

W4 CONNECTORS FOR JAVA

MANUEL D'UTILISATION

Référence: W4JC_USER_022_FR

Les prochaines mises à jour de ce document seront disponibles sur
www.myw4.com

W4 CONNECTORS FOR JAVA

MANUEL D'UTILISATION

Référence: W4JC_USER_022_FR

© 2003 - 2010 W4. Tous droits réservés.

L'acquisition du présent document confère un droit d'utilisation incessible, non exclusif et personnel et non un droit de propriété. L'utilisation, la copie, la reproduction et la distribution du présent document sont permises à condition:

1. que la mention de droits d'auteur ci-dessus figure sur toutes les copies et que cette mention de droits d'auteur et la présente mention d'autorisation apparaissent conjointement ;
2. que le présent document ne soit utilisé qu'à des fins d'information et non commerciales ;
3. que le présent document ne soit modifié de quelque manière que ce soit.

Tous les produits et marques cités sont la propriété de leurs titulaires respectifs.

Les informations contenues dans ce document pourront faire l'objet de modifications sans préavis.

Vue d'ensemble ⁶

Principe de fonctionnement ⁶

Structure interne ⁷

Installation ¹⁰

Conditions préalables ¹⁰

Installation ¹⁰

Vérifier la version du JDK ou JRE ¹¹

Spécifier le répertoire W4Engine_Home ¹¹

Sélectionner l'instance W4 Engine d'installation ¹¹

Sélectionner le type d'installation ¹¹

Créer la langue XML, l'acteur XMLAgent, les rôles jconnector et XMLConnector ¹²

Mise à jour du fichier de configuration de W4 Engine (sys/w4server.cfg) ¹²

Journal de l'installation ¹²

Fichiers créés lors de l'installation ¹³

Configuration ¹⁴

Fichier sys/w4server.cfg ¹⁴

Fichier connector/lib/resources/connector.properties ¹⁶

Gestion du serveur de connecteurs ¹⁸

Le client TWFFconnectorClient ¹⁸

Possibilités ¹⁸

Préalables ¹⁹

Codes d'erreur ²⁰

Arrêt et relance ²⁰

Récupération des tâches perdues ²¹

Monitoring du serveur de connecteurs ²²

Les journaux et traces ²²

La console (monitor.bat ou monitor.sh) ²²

Installation de la console sur un autre poste que le serveur W4 Engine ²²

Lancement de la console ²³

Configuration de la console ²³

Informations fournies par la console ²⁴

Arrêt du serveur ²⁴

Connecteur XML ²⁶

Condition préalable : connaître les API XML de W4 ²⁶

Fonctionnalités du connecteur XML ²⁶

Arborescence ²⁷

Configuration du connecteur XML ²⁷

Utilisation des feuilles de styles XSL ²⁸



Vue d'ensemble

Ce chapitre décrit le fonctionnement du serveur de connecteurs Java. Le serveur de connecteurs est livré avec un kit de développement pour créer de nouveaux connecteurs (package java.connector.jar) et avec le connecteur XML (package java.xmlConnector.jar).

1.1 Principe de fonctionnement

Le schéma ci-dessous décrit l'intégration entre le serveur W4 Engine et le serveur de connecteurs.

