

वर्बोज़ लॉगिंग इन वेबस्पीयर लिबर्टी सर्वर

आपके `WILDFLY_HOME` `LOGS` `LOG` (`INFO`) में `server` कमांड के लिए वर्बोज़ लॉग ऑशन के बारे में पूछे गए प्रश्न को समझने के लिए, हम `server` कमांड की कार्यवाही और वर्बोज़िटी को कैसे नियंत्रित किया जा सकता है, इस बारे में स्पष्टीकरण करेंगे।

`INFO` में `server` कमांड का उपयोग सर्वर इंस्टेंस को प्रबंधित करने के लिए किया जाता है, जैसे कि `start`, `stop`, `run`, या `status` जैसे कार्यों के साथ। उदाहरण के लिए:

- `server start <serverName>` सर्वर को पृष्ठभूमि में शुरू करता है, और आउटपुट को लॉग फ़ाइलों में (जैसे `logs/console.log`) रीडायरेक्ट करता है।
- `server run <serverName>` सर्वर को फॉरग्राउंड में शुरू करता है, और आउटपुट को सीधे टर्मिनल में दिखाता है।

आपने “वर्बोज़ लॉग ऑशन” के बारे में पूछा है, जो कमांड की खुद की वर्बोज़ आउटपुट या कमांड द्वारा प्रबंधित सर्वर की वर्बोज़ लॉगिंग का संकेत कर सकता है। विकल्पों को खोजने के बाद, स्पष्ट है कि `server` कमांड में `--verbose` या `-v` जैसा कोई सीधा फ्लैग नहीं है, जो अपनी खुद की आउटपुट वर्बोज़िटी बढ़ाने के लिए है। बजाय इसके, वर्बोज़िटी सर्वर की लॉगिंग व्यवहार से संबंधित होती है, जो कमांड को चलाने के समय प्रभावित हो सकती है।

वर्बोज़ लॉगिंग को सक्रिय करने के लिए

`INFO` में, लॉग्स की वर्बोज़िटी को कमांड के सीधे फ्लैग के बजाय सर्वर की लॉगिंग कनफिगरेशन के माध्यम से नियंत्रित किया जाता है। यहाँ यह कैसे किया जा सकता है:

1. server.xml में लॉगिंग कनफिगर करें वर्बोज़ लॉगिंग को सक्रिय करने का प्राथमिक तरीका है सर्वर के `server.xml` फ़ाइल में `<logging>` तत्व को सेट करने से, जो आमतौर पर `<WLP_HOME>/usr/servers/<serverName>/` में स्थित होता है। आप एक विस्तृत ट्रेस स्पेसिफिकेशन सेट कर सकते हैं, जिससे लॉग वर्बोज़ हो जाएं। उदाहरण के लिए:

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

- `*=all` सभी ट्रेस पॉइंट्स को सक्रिय करता है, जिससे लॉग्स बहुत वर्बोज़ हो जाते हैं (डिबगिंग के लिए उपयोगी)।
- अधिक लक्षित वर्बोज़िटी के लिए, आप घटकों को स्पेसिफिक कर सकते हैं, उदाहरण के लिए, `*=info:com.example.*=debug`, जो डिफॉल्ट स्तर को `info` सेट करता है लेकिन `com.example` पैकेज के लिए `debug`.

अन्य उपयोगी गुण हैं: - `logLevel`: सामान्य लॉग स्तर सेट करता है (जैसे `INFO`, `DEBUG`, `TRACE`)। - `consoleLogLevel`: `console.log` या टर्मिनल में लिखे गए संदेशों की स्तर को नियंत्रित करता है (जैसे `DEBUG` या `TRACE`)।

एक फाइलर कंसोल स्तर के साथ उदाहरण:

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

जब आप `server start` चलाते हैं, तो लॉग्स (वर्बोज़ आउटपुट सहित) `logs/console.log` में जाते हैं। `server run` के साथ, यह वर्बोज़ आउटपुट सीधे आपकी टर्मिनल में दिखाई देता है।

