ex. sauvegarde externe déficiente). Ce système de sauvegarde est par défaut actif mais il peut être désactivé (onglet Avancé de l'éditeur de propriétés).

La sauvegarde du store comporte 3 paramètres :

1. La planification des sauvegardes (`store.backup.schedule`). Par défaut, tous les jours à 3h00 ;
2. Le nombre de sauvegardes à préserver (`store.backup.max`). Par défaut, 10 sauvegardes ;
3. Le répertoire des sauvegardes (`store.backup.dir`). Par défaut, `WEB-INF/data/backups/`.

### 7.9.2 Sauvegarde automatique de JcmsDB (Derby)

JPlatform intègre un système de sauvegarde régulière de la base JcmsDB gérée par Derby. **Si vous avez paramétré un SGBDR externe, la sauvegarde de la base n'est pas prise en charge par JPlatform.**

La sauvegarde de la base JcmsDB est par défaut active mais elle peut être désactivée (via la propriété `derby.backup.enabled`). La totalité du répertoire `WEB-INF/derby/jcmsdb` est sauvegardée sous forme de fichier zip. Pour restaurer l'une des archives, il suffit de la décompresser et de la placer dans ce même répertoire.

La sauvegarde de JcmsDB comporte 3 paramètres :

4. La planification des sauvegardes (`derby.backup.schedule`). Par défaut, tous les jours à 3h00 ;
5. Le nombre de sauvegardes à préserver (`derby.backup.max`). Par défaut, 10 sauvegardes ;
6. Le répertoire des sauvegardes (`derby.backup.dir`). Par défaut, `WEB-INF/data/backups/`.

### 7.9.3 Procédure de sauvegarde de la webapp JPlatform

Avant de procéder à une sauvegarde, il est conseillé d'arrêter JPlatform. Si cela ne peut être fait, il est important de vérifier que le contenu de la copie du répertoire `<WEBAPP_DIR>/WEB-INF/data/` est bien identique à l'original.

Dans un environnement répliqué par JSync, il est recommandé de sauvegarder chaque réplica.

### 7.9.4 Procédure de restauration de la webapp JPlatform

La procédure de restauration se fait, bien entendu, à l'inverse de la procédure de sauvegarde. Toutefois, il y a deux approches possibles.

Dans le cas d'une restauration *simple*, il faut stopper l'application JPlatform 10.0 avant de restaurer en écrasant les fichiers.

Dans le cas d'une restauration *complète*, il faut stopper le serveur d'application, supprimer la version courante du répertoire de l'application et la remplacer par la sauvegarde.

La restauration simple suffit dans l'essentiel des cas, car tous les anciens fichiers sont remis en place. Toutefois, les nouveaux fichiers ne sont pas remplacés, ce qui peut entraîner des effets de bord parfois gênants. La restauration complète assure le rétablissement d'une image cohérente et exhaustive.

Dans les deux cas, des effets de cache peuvent se produire. Il peut notamment être utile de nettoyer le répertoire de compilations des JSP :

- Apache Tomcat : `<TOMCAT_DIR>/work/Standalone/localhost/jcms/`
- IBM WebSphere AS : `<WAS_DIR>/temp/<machine>/<serveur>/jcms`

## 8 Mise à jour et déploiement

Le fonctionnement du script de déploiement est expliqué en détail sur Jalios Community : <https://community.jalios.com/howto/deploy>

### 8.1 Procédure générale

- Les développeurs travaillent sur leur webapp de développement (développements de nouveautés applicatives, migration de version, autre, ...).
- Le mainteneur récupère une version et génère un nouveau WAR complet.
- Le mainteneur déploie ce WAR sur la production en une ligne de commande grâce au script `deploy.sh`.

Le script déploie les modifications applicatives (fichiers statiques, jsp, plugin, classes, libraire, etcetc) et structurelles (workflow, types). Toutes les données de l'environnement de production sont conservées (upload, propriétés, base).

Concernant le fichier `store.xml` : c'est lors du déploiement que l'on décide comment le traiter, 3 possibilités :

1. le `store.xml` du nouveau war est ignoré : **c'est le comportement par défaut**, le `store.xml` de la production est conservé tel quel.
2. le `store.xml` du war et de la production sont fusionnés : l'application `StoreMerge` est invoqué lors du déploiement et les 2 stores sont fusionnés.
3. le `store.xml` du war remplace celui de la webapp production.