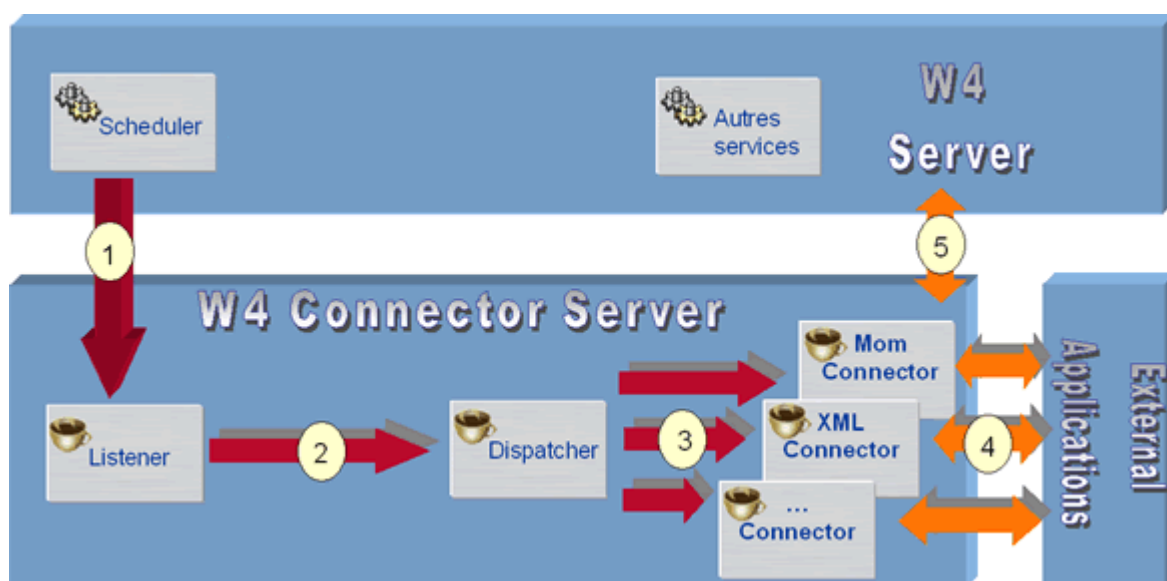


Fig 1.1 **Principe de fonctionnement**

- Lorsqu'une tâche est créée, le scheduler (service gestionnaire des tâches et des dossiers) vérifie que cette tâche n'est pas destinée à un acteur ayant le rôle `jconnector`. Si ce test est vrai, il envoie un message via socket IP au serveur de connecteurs. Ce message contient l'identifiant de la tâche créée, celui de l'acteur concerné, ses rôles et éventuellement l'adresse d'un fichier généré à partir du 'template' de l'activité.
- Le message est transmis au dispatcher du serveur de connecteurs.
- Le dispatcher envoie le message vers les connecteurs en fonction de leurs rôles. Le dispatcher envoie ensuite le message aux connecteurs en deux phases. Dans un premier temps, il interroge chaque connecteur afin de savoir s'il souhaite traiter ce message. Cela se traduit concrètement par l'appel de la méthode `acceptMessage` de chaque connecteur. A charge à chaque connecteur d'explorer le message (`TWFmessage`) afin de décider s'il souhaite le traiter (l'usage veut qu'un connecteur se base sur les rôles contenus dans le message pour décider de l'accepter ou non). Dans un deuxième temps, le dispatcher envoie une COPIE du message à chaque connecteur qui a précédemment décidé de l'accepter. Ainsi rien n'empêche un même message d'être traité par plusieurs connecteurs à la fois, pour peu que chacun l'ait accepté lors de la première phase. Toutefois ce type de besoin est très spécifique et il est recommandé de n'avoir qu'un seul connecteur qui répond à chaque message.
- Les connecteurs reçoivent le message et appellent des applications externes (facultatif). Il est important de noter que les connecteurs sont en mesure d'interagir également avec W4 Engine.
- Les connecteurs terminent la tâche workflow (facultatif).

Le serveur de connecteurs est une entité indépendante de W4 Engine, bien que les connecteurs maintiennent une connexion permanente avec ce dernier.

Le serveur de connecteurs reçoit des notifications en provenance de W4 Engine et, en fonction de certains critères émanant de la modélisation, assigne la notification à tel ou tel connecteur pour traitement.

Les critères de modélisation en question ne seront pas traités dans le présent document qui se limite à l'administration du serveur même plutôt qu'à la réalisation de connecteurs.

1.2 Structure interne

Le serveur de connecteurs est une application Java multi-threadée. On y trouvera :

- un Thread principal représentant le noyau du serveur ;
- un ThreadGroup dépendant de ce thread contenant tous les threads des connecteurs ;
- un Thread indépendant par connecteur.

En standard, 3 connecteurs sont livrés par W4 :

- **SocketServer** : connecteur ouvrant la socket d'écoute IP sur le port spécifié dans `w4server.cfg`. Toutes les notifications en provenance de W4 Engine sont transmises au connecteur **TWFdispatcher**.
- **TWFdispatcher** : connecteur répartissant les notifications en provenance de W4 Engine vers les connecteurs suivant les critères définis dans la modélisation de l'activité concernée.
- **[W4]XMLAgent** : connecteur XML interprétant l'API XML W4.

Pour de plus amples informations, veuillez consulter le chapitre suivant :

[6 Connecteur XML, page 26](#)

Si pour une quelconque raison un connecteur ne s'initialise pas correctement (par exemple s'il fait appel à une librairie externe qui n'est pas trouvée), ou rencontre un crash important durant son fonctionnement (une librairie manquante ou une exception Java non traitée par le connecteur), il sera désactivé par le serveur de connecteurs mais les autres connecteurs continueront à fonctionner.

Lorsque le dernier connecteur est désactivé, le serveur de connecteurs s'arrête.

Ainsi, lorsque W4 Engine crée une tâche destinée à un connecteur, une notification est envoyée au serveur de connecteurs. Dans l'hypothèse où le serveur de connecteurs ne serait pas lancé à cet instant, la tâche resterait offerte et la notification ne serait pas relancée ultérieurement par W4 Engine.

Il conviendra alors de relancer la notification manuellement.

Pour de plus amples informations, veuillez consulter le chapitre suivant :

[4.3 Récupération des tâches perdues, page 21](#)



Installation

Ce chapitre décrit la procédure d'installation du serveur de connecteurs.

2.1 Conditions préalables

IMPORTANT L'installation du serveur de connecteurs doit s'effectuer directement sur le serveur W4 Engine.

IMPORTANT Pour pouvoir lancer le programme d'installation du serveur de connecteurs, vous devez avoir installé au préalable un JDK ou un JRE de version 1.2.x ou ultérieure. Ces produits sont disponibles auprès de Sun Microsystems. Une fois le JDK ou JRE installé, vous devez mettre à jour la variable d'environnement PATH de telle sorte qu'elle pointe sur JAVA_HOME/bin où JAVA_HOME représente le répertoire d'installation du JDK ou JRE.

IMPORTANT Utilisation sur IBM AIX
Sur IBM AIX, le JDK 1.2.x présente des dysfonctionnements. Vous devez par conséquent utiliser une version 1.3.x ou ultérieure.
Or le JDK 1.3.x nécessite la version 4.3.3 ou ultérieure de IBM AIX

2.2 Installation

IMPORTANT Sous Unix, vous devez procéder à ces opérations en utilisant le compte de w4adm.

- Pour lancer le programme d'installation, ouvrez un shell.

- Placez-vous dans le répertoire du CD-ROM contenant le serveur de connecteur (répertoire `.../Setup/connserv/java` contenant le fichier `installCS.jar`).
- Lancez la commande suivante sous Windows : `install.bat`
- Lancez la commande suivante sous UNIX : `./install.sh`

NOTE Sur les systèmes UNIX, en cas de problème de lecture dû au type de système de fichiers choisi lors du montage du CD-ROM, une alternative consiste à copier le fichier `installCS.tar` sur le disque dur local : cette archive contient une copie de tous les fichiers nécessaires à l'installation.

2.2.1 Vérifier la version du JDK ou JRE

Cette étape est transparente si la version du JDK ou JRE installée est bien égale ou supérieure à la 1.2.x. Dans le cas contraire, un message d'erreur est lancé et l'installation se termine.

2.2.2 Spécifier le répertoire W4Engine_Home

Le programme d'installation demande le répertoire dans lequel W4 Engine est installé.

Exemple : `C:/Program Files/w4`

2.2.3 Sélectionner l'instance W4 Engine d'installation

Le programme d'installation demande sur quelle instance de W4 Engine le serveur de connecteurs doit être installé.

Vous pouvez sélectionner plusieurs instances.

2.2.4 Sélectionner le type d'installation

NOTE Le programme d'installation se compose de 3 étapes indépendantes les unes des autres. Si une erreur survient pendant l'installation, il est possible de relancer le programme d'installation en ne sélectionnant que les étapes à rejouer.