2. परिवेश चर का उपयोग करें आप लॉग वर्बोज़िटी को परिवेश चरों के माध्यम से भी नियंत्रित कर सकते हैं, जो server.xml सेटिंग्स को ओवरराइड या पूरक कर सकते हैं। उदाहरण के लिए, server कमांड को चलाने से पहले WLP_LOGGING_CONSOLE_LOGLEVEL चर को सेट करें:

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

- यह कंसोल लॉग स्तर को DEBUG सेट करता है, जिससे अधिक विस्तृत आउटपुट सुनिश्चित होता है।
- मान्य स्तरों में INFO, DEBUG, FINE, FINER, FINEST शामिल हैं।

server run के लिए, यह टर्मिनल आउटपुट को अधिक वर्बोज़ बना देता है; server start के लिए, यह console.log को प्रभावित करता है।

3. ०००-वशिष्ट वर्बोज़ ऑप्शन अगर आप विशेष वर्बोज़ आउटपुट जैसे गैरेज़ कलेक्शन लॉग्स में रुचि रखते हैं, तो आप ००० ऑप्शन जोड़ सकते हैं (हालांकि यह सर्वर रनटाइम से संबंधित है, नहीं server कमांड खुद से)। jvm.options फ़ाइल में -verbose:gc जोड़ें, जो <WLP_HOME>/usr/servers/<serverName>/ में स्थित है, या इसे server.xml के माध्यम से शामिल करें:

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

यह ०००-संबंधित लॉग्स पैदा करता है, जो आमतौर पर console.log या server run के साथ टर्मिनल में होते हैं।

यह server कमांड से कैसे संबंधित है

- **कोई सीधा --verbose ऑप्शन नहीं:** server कमांड (जैसे server start) में कोई --verbose या समान फ्लैग नहीं है, जो अपने कार्यवाही चरणों को विस्तृत रूप में दिखाने के लिए है, बस बुनियादी स्थिति संदेशों (जैसे “सर्वर शुरू हुआ”) के अलावा।
- **सर्वर लॉगिंग वर्बोज़िटी को नियंत्रित करता है:** आप देखते हैं वर्बोज़िटी, सर्वर की लॉगिंग कनफिगरेशन पर निर्भर करती है। server run का उपयोग करें, ताकि लॉग्स को सीधे टर्मिनल में देखें, या server start के बाद लॉग फ़ाइलें देखें।

उदाहरण के लिए: -server.xml में <logging traceSpecification="*=all"/> के साथ, server run <serverName> चलाने से टर्मिनल में विस्तृत लॉग्स आते हैं। -server start <serverName> के साथ, वही वर्बोज़िटी logs/console.log में दिखाई देती है।

सिफारिश

server कमांड का उपयोग करते समय वर्बोज़ लॉगिंग प्राप्त करने के लिए: 1. server.xml को संपादित करें, ताकि एक विस्तृत traceSpecification शामिल हो (जैसे *=all अधिकतम वर्बोज़िटी के लिए या एक विशेष स्तर जैसे *=debug). 2. या तो WLP_LOGGING_CONSOLE_LOGLEVEL=DEBUG को अपने परिवेश में सेट करें। 3. server run का उपयोग करें, ताकि वर्बोज़ आउटपुट को सीधे देखें, या server start और logs/console.log देखें।

यह तरीका आपको अपने डिबगिंग या मॉनिटरिंग आवश्यकताओं के अनुसार विस्तृत लॉग्स सुनिश्चित करता है।

अगर आप कुछ और विशेष (जैसे कमांड कार्यवाही वर्बोज़िटी की बजाय सर्वर लॉग्स) का मतलब रखते हैं, तो स्पष्टीकरण के लिए स्वतंत्र हैं! अन्यथा, सर्वर की लॉगिंग को जैसा वर्णित किया गया है, server कमांड में वर्बोज़ लॉगिंग को सक्रिय करने का मानक तरीका है।