### 8.2 Développement et préparation du fichier WAR

1. Récupérez une copie complète de la production.
  - Celle-ci doit impérativement contenir **les plugins**.
  - La présence des répertoires `upload`, `archives`, `log` ou autres données n'a aucune importance pour le déploiement ultérieur, récupérez les si vous en avez besoin pour vos développements.
2. Effectuez et validez vos développements
3. Packagez la nouvelle version de la webapp dans un war en respectant les contraintes suivantes :
  - La présence de tous les plugins est impérative.
  - La présence du fichier `store.xml` n'est nécessaire que si des modifications doivent être réincorporées en production.
  - Il est inutile de mettre les répertoires `upload` et `archives` dans ce nouveau war. Ils seront ignorés.
  - Il est inutile de mettre les répertoires ou fichiers suivants (cependant les laisser dans le war n'a aucune incidence sur le déploiement) :
    - Base derby (`WEB-INF/derby`)
    - Index lucene (`WEB-INF/lucene`)
    - Statistiques, logs, monitoring, backup, ... (`WEB-INF/stats/`, `WEB-INF/logs/`, `WEB-INF/jsynclog/`, `WEB-INF/backups/`, `WEB-INF/monitoring.xml`)

### 8.3 Déploiement du WAR sur le serveur

#### 8.3.1 Mise en place initiale

Cette configuration doit être effectuée une seule fois au début de la vie du site :

- Déplacez les 3 répertoires suivant hors de la webapp
  - archives/
  - upload/
  - WEB-INF/data/
- Créez des liens symboliques **absolus** (p. ex. /opt/jcms/data/) pour référencer chacun de ces répertoires dans la webapp.

Toutes ces contraintes sont vérifiées par le script de déploiement (qui n'effectuera aucune modification si elles ne sont pas respectées).

Avec Tomcat, **et uniquement sur environnement linux**, il est nécessaire de configurer l'attribut `allowLinking` dans le contexte de la webapp :

```
<Context ...>
  <Resources allowLinking="true" />
</Context>
```

ou dans le contexte par défaut (`conf/context.xml`)

**Important : Pour garantir la sécurité de l'application, ne positionnez en aucun cas cet attribut dans un environnement Windows.**

#### 8.3.2 Déploiement d'une nouvelle version

##### Usage basique :

Cet usage conviendra dans la majorité des cas :

1. Déposez le nouveau war sur le serveur (ftp, ssh, ...)
2. Arrêtez le serveur Java EE,
3. Invoquez le script de déploiement

```
./deploy.sh nouveau.war repertoire-webapp/
```

4. Redémarrez le serveur Java EE

##### Usage avancé :

- Gestion du store :
  - utilisez l'option `-s "merge"` pour fusionner le store de la production et le store du war (une sauvegarde est effectuée avant modification)
  - utilisez l'option `-s "overwrite"` pour remplacer le store de la production par le store du war (une sauvegarde est effectuée avant modification)
- Option `-b` pour la sauvegarde de la webapp dans un autre répertoire : à la fin du processus, la webapp originale est sauvegardée dans le même répertoire dans lequel elle se trouvait, simplement suffixée avec la date de déploiement. Pour éviter que cette sauvegarde interfère avec le fonctionnement du serveur Java EE (par exemple avec Tomcat en mode déploiement automatique dans le répertoire webapp), il est possible

de spécifier un autre répertoire dans lequel la webapp originale sera déplacé avec l'option `-b`.

- Mode dry-run (option `-d`): dans ce mode aucune modification n'est effectuée, seul les pré-requis sont vérifiés et les opérations qui aurait été effectuées sont indiquées.
- Mode verbeux (option `-v`): à utiliser pour obtenir des logs détaillés sur l'exécution du script `deploy.sh` (utile pour le support, pour débogger ou pour comprendre le fonctionnement du script)

#### 8.4 Remarques

- **rollback** : En cas d'échec du déploiement (disque plein, échec de la fusion du store, autre erreur) un rollback est effectué et la webapp revient à son état initial.
- **sauvegarde** : si le `store.xml` est fusionné ou remplacé, la version existante est sauvegardée
- **structure** : les `types` et `workflows` de la production sont sauvegardés avant d'être remplacés par ceux du war
- **upload** : Le répertoire `upload` n'est pas récupéré du war, donc il ne peut pas y avoir de nouveaux documents déployés via ce mécanisme.
- **propriétés** : Le répertoire `WEB-INF/data` n'est PAS récupéré du nouveau war (sauf `store.xml`, `types` et `workflow`), les fichiers propriétés qui seraient modifiés dans le war (`custom.prop`) ne sont donc pas propagés. Le script de déploiement peut être modifié ou complété avec un autre script pour répondre à un besoin différent.

