

# Journalisation Verbale dans WebSphere Liberty Server

Pour répondre à votre question concernant l'option de journalisation verbale pour la commande `server` dans WebSphere Liberty Profile (WLP), clarifions ce que fait la commande `server` et comment la verbosité peut être contrôlée.

La commande `server` dans WLP est utilisée pour gérer les instances de serveur, avec des actions comme `start`, `stop`, `run`, ou `status`. Par exemple :

- `server start <serverName>` démarre le serveur en arrière-plan, avec la sortie redirigée vers des fichiers journaux (par exemple, `logs/console.log`).
- `server run <serverName>` démarre le serveur en premier plan, affichant la sortie directement dans le terminal.

Votre question porte sur une “option de journalisation verbale” pour la commande `server`, ce qui pourrait impliquer soit une sortie plus détaillée de la commande elle-même, soit une journalisation plus détaillée du serveur qu’elle gère. Après avoir exploré les options, il est clair que la commande `server` ne dispose pas d’un drapeau direct comme `--verbose` ou `-v` pour augmenter sa propre verbosité de sortie. Au lieu de cela, la verbosité concerne le comportement de journalisation du serveur, qui peut être influencé lors de l’appel de la commande.

## Activation de la Journalisation Verbale

Dans WLP, la verbosité des journaux est contrôlée par la configuration de journalisation du serveur, et non directement par un drapeau de la commande `server`. Voici comment vous pouvez activer la journalisation verbale :

**1. Configurer la Journalisation dans `server.xml`** La méthode principale pour activer la journalisation verbale consiste à ajuster l’élément `<logging>` dans le fichier `server.xml` du serveur, généralement situé dans `<WLP_HOME>/usr/servers/<serverName>/`. Vous pouvez définir une spécification de trace détaillée pour augmenter la verbosité des journaux. Par exemple :

```
<logging traceSpecification="*=all" />
```

- `*=all` active tous les points de trace, rendant les journaux extrêmement verbaux (utiles pour le débogage).
- Pour une verbosité plus ciblée, vous pouvez spécifier des composants, par exemple, `*=info:com.example.*=debug` en définissant le niveau par défaut sur `info` mais `debug` pour le package `com.example`.

D’autres attributs utiles incluent : - `logLevel` : Définit le niveau de journalisation général (par exemple, `INFO`, `DEBUG`, `TRACE`). - `consoleLogLevel` : Contrôle le niveau des messages écrits dans `console.log` ou le terminal (par exemple, `DEBUG` ou `TRACE`).

Exemple avec un niveau de console plus fin :

```
<logging consoleLogLevel="DEBUG" traceSpecification="*=audit" />
```

Lorsque vous exécutez `server start`, les journaux (y compris la sortie verbale) vont dans `logs/console.log`. Avec `server run`, cette sortie verbale apparaît directement dans votre terminal.

**2. Utiliser des Variables d'Environnement** Vous pouvez également contrôler la verbosité des journaux via des variables d'environnement, qui remplacent ou complètent les paramètres de `server.xml`. Par exemple, définissez la variable `WLP_LOGGING_CONSOLE_LOGLEVEL` avant d'exécuter la commande `server` :

```
export WLP_LOGGING_CONSOLE_LOGLEVEL=DEBUG  
server start <serverName>
```

- Cela définit le niveau de journalisation de la console sur `DEBUG`, assurant une sortie plus détaillée.
- Les niveaux valides incluent `INFO`, `DEBUG`, `FINE`, `FINER`, `FINEST`, etc.

Pour `server run`, cela rendra la sortie du terminal plus verbale ; pour `server start`, cela affecte `console.log`.

**3. Options Verbales Spécifiques à la JVM** Si vous êtes intéressé par des sorties verbales spécifiques comme les journaux de collecte de déchets, vous pouvez ajouter des options JVM (bien que cela soit lié au temps d'exécution du serveur, et non à la commande `server` elle-même). Ajoutez `-verbose:gc` au fichier `jvm.options` dans `<WLP_HOME>/usr/servers/<serverName>/`, ou incluez-le via `server.xml` :

```
<jvmEntries verbose="gc" />
```

Cela génère des journaux liés à la collecte de déchets, généralement dans `console.log` ou le terminal avec `server run`.

### Comment Cela Se Relie à la Commande `server`

- **Pas d'Option `--verbose` Directe** : La commande `server` elle-même (par exemple, `server start`) ne dispose pas d'une option `--verbose` ou similaire pour détailler ses étapes d'exécution au-delà des messages d'état de base (par exemple, "Serveur démarré").
- **La Journalisation du Serveur Contrôle la Verbosité** : La verbosité que vous voyez dépend de la configuration de journalisation du serveur. Utilisez `server run` pour observer les journaux en direct dans le terminal, ou vérifiez les fichiers journaux après `server start`.

Par exemple : - Avec `<logging traceSpecification="*=all"/>` dans `server.xml`, l'exécution de `server run <serverName>` inonde le terminal de journaux détaillés. - Avec `server start <serverName>`, la même verbosité apparaît dans `logs/console.log`.

## **Recommandation**

Pour obtenir une journalisation verbale lors de l'utilisation de la commande `server` : 1. Modifiez `server.xml` pour inclure une spécification de trace détaillée (par exemple, `*=all` pour une verbosité maximale ou un niveau spécifique comme `*=debug`). 2. Alternativement, définissez `WLP_LOGGING_CONSOLE_LOGLEVEL=DEBUG` dans votre environnement. 3. Utilisez `server run` pour voir la sortie verbale directement, ou `server start` et vérifiez `logs/console.log`.

Cette approche garantit que vous obtenez les journaux détaillés dont vous avez besoin, adaptés à vos exigences de débogage ou de surveillance.

Si vous vouliez quelque chose de plus spécifique (par exemple, la verbosité de l'exécution de la commande plutôt que les journaux du serveur), n'hésitez pas à clarifier ! Sinon, configurer la journalisation du serveur comme décrit est la méthode standard pour activer la journalisation verbale avec la commande `server` dans WLP.

---

Note : Soyez prudent avec le répertoire `logs`. Il contient des journaux, tels que des fichiers de trace.