Les éléments suivants peuvent être installés :

- les fichiers concernant le serveur de connecteurs ;
- la création de la langue **XML**, de l'acteur **XMLAgent** et des rôles **jconnector** et **XMLConnector** ;



- la mise à jour du fichier de configuration du serveur W4 Engine (`W4Engine_Home/sys/w4server.cfg`).

2.2.5 Créer la langue XML, l'acteur XMLAgent, les rôles jconnector et XMLConnector

Les connecteurs sont avant tout des acteurs workflow. Pour que le serveur W4 Engine et le serveur de connecteurs soient en mesure de les distinguer des autres acteurs workflow, il faut que les acteurs connecteurs aient le rôle **jconnector**. Le programme d'installation se charge de créer ce rôle. L'acteur **XMLAgent** (qui a la langue **xml**) et le rôle **XMLConnector** sont dédiés au connecteur XML.

NOTE Cette étape est effectuée sur chaque instance choisie précédemment.

2.2.6 Mise à jour du fichier de configuration de W4 Engine (sys/w4server.cfg)

Il s'agit ici de l'ajout de paramètres de configuration. Ces paramètres sont détaillés ci-dessous. Rappelons qu'il peut exister un serveur de connecteurs par instance W4 Engine.

NOTE Cette étape est effectuée sur chaque instance choisie précédemment. Le numéro de port de base est 7777 ; il est ensuite incrémenté de 1 pour chaque instance.

2.3 Journal de l'installation

Une fois l'installation terminée, il est possible de consulter le journal de l'installation (stocké dans le répertoire temporaire de l'utilisateur ayant le programme d'installation ; le chemin complet est indiqué à la fin de l'installation). Pour repérer les éventuelles erreurs, le "visualiseur" du journal fait figurer dans la marge une croix rouge.

NOTE Le serveur de connecteurs se lance et s'arrête avec le serveur W4 Engine. Pour que l'installation soit donc prise en compte, vous devez redémarrer le serveur W4 Engine.

2.4 Fichiers créés lors de l'installation

Le programme d'installation copie les fichiers suivants dans le répertoire W4Engine_Home :

- **Sous le répertoire connector :**
 - java.policy (autorise le serveur de connecteurs et le moniteur à ouvrir une socket)
 - manageServer.bat (permet l'arrêt/relance du serveur de connecteurs)
 - manageServer.sh (version Unix)
 - monitor.bat (lance le moniteur du serveur de connecteurs)
 - monitor.sh (version Unix)
 - recoverTasks.bat (rejoue les tâches perdues du serveur de connecteurs)
 - recoverTasks.sh (version Unix)
- **répertoire connector/lib (contient les librairies java nécessaires au fonctionnement du serveur de connecteurs) :**
 - parser.jar
 - jaxp.jar
 - saxon.jar
 - xalan.jar
 - xerces.jar
- **répertoire connector/lib/int (contient les connecteurs internes) :**
 - connector.jar (le kernel du serveur de connecteurs)
 - wfsocserver.jar (le listener)
- **répertoire connector/lib/ext (les connecteurs externes) :**
 - xmlConnector.jar (le connecteur XML)
- **répertoire connector/lib/resources (les fichiers de configuration) :**
 - connector.properties (le fichier de configuration du serveur de connecteurs)
 - xmlConnector.properties (le fichier de configuration du connecteur XML)
- **répertoire connector/xml (le connecteur xml) :**
 - pickup (les fichiers placés dans ce répertoire sont automatiquement analysés)
 - error (répertoire où sont déplacés les fichiers XML erronés qui ont placés dans le pickup)
 - xsl (répertoire contenant les feuilles de style associées aux acteurs W4)
- **répertoire connector/docs (la documentation)**



Configuration

Ce chapitre décrit les paramètres de configuration du serveur de connecteurs.

3.1 Fichier sys/w4server.cfg

```
w4adm.instanceHome:c:/w4
w4adm.databaseServer:vodka8i
w4adm.databaseOwner:w4
w4adm.databasePassword:7ieh
w4adm.licenseKey:391FA756D1CD5B8244824049D82E2E285853F3E12A462
D123D972E
#w4adm.lockingDelay.minute:1
w4adm.maxPendingConnections:25
# JAVA CONNECTOR PARAMETERS ###
# true: Generate <instanceName>.runSrv.bat in <logDir> when W4@
Engine starts.
# This file is used by start/stop script 'manageServer.sh' and by
'recoverTasks.sh'
w4adm.attachJVM:true
# true: Launch W4 connector server when W4@ Engine starts
w4adm.jconnector:true
# W4 Connector server port
w4adm.jconnector_port:7777
# Directory containing java binary
w4adm.java_home:C:/Progra~1/Java/j2re1.4.1_01
# Directory containing W4 connector java packages
w4adm.java_classpath:". ;C:\Program
Files\w4\connector\lib\saxon.jar;C:\Program
Files\w4\connector\lib\jaxp.jar;C:\Program
Files\w4\connector\lib\parser.jar;C:\Program
Files\w4\lib\java\wfjlib.jar;C:\Program
Files\w4\connector\lib\xalan.jar;C:\Program
Files\w4\connector\lib\xerces.jar;C:\Program
Files\w4\connector\lib\;C:\Program
Files\w4\connector\lib\int\connector.jar"
# true: Disables the class garbage collection
#w4adm.java_xnoclassgc:true
```



```

# Define minimum heap size
#w4adm.java_xms:10m
# Define maximum heap size
#w4adm.java_xmx:100m
# W4® Engine timeout in seconds when waiting for connector server
launch.
#w4adm.java_sleepTime:30
# Uncomment this line to activate the trace mode
#w4adm.java_other_param:-DWFTRACE=+t+all
# JAVA CONNECTOR PARAMETERS END ###

```

Les paramètres concernant le serveur de connecteurs sont les suivants :

