

# GPU Factory Test Framework

The **GPU Factory Test Framework** is a modular, Java 21-based validation suite designed for **post-silicon GPU bring-up and production testing**. It automates tasks like PCIe link verification, thermal monitoring, memory usage checks, fan speed validation, and power consumption analysis. The framework is intended for seamless integration into CI/CD pipelines and factory automation systems.

---

## Features

- PCIe link training validation
  - Thermal monitoring
  - GPU memory usage validation
  - Fan speed verification
  - Power consumption analysis
  - Command-line interface with task selection and device targeting
  - Report export in JSON, CSV, and ZIP formats
  - Unit and integration tests with JUnit 5
  - Docker support for reproducible builds and isolated execution
- 

## Build Instructions

Ensure **Java 21** and **Maven** are installed.

```
mvn clean package
```

Output:

```
target/gpu-factory-test-framework-1.0-SNAPSHOT-shaded.jar
```

---

## Running the Framework

```
java -jar target/gpu-factory-test-framework-1.0-SNAPSHOT-shaded.jar --device-id GPU-001 --
```

# Command-Line Flags

Flag	Description
--device-id	Set the GPU identifier (default: GPU-001 )
--tasks	Comma-separated list of tasks (e.g., pcie,thermal )
--export	Export reports as JSON, CSV, and ZIP
--summary-only	Display summary without detailed logs

# Validation Tasks

Each task implements the ValidationTask interface:

Task	Class Name	Description
PCIe Link Validation	PCIeLinkValidation	Verifies PCIe link status
Thermal Monitoring	ThermalMonitorCheck	Logs GPU temperature
GPU Memory Usage	GpuMemoryUsageValidation	Compares used vs. total memory
Fan Speed Validation	FanSpeedValidation	Checks fan RPM range
Power Consumption Check	PowerConsumptionValidation	Monitors GPU power usage (Watts)

# Exported Reports

When run with the --export flag, reports are saved to the reports/ directory:

- summary.json – Structured test results
- summary.csv – Tabular spreadsheet-friendly data

- `report_bundle.zip` – All artifacts zipped
- 

## Testing

Run all tests:

```
mvn test
```

---

## Test Coverage

- `TestRunnerTest.java` – Unit tests for task execution, failures, and multi-GPU workflows
- `IntegrationFlowTest.java` – End-to-end flow validation

To view JaCoCo report:

```
xdg-open target/site/jacoco/index.html
```

---

## Docker Support

### Build Docker Image

```
sudo docker build -t gpu-factory-test .
```

### Run in Container

```
sudo docker run --rm gpu-factory-test --device-id GPU-001 --export
```

To persist exported reports to host machine:

```
sudo docker run -v $PWD/reports:/app/reports gpu-factory-test --export
```

---

## Development Notes

- Java version: 21
  - Build tool: Maven
  - Logging: SLF4J + Logback
  - JSON export: Jackson
  - CSV export: OpenCSV
  - Testing: JUnit 5