## 9 Equipement et configuration

### 9.1 Configuration du serveur

#### 9.1.1 Serveur et système d'exploitation

JPlatform 10.0 nécessite d'être installé sur un équipement serveur correspondant à l'une des configurations suivantes serveur compatible Intel :

- Windows Server 2012, 2016, 2019 ou 2022
- Linux sur base Debian ou RedHat,  
(Version stable supportée par l'éditeur et disposant d'un environnement Java supporté)

Le serveur doit disposer d'au moins 2 Go de RAM et 10 Go d'espace disque. Selon la volumétrie de données, il peut être nécessaire de disposer de plus mémoire. Pour les usages avec des volumétries dépassant les 100 000 objets, il est nécessaire de disposer de la quantité de RAM adaptée et si nécessaire d'un serveur et d'un système d'exploitation 64-bits.

#### 9.1.2 Serveur d'applications Java EE

JPlatform 10.0 est une application web conforme à Java EE6. Elle n'est pas dépendante d'un système d'exploitation mais nécessite un serveur d'application compatible Java EE6 (en particulier pour l'API Servlet 3.0).

JPlatform 10.0 requiert une JVM Oracle ou OpenJDK, en version 8 ou 11 (Le support de Java 11 est disponible à partir de JPlatform 10 SP2). L'utilisation de la JVM IBM est également certifiée sur IBM WebSphere AS.

- Apache Tomcat : Oracle JavaSE 8, OpenJDK 8, Oracle JDK 11 ou OpenJDK 11
- RedHat JBoss EAP : Oracle JDK ou OpenJDK, en version 8 ou 11, en respectant les certifications de votre version de JBoss EAP. Consultez l'article <https://access.redhat.com/articles/2026253> pour plus d'information à ce sujet.
- IBM WebSphere AS : Oracle JVM 1.8 ou IBM JVM 1.8

Le tableau ci-dessous récapitule les systèmes d'exploitation et les serveurs d'applications certifiés pour JPlatform 10.0.

Serveurs d'applications	Systèmes d'exploitation	
	Windows	Linux
Apache Tomcat 8.5 ou 9.0	•	•
IBM WebSphere AS 9.0	•	
JBoss EAP 7.0, 7.1, 7.2 ou 7.3	•	•

Afin de garantir la sécurité de votre plateforme et de vous permettre de toujours bénéficier des derniers correctifs, nous vous recommandons d'utiliser systématiquement la dernière version de maintenance du serveur d'application (en conservant la même version majeure indiquée).



### 9.1.3 SGBDR

JPlatform 10.0 est livré avec un SGBDR embarqué (Derby). JPlatform n'est pas certifié pour fonctionner en production avec Derby. Pour les environnements de production, il est recommandé de remplacer Derby par un SGBDR externe. JPlatform 10.0 est certifié sur plusieurs SGBDR externes :

- PostgreSQL 9, 10, 11, 12, 13 et 14
- MySQL 5 et 8 (moteur de stockage InnoDB)
- Oracle 11g, 12c, 19c et 21c
- Microsoft SQL Server 2012, 2016 et 2019
- MariaDB 10

### 9.1.4 Serveur Web

JPlatform 10.0 est compatible avec les serveurs web suivant :

- Apache HTTP 2.2 ou 2.4 et mod\_jk 1.2 ou mod\_proxy\_ajp comme connecteur Tomcat.
- IBM HTTP Server (IHS) dans les environnements WebSphere AS.

### 9.1.5 Serveur SMTP

JPlatform 10.0 requiert un serveur SMTP pour effectuer les envois de mails (notifications, alertes, bulletins, recommandation, ...).

### 9.1.6 Serveur POP3/IMAP (optionnel)

Si le mail entrant est activé, JPlatform 10.0 requiert un compte de messagerie sur un serveur POP3 ou IMAP pour gérer le mail entrant.

### 9.1.7 Annuaire LDAP (optionnel)

JPlatform 10.0 peut gérer les authentifications de manière autonome ou bien être interfacé avec un serveur LDAP/LDAPS. JPlatform 10.0 est certifié sur les annuaires suivants :

- Microsoft Active Directory
- Novell eDirectory
- OpenLDAP
- Oracle Directory Server

### 9.1.8 Configuration recommandée

Pour garantir la **sécurité** de votre plateforme, nous vous recommandons d'installer systématiquement les dernières versions de maintenance des applicatifs.