- `w4adm.attachJVM`
si `true`, signifie que le serveur de connecteurs doit pouvoir être lancé séparément de W4 Engine. Un fichier `<instance>.runSrv.bat` sera donc généré par W4 Engine dans le répertoire de journalisation de l'instance concernée. Ce dernier est construit à partir des informations contenues dans `w4server.cfg`, ce qui signifie que s'il doit être régénéré après modification de la configuration, W4 Engine devra être redémarré.
- `w4adm.jconnector`
si `true`, le serveur de connecteurs sera démarré automatiquement à chaque démarrage de W4 Engine, et arrêté à chaque arrêt de W4 Engine.
- `w4adm.jconnector_port`
le port sur lequel la socket IP de communication est ouverte. Valeur par défaut : 7777.
- `w4adm.java_home`
chemin vers le JAVA_HOME. En cas d'utilisation d'un JDK/JRE 1.4.x, un autre paramètre doit être ajouté ; voir ci-dessous.
- `w4adm.java_classpath`
spécifie le classpath JAVA qui sera utilisé par le serveur de connecteurs. Ainsi, si l'on souhaite utiliser une librairie supplémentaire pour l'un des connecteurs, on l'ajoutera ici.
- `w4adm.java_xnoclassgc`
`true` : désactive le ramasse-miettes JVM.
`false` : active le ramasse-miettes JVM. Valeur par défaut.
- `w4adm.java_xms`
taille minimale du heap.
- `w4adm.java_xmx`
taille maximale du heap.
- `w4adm.java_sleepTime`
durée en secondes durant laquelle W4 Engine attend le démarrage du serveur de connecteurs avant de le déclarer en timeout. Valeur par défaut : 30.
- `w4adm.java_other_param`
paramètres additionnels qui seront ajoutés à la ligne de commande JAVA. On y trouvera en général des paramètres de JVM du type `-Dparam=value`. Dans le fichier `w4server.cfg`, par défaut, on peut trouver plusieurs exemplaires de cette ligne pour des usages divers sous forme de commentaires. Si ce paramètre figure plusieurs fois dans le fichier de configuration, seule la dernière valeur sera prise en compte. Il ne peut donc y en avoir qu'une à un instant donné. Il conviendra alors de concaténer éventuellement les valeurs sur un même exemplaire de ce paramètre.

IMPORTANT Ne pas activer les traces en production.



IMPORTANT Si les deux paramètres `jconnector` et `attachJVM` valent `false`, le serveur de connecteurs sera considéré comme non utilisé par W4 Engine, qui ne lui enverra aucune notification.

3.2 Fichier `connector/lib/resources/connector.properties`

IMPORTANT Sous Unix, vous devez procéder à ces opérations en utilisant le compte de `w4adm`.

Ce fichier permet de procéder à une configuration plus fine du serveur. En particulier, il s'agit des temps de lancement et d'arrêt.

```
#Timeout to connect to w4 server default(30000)
w4adm.launch.timeout=30000
#Time to wait before each try to connect to w4@ Engine (default
1000)
w4adm.launch.waittime=1000
#Timeout to stop a connector (default 10000)
w4adm.stop.timeout=20000
```

Les paramètres concernant le serveur de connecteurs sont les suivants :

- `w4adm.launch.timeout`
Délai total en ms pour se connecter au serveur W4 Engine (défaut : 30000).
- `w4adm.launch.waittime`
Temps d'attente en ms entre deux tentatives de connexion (défaut : 1000).
- `w4adm.stop.timeout`
Délai en ms pour l'arrêt de chaque connecteur (défaut : 10000).



Gestion du serveur de connecteurs

Ce chapitre décrit les outils et commandes nécessaires à la gestion du serveur de connecteurs.

4.1 Le client TWFconnectorClient

Un outil client a été ajouté au package du serveur de connecteurs depuis sa version 5.0. Il s'agit d'un programme Java offrant principalement deux fonctionnalités.

4.1.1 Possibilités

Les deux opérations réalisables par le TWFconnectorClient sont :

- l'arrêt du serveur de connecteurs ;
- la récupération de tâches "perdues" après un arrêt du serveur de connecteurs.

La syntaxe du programme TWFconnectorClient peut être obtenue via la ligne de commande :

```
java -cp lib\int\connector.jar;..\lib\java\wfjlib.jar  
fr.w4.connector.TWFconnectorClient -help
```

où les chemins d'accès des librairies `connector.jar` et `wfjlib.jar` sont à adapter.

```
Usage      :   fr.w4.connector.TWFconnectorClient  [-help]  [-v]  
[-w4Instance <W4 instance name>] [-port <port> | -logDir <W4 log  
directory>] [-host <host>] -cmd <shutdown|shutdownNow>  
fr.w4.connector.TWFconnectorClient  [-help]  [-v]  
[-w4Instance <W4 instance name>] [-port <port> | -logDir <W4 log  
directory>] [-host <host>] -cmd recover [-w4Server <W4 Server>]  
[-w4Admin <actor>] [-w4Pass <password>] [-w4Actors <actors>]  
fr.w4.connector.TWFconnectorClient -help
```

```

-help: print this screen
-v: verbose mode
-w4Instance <W4 instance name>: W4 instance name
  Defaults to 'w4adm'.
-port <port>: specify W4 connector server port (on <host>)
  Overrides '-logDir' parameter.
  Defaults to '7777'.
-logDir <W4 log directory>: specify W4 log directory
  will be used to guess the W4 connector server port from
'runSrv.bat'
-host <host>: host running the W4 connector server
  Defaults to 'localhost'.
-cmd <shutdown|shutdownNow|recover>: command to send to W4
connector server
  Specific options for recover command:
  -w4Server <W4 Server:port>: host running the W4 server and
wflistener port
  Defaults to 'localhost'.
  -w4Admin <actor>: W4 actor with enough rights to reassign
tasks.
  Defaults to 'w4adm'.
-w4Pass <password>: password of W4 actor given with 'w4Admin'.
  Defaults to 'w4adm'.
-w4Actors <actors>: comma separated list of W4 actor names
  Lost tasks will be recovered for these actors.
  Defaults to all tasks for actors with 'jconnector' role if
no w4Actors specified
  Ex: java fr.w4.connector.TWFconnectorClient -v -port 7777 -cmd
shutdown
  Ex: java fr.w4.connector.TWFconnectorClient -v -cmd recover
-port 7777 -w4Server myServer -w4Instance w4prod -w4Actors
actor1,my actor2,actor3,actor with,,comma,actor5

```

Ainsi, trois commandes sont disponibles :

- `shutdown` : déclenche un arrêt du serveur de connecteurs, après que toutes les tâches connecteurs en cours ont été terminées, ou que la durée `<instance>.stop.timeout` est atteinte.

