✅ **Run JMeter Tests Using Command Line**

**🚀 Basic Command**

bash

jmeter -n -t <test\_plan.jmx> -l <results.jtl>

| **Flag** | **Meaning** |
| --- | --- |
| -n | Non-GUI mode |
| -t | Test Plan path |
| -l | Results file (JTL) output |

**🎨 To Generate HTML Report**

bash

jmeter -n -t <test\_plan.jmx> -l <results.jtl> -e -o <output\_folder>

| **Flag** | **Meaning** |
| --- | --- |
| -e | Generate HTML dashboard |
| -o | Folder to save report *(must be empty or non-existent)* |

**🧪 Real-World Example (Windows)**

Assume:

* JMeter location: C:\apache-jmeter-5.6.2\bin
* Test Plan: C:\PerfTests\LoginTest.jmx
* Output files:
  + JTL: C:\PerfTests\Results\Login\_Results.jtl
  + HTML Report: C:\PerfTests\Reports\Login\_20250712

**Command:**

bash

cd C:\apache-jmeter-5.6.2\bin

jmeter -n -t "C:\PerfTests\LoginTest.jmx" -l "C:\PerfTests\Results\Login\_Results.jtl" -e -o "C:\PerfTests\Reports\Login\_20250712"

**🔧 Parameterize Test Plan (Dynamic via CLI)**

In your .jmx file, define:

bash

Thread Count: ${\_\_P(users,100)}

Ramp-Up: ${\_\_P(rampup,30)}

Run with:

bash

jmeter -n -t MyTest.jmx -l results.jtl -Jusers=500 -Jrampup=60

**📦 Advanced CLI Flags**

| **Flag** | **Purpose** | **Example** |
| --- | --- | --- |
| -j logfile.log | JMeter internal log | -j logs/run\_1.log |
| -Jprop=value | Override JMeter property | -Jusers=100 |
| -Gprop=value | Send prop to remote engines | -Genv=staging |
| -Dprop=value | Java system properties | -Duser.timezone=UTC |
| -Llogger=LEVEL | Logging verbosity | -Lorg.apache=DEBUG |

**🛡 Best Practices for CLI Execution**

✅ **Use CLI for Load Testing**  
Avoid GUI mode to prevent memory bottlenecks and inaccurate results.

✅ **Remove Visual Listeners (e.g., View Results Tree)**  
These are helpful during test design but should be removed or disabled for load tests.

✅ **Keep Results in Separate Folders**  
Use date or timestamp to organize results:

bash

-o Reports/Test\_$(date +%Y%m%d%H%M)

✅ **Add Sleep/Delays in CI if Needed**  
Allow time for HTML report generation before next stage.

✅ **Check JMeter Logs**  
Always review -j logfile.log for warnings or thread issues.

✅ **Automate with CI Tools (Optional)**  
Jenkins, GitLab CI, Azure DevOps can run the CLI command and collect artifacts (JTL, HTML).

**📁 Folder Structure Example**

bash

/PerfTests

│

├── LoginTest.jmx

├── Results/

│ └── Login\_Results.jtl

├── Reports/

│ └── Login\_20250712/

├── Logs/

│ └── Login\_Run.log

**📌 Troubleshooting Tips**

| **Issue** | **Fix** |
| --- | --- |
| "Output folder not empty" error | Use a clean or new folder for -o |
| JTL not created | Ensure file path is valid and no permission issue |
| HTML not generated | Ensure -e and -o used correctly; no syntax error in test plan |
| Parameters not passed | Confirm variables use ${\_\_P()} in the JMX |

**🧠 Pro Tip: Batch Script Example (Windows)**

bat

@echo off

set JMETER\_PATH=C:\apache-jmeter-5.6.2\bin

set TEST\_PLAN=C:\PerfTests\LoginTest.jmx

set RESULTS=C:\PerfTests\Results\Login\_Results.jtl

set REPORT=C:\PerfTests\Reports\Login\_%date:~-4%%date:~4,2%%date:~7,2%

cd %JMETER\_PATH%

jmeter -n -t "%TEST\_PLAN%" -l "%RESULTS%" -e -o "%REPORT%" -Jusers=100 -Jrampup=30 -j "C:\PerfTests\Logs\login\_run.log"