Pour les environnements nécessitant le support d'une **forte volumétrie** et une **forte capacité de montée en charge**, Jalios recommande fortement l'utilisation des applicatifs suivants pour la mise en œuvre de JPlatform :

- *OS* : Linux Server 64-bit
- *SGBDR* : PostgreSQL 9, 10, 11, 12, 13 ou 14
- *JDK* : Oracle JDK 11 ou OpenJDK 11

- *Frontal* : Apache 2.4.x en utilisant le connecteur mod\_jk 1.2.x ou mod\_proxy\_ajp
- *Serveur d'application* : Tomcat 8.5.x ou Tomcat 9.x

L'environnement serveur **minimum** recommandé (matériel ou virtualisé) :

- *Processeur Serveur* : Intel Xeon (8-16 cores) à 3 GHz ou équivalent
- *RAM* : 10 Go (8 Go pour l'application et 2 Go minimum pour le système)
- *Disques* :
  - 50 Go pour les données de l'application (hors système linux)
  - 5 Go pour l'installation de l'application (hors système linux)

Le dimensionnement de l'environnement serveur et la configuration des applicatifs dépend fortement de vos usages. Un dimensionnement correct doit s'effectuer par la mise en œuvre de tests en conditions proches du réel, avec la volumétrie et la charge utilisateur ciblées.

Pour des installations conséquentes, Jalios peut vous accompagner pour vous apporter son expertise dans le dimensionnement, les tests et le paramétrage de votre environnement. Rapprochez-vous de nos services professionnels pour bénéficier de recommandations ([info@jalios.com](mailto:info@jalios.com)). Jalios propose également un ensemble de ressources documentaires dédiées aux problématiques d'architecture technique : <https://community.jalios.com/architecture> (authentification nécessaire).

## **9.2 Configuration des postes**

### **9.2.1 Postes des contributeurs et des administrateurs**

Les contributeurs et les administrateurs doivent disposer d'une machine avec au moins 512 Mo de mémoire vive et un écran d'une résolution d'au moins 1280x800. Ils interagissent avec JPlatform 10.0 depuis un navigateur qui peut-être, au choix :

- Microsoft Edge
- Microsoft Internet Explorer 11 ou +, sans aucun mode de compatibilité
  - Fin de support de ce navigateur par Jalios à partir du 31 décembre 2021
  - Les nouveaux modules développés par Jalios et mis en téléchargement à partir de juillet 2021 n'assureront pas la compatibilité avec IE11
- Mozilla Firefox - dernières versions ESR supportées par Mozilla ou + récentes
- Google Chrome - dernières versions stables supportées par Google ou + récentes
- Apple Safari - dernières versions stables supportées par Apple ou + récentes

### **9.2.2 Postes des utilisateurs**

Les sites faits avec JPlatform 10.0 peuvent être consultés avec les navigateurs suivants, sous réserve que la charte graphique le permette :

- Microsoft Edge
- Microsoft Internet Explorer 11 ou +, sans aucun mode de compatibilité
  - Fin de support de ce navigateur par Jalios à partir du 31 décembre 2021
  - Les nouveaux modules développés par Jalios et mis en téléchargement à partir de juillet 2021 n'assureront pas la compatibilité avec IE11
- Mozilla Firefox - dernières versions ESR supportées par Mozilla ou + récentes
- Google Chrome - dernières versions stables supportées par Google ou + récentes
- Apple Safari - dernières versions stables supportées par Apple ou + récentes

Le lecteur de flux vidéo nécessite la présence d'un navigateur supportant le HTML 5 (ou du composant Adobe Flash Player 9 minimum pour les plus anciens navigateurs). Certains modules peuvent également nécessiter la présence du lecteur Flash ou de version spécifique des navigateurs.

### **9.2.3 Postes des développeurs**

Jalios recommande l'IDE Eclipse pour développer avec JPlatform 10.0. Le développeur doit disposer d'une machine avec au moins 4 Go de mémoire vive et un écran d'une résolution d'au moins 1280x800. Le développeur doit-être équipé d'un navigateur équivalent à celui des postes administrateurs.

© Jalios SA. Tous droits réservés. Le logo Jalios est une marque déposée de Jalios en France et à l'étranger. Les autres marques et logos cités sont déposés par leurs propriétaires respectifs. Les caractéristiques du produit sont susceptibles de changer sans préavis.