Pour plus de détails sur ce paramètre, voir le chapitre :

[3.2 Fichier connector/lib/resources/connector.properties, page 16](#)

- `shutdownNow` : provoque un arrêt immédiat du serveur de connecteurs ; toutes les opérations en cours sont interrompues même si elles ne sont pas terminées.
- `recover` : lance la récupération des tâches connecteurs perdues.

Voir le chapitre :

[4.3 Récupération des tâches perdues, page 21](#)

4.1.2 Préalables

Il s'agit d'une application cliente à part entière qui, dans le cas de la commande `recover`, se connecte à W4 Engine en utilisant l'API Java.

Il convient donc de configurer un environnement client :

- Les bibliothèques W4 ".dll" ou ".so" doivent être accessibles, c'est-à-dire dans la variable d'environnement `PATH` sous Windows, `LD_LIBRARY_PATH` sous Solaris ou Linux, `LIBPATH` sous AIX, ou encore simplement spécifiées à la commande `java` elle-même via le paramètre `-Djava.library.path=<path>`.



- Les bibliothèques Java `connector.jar` et `wfjlib.jar` doivent être disponibles dans le classpath de la commande `java`.
- Dans le cas où le moteur ciblé est en mode "installation cloisonnée", et que l'on tente d'y accéder en mode de communication local (c'est-à-dire que l'on ne spécifie pas l'option `-w4Server`), il convient de configurer la variable d'environnement `W4_ADM_DIR` de la manière adéquate.

4.1.3 Codes d'erreur

```

1 : Help
2 : Critical error occurred
3 : Unable to connect to W4 connector server
4 : Missing or wrong argument. See help.
5 : Unable to guess port from file '<instance_name>.runSrv.bat'.
   Check file, or use '-port' option
6 : Unable to find file '<instance_name>.runSrv.bat' in given log
   directory. Check file and path, or use '-port' option
20 : W4 error occurred: unable to connect to W4 server. Check
    server and instance. Is W4 server running
21 : W4 error occurred: Given W4 actor has insufficient
    privileges. Check that actor is able to reassign tasks.
22 : W4 error occurred: Wrong actor or password. Unable to login
    to W4 server.

```

4.2 Arrêt et relance

Le serveur de connecteurs peut actuellement être arrêté de trois manières :

- lorsque l'on stoppe W4 Engine, et que le serveur de connecteurs a été lancé initialement par W4 Engine (c'est-à-dire que le paramètre `<instance>.jconnector` est positionné à `true`) ;
- en cliquant sur les boutons Normal Shutdown ou Shutdown Now dans le moniteur des connecteurs ;
- en ligne de commande via un appel au programme `TWFconnectorClient` avec les commandes `shutdown` ou `shutdownNow`.

Le serveur de connecteurs peut être démarré de deux manières :

- lorsque l'on démarre W4 Engine, et que le paramètre `<instance>.jconnector` est positionné à `true` ;
- en ligne de commande via un appel au script `manageServer.bat/.sh` avec la commande `start`.

IMPORTANT L'appel à cette commande ne peut fonctionner que si le moteur W4 Engine a été démarré au moins une fois avec le paramètre `<instance>.attachJVM` positionné à `true` dans le fichier `w4server.cfg`. En effet, cela provoque la génération d'un fichier indispensable au démarrage différé du serveur de connecteurs.

4.3 Récupération des tâches perdues

La récupération de tâches perdues peut prendre place dans un contexte tel que celui-ci :

- le moteur W4 Engine est démarré ;
- le serveur de connecteurs est démarré ;
- des tâches connecteurs sont régulièrement créées et sont réalisées par les connecteurs ;
- le serveur de connecteurs est stoppé pour une quelconque raison ;
- des tâches connecteurs continuent à être créées, mais les connecteurs ne peuvent les réaliser ;
- on redémarre le serveur de connecteurs, les tâches connecteurs sont à nouveau traitées, mais celles créées dans l'intervalle restent en l'état ;
- on appelle le TWFconnectorClient comme suit :

```
java fr.w4.connector.TWFconnectorClient -v -cmd recover port 7777  
-w4Admin w4adm -w4Pass admpass -w4Server Punch -w4Instance w4prod
```

(On considère que les préalables en termes de bibliothèques présentés ci-dessus sont remplis).

Cet appel va :

- se connecter au serveur de connecteurs sur le port 7777 et obtenir sa date exacte de démarrage ;
- se connecter au moteur W4 Engine Punch (en mode RPC), sur l'instance `w4prod`, avec l'utilisateur `w4adm` et le mot de passe `admpass` ;
- rechercher tous les acteurs W4 possédant le rôle `jconnector` ;
- rechercher pour chacun de ces acteurs toutes les tâches à l'état actif (offert, en cours ou suspendu) créées avant le redémarrage du serveur de connecteurs ;
- pour chaque tâche trouvée, la réassigner au même acteur afin de lancer une nouvelle notification au serveur de connecteurs.

NOTE des acteurs sur lesquels restreindre l'opération peuvent être nommément spécifiés sur la ligne de commande via l'option `w4Actors`.



Monitoring du serveur de connecteurs

Ce chapitre décrit les outils de supervision du serveur de connecteurs.

5.1 Les journaux et traces

Ils sont placés dans les mêmes répertoires que ceux du serveur W4 Engine, c'est-à-dire sur plate-forme Windows (`W4Engine_Home/adm/log`) et UNIX (`var/adm/w4/log`).

Ces fichiers suivent la convention de nommage suivante :

`System***.log` et `System***.tra`

5.2 La console (`monitor.bat` ou `monitor.sh`)

La console du serveur de connecteurs permet de consulter en temps réel l'activité des connecteurs.

5.2.1 Installation de la console sur un autre poste que le serveur W4 Engine

- Copier le fichier `W4Engine_Home/connector/lib/int/connector.jar` sur le poste cible
- Mettre à jour la variable d'environnement `CLASSPATH` du poste cible de sorte que :
`CLASSPATH=%CLASSPATH%;connector.jar` (sur Windows)
ou `CLASSPATH=$CLASSPATH:connector.jar` (sur Unix)

- Lancer la commande : `java fr.w4.connector.monitor.TWFmonitor`

5.2.2 Lancement de la console

- Sur Windows, lancer `monitor.bat`
- Sur Unix, lancer `monitor.sh` (vérifier si le script est exécutable)

Ces 2 fichiers de commandes sont situés dans le répertoire `W4Engine_Home/connector/`

5.2.3 Configuration de la console

Renseignez les champs suivants pour configurer la console :

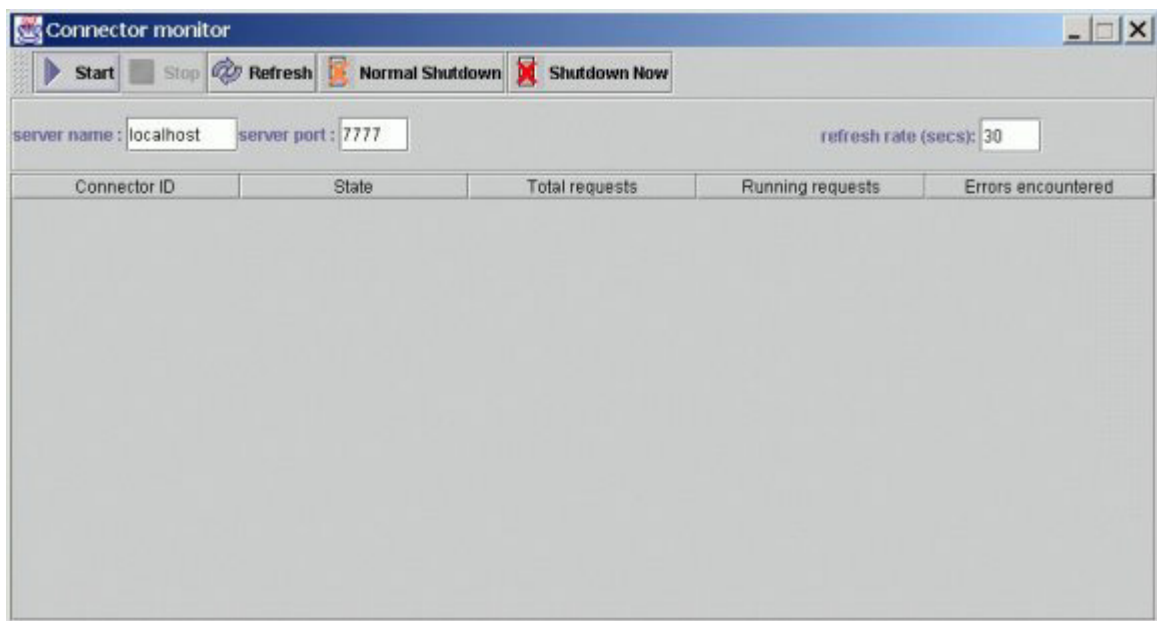


Fig 5.1 **Configuration de la console**

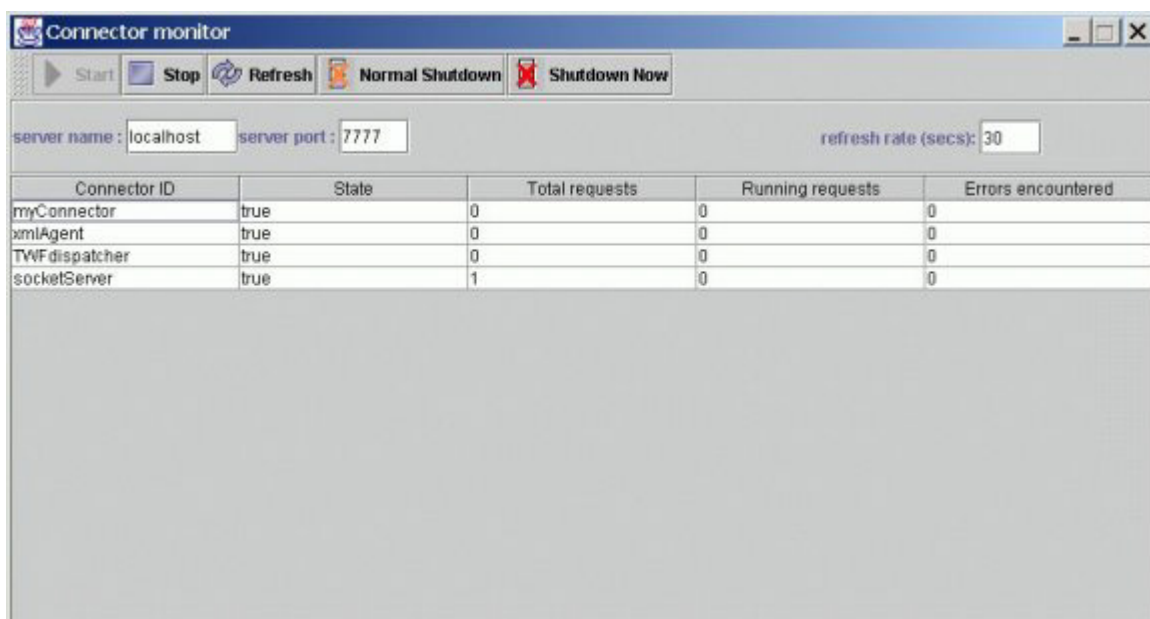
- Dans le champ **server name**, entrer le nom du serveur (ou son adresse IP) sur lequel s'exécute le serveur de connecteurs.
- Dans le champ **server port**, entrer le port utilisé par le serveur de connecteurs. Il s'agit du même numéro de port que celui précisé dans le fichier de configuration `w4server.cfg`.
- Le champ **refresh rate** indique le taux de rafraîchissement des informations fournies par la console.

Pour lancer la console, appuyez sur le bouton **start**. Pour arrêter la console, appuyez sur le bouton **stop**. Le bouton **refresh** a pour effet, non de lancer un rafraîchissement des informations affichées mais de mettre à jour la configuration de la console lorsqu'elle est déjà démarrée.

NOTE Si aucun serveur de connecteurs ne tourne sur le serveur et le port indiqués, une fenêtre contenant le message **Connection refused: connect** apparaît.

5.2.4 Informations fournies par la console

La console dresse la liste des connecteurs en cours d'exécution. Par défaut, les connecteurs suivants doivent apparaître : **xmlAgent** (le connecteur XML), **listener** (le connecteur chargé de la communication avec le serveur W4 Engine et avec les consoles), **dispatcher** (connecteur distribuant les messages du serveur W4 Engine vers les connecteurs chargés de leurs traitements).



The screenshot shows a window titled "Connector monitor" with a toolbar containing buttons for Start, Stop, Refresh, Normal Shutdown, and Shutdown Now. Below the toolbar, there are input fields for "server name" (localhost) and "server port" (7777), and a "refresh rate (secs)" field set to 30. A table displays the status of four connectors: myConnector, xmlAgent, TWF dispatcher, and socketServer. The table has columns for Connector ID, State, Total requests, Running requests, and Errors encountered.

| Connector ID | State | Total requests | Running requests | Errors encountered |
|----------------|-------|----------------|------------------|--------------------|
| myConnector | true | 0 | 0 | 0 |
| xmlAgent | true | 0 | 0 | 0 |
| TWF dispatcher | true | 0 | 0 | 0 |
| socketServer | true | 1 | 0 | 0 |

Fig 5.2 Informations fournies par la console

Pour chaque connecteur, on obtient les informations suivantes : son état, le nombre de messages traités, le nombre de messages en cours d'exécution, le nombre d'exceptions rencontrées.

Un connecteur qui n'y figure plus a probablement été détruit par le serveur pour une opération non conforme.

5.2.5 Arrêt du serveur

IMPORTANT A exécuter au cas où le serveur W4 Engine n'a pas réussi à arrêter le serveur de connecteurs.

Il y a deux manières d'arrêter le serveur de connecteurs avec la console.

- **Normal shutdown**

Le serveur de connecteurs va arrêter les connecteurs normalement.

- **Shutdown now**

Le serveur de connecteurs s'arrête sans tenir compte de l'activité des connecteurs.



Connecteur XML

Ce chapitre décrit l'utilisation du connecteur XML. Ce connecteur est installé par défaut avec le serveur de connecteurs.

6.1 Condition préalable : connaître les API XML de W4

Pour être en mesure d'utiliser le connecteur XML, il est nécessaire de connaître les API XML de W4. Ces API sont décrites dans le Java Toolkit de W4 (si vous utilisez un serveur W4 Engine antérieur à la version 03.00.00) ou dans le répertoire `W4Engine_Home/Docs/fr/java` (si vous utilisez un serveur W4 Engine dont la version est supérieure ou égale à la 03.00.00).

6.2 Fonctionnalités du connecteur XML

Le connecteur XML assure les fonctionnalités suivantes :

- gestion d'un répertoire pickup (possibilité de placer des fichiers XML dans le répertoire pickup qui seront automatiquement interprétés) ;
- interprétation de messages XML en provenance d'autres connecteurs (comme le connecteur MOM) ;
- possibilité d'associer à un message XML une feuille de style XSL.

6.3 Arborescence

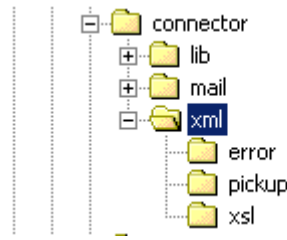


Fig 6.1 **Arborescence**

Le répertoire du connecteur XML est par défaut `W4Engine_Home/connector/xml`. Il contient 3 répertoires :

- `pickup` : répertoire dans lequel il est possible de déposer des fichiers XML qui seront automatiquement interprétés.
- `error` : le connecteur XML y déplace les fichiers XML en provenance du répertoire `pickup` lorsqu'une erreur intervient pendant leur interprétation.
- `xsl` : contient les feuilles de style associées aux acteurs.

6.4 Configuration du connecteur XML

L'acteur `xmlAgent`

L'acteur `xmlAgent` est utilisé par le connecteur XML pour passer les ordres sur le serveur W4 Engine contenus dans les fichiers déposés dans le répertoire `pickup`.

NOTE L'acteur `xmlAgent` est automatiquement créé lors de l'installation du serveur de connecteurs.

Il n'y a que 3 points à retenir de la définition de l'acteur `xmlAgent` :

- son nom de connexion : `xmlAgent` ;
- sa langue : le XML (automatiquement créée lors de l'installation du serveur de connecteurs) ;
- son mot de passe : `xmlAgent`.

Comme tous les connecteurs, le connecteur XML a le rôle `jconnector`. Il se voit également assigner le rôle `xmlConnector` qui lui est spécifique. Ses rôles sont automatiquement créés lors de l'installation du serveur de connecteurs.

Configuration du connecteur XML pour l'instance `w4adm`

Cette configuration s'effectue dans le fichier

`connector/lib/resources/xmlConnector.properties`.

```
#login of the w4's actor  
w4adm.w4.actor=xmlAgent
```

```

#name of w4's role
w4adm.w4.role=xmlConnector
#enable or disable the xml peeker
w4adm.peeker.enabled=false
#the peeker's delay (milliseconds)
w4adm.peeker.delay=10000
#pickup, error, xsl directories : default is
W4_HOME/connector/xml/pickup, error, xsl
w4adm.pickUpDir=C:/data/w4adm/pickup
w4adm.errorDir=C:/data/w4adm/error
w4adm.xslDir=C:/data/w4adm/xsl

```

Les paramètres `w4adm.w4.actor` et `w4adm.w4.role` s'appuient sur la définition de l'acteur et des rôles évoqués ci-dessous. `w4adm.w4.actor` doit contenir le nom de l'acteur affecté au connecteur XML. Si vous le modifiez, il faudra également modifier la définition de l'acteur `xmlAgent` dans l'interface d'administration de W4 (accessible à http://SERVER_NAME/W4Public). Il en va de même pour le rôle `xmlConnector`.

Les paramètres `w4adm.peeker.enabled` et `w4adm.peeker.delay` s'appliquent à la gestion du répertoire `pickup` du connecteur XML. Le paramètre `w4adm.peeker.enabled` peut avoir 2 valeurs possibles : `true` ou `false`. S'il est positionné à `false`, le démon vérifiant l'arrivée de nouveaux fichiers XML dans le répertoire `pickup` est désactivé. S'il est positionné à `true`, le démon vérifiera la présence de fichiers XML dans le répertoire `pickup` toutes les `m` millisecondes, `m` étant défini par le paramètre `w4adm.peeker.delay`.

6.5 Utilisation des feuilles de styles XSL

Il est possible d'associer au connecteur XML une feuille de styles qu'il devra appliquer avant de traduire les flux XML. Cette feuille de styles doit être placée dans le répertoire `W4Engine_Home/connector/xml/xsl`. Cette feuille de style doit respecter la convention de nommage suivante : `nomDeLActeurXML.xsl`.

Si vous avez conservé l'acteur `xmlAgent`, sa feuille de style doit être appelée `xmlAgent.xsl`.



| | |
|--------------------------------------|----|
| Principe de fonctionnement | 6 |
| Configuration de la console | 23 |
| Informations fournies par la console | 24 |
| Arborescence | 27 |



A

`attachJVM`, paramètre d'instance, W4 Engine 15, 16, 20

C

Connecteur XML, configuration, W4 Java Connectors 27

Connecteur XML, W4 Java Connectors 26

`connector.jar` 20

D

`dispatcher`, W4 Java Connectors 7, 24

F

Fichier `connector.properties`, W4 Java Connectors 16

J

`java_classpath`, paramètre d'instance, W4 Engine 15

`java_home`, paramètre d'instance, W4 Engine 15

`java_other_param`, paramètre d'instance, W4 Engine 15

`java_sleepTime`, paramètre d'instance, W4 Engine 15

`java_xms`, paramètre d'instance, W4 Engine 15

`java_xmx`, paramètre d'instance, W4 Engine 15

`java_xnoclassgc`, paramètre d'instance, W4 Engine 15

`jconnector_port`, paramètre d'instance, W4 Engine 15

`jconnector`, paramètre d'instance, W4 Engine 15, 16, 20

JDK 10, 11

JRE 10, 11

L

launch.timeout, paramètre d'instance, W4 Engine 16
launch.waittime, paramètre d'instance, W4 Engine 16
LD_LIBRARY_PATH 19
LIBPATH 19
listener, W4 Java Connectors 24

M

monitor.bat, W4 Java Connectors 22

N

Normal shutdown W4 Java Connectors 20
Normal shutdown, W4 Java Connectors 24

P

PATH 19
peeker.delay, paramètre, W4 Java Connectors 28
peeker.enabled, paramètre, W4 Java Connectors 28
Pickup, W4 Java Connectors 26

R

Répertoire pickup, W4 Java Connectors 26
Rôle jconnector 7, 11, 12, 21
Rôle XMLConnector 11, 12
Rôle xmlConnector 28

S

Service scheduler, W4 Engine 7
Shutdown now, W4 Java Connectors 20, 25
SocketServer, W4 Java Connectors 7
stop.timeout, paramètre d'instance, W4 Engine 16

T

Traces, W4 Java Connectors 15, 22
TWfconnectorClient, W4 Java Connectors 18, 20, 21
TWfdispatcher, W4 Java Connectors 7



V

Variable d'environnement PATH 10

W

W4 Java Connectors récupération de tâches perdues 18
W4 Java Connectors, arrêt 18, 19, 20
W4 Java Connectors, arrêt immédiat 19
W4 Java Connectors, classpath JAVA 15
W4 Java Connectors, client TWFconnectorClient 18
W4 Java Connectors, configuration 14
W4 Java Connectors, connecteur XML 26
W4 Java Connectors, console 22
W4 Java Connectors, délai d'attente W4 Engine 15
W4 Java Connectors, démarrage automatique 15
W4 Java Connectors, démarrage indépendant de W4 Engine 15
W4 Java Connectors, fichiers journaux 22
W4 Java Connectors, installation 10
W4 Java Connectors, moniteur 20, 22
W4 Java Connectors, monitor.bat 22
W4 Java Connectors, monitor.sh 22
W4 Java Connectors, paramètres pour la ligne de commande JAVA 15
W4 Java Connectors, port de communication 15
W4 Java Connectors, principe de fonctionnement 6
W4 Java Connectors, programme TWFconnectorClient 20, 21
W4 Java Connectors, ramasse-miettes JVM 15
W4 Java Connectors, récupération des tâches perdues 19, 21
W4 Java Connectors, relance 20
W4 Java Connectors, taille maximale du heap 15
W4 Java Connectors, taille minimale du heap 15
W4_ADM_DIR 20
w4.actor, paramètre, W4 Java Connectors 28
w4.role, paramètre, W4 Java Connectors 28
wfjlib.jar 20

X

XML, langue, W4 Java Connectors 11, 12
XMLAgent, acteur, W4 Java Connectors 11, 12
xmlAgent, acteur, W4 Java Connectors 27
XMLAgent, W4 Java Connectors 7
xmlAgent, W4 Java Connectors 24
xmlConnector.properties 27



W4 CONNECTORS FOR JAVA

MANUEL D'UTILISATION

Référence: W4JC_USER_022_FR

Pour toute remarque ou suggestion concernant ce document, vous pouvez contacter le support technique W4, en précisant la référence du document :

- par le service de traitement des dossiers Supportflow sur MyW4.com, à l'adresse suivante : <http://support.myw4.com>
- Par courrier électronique : support@w4global.com
- Par téléphone : +33 (0) 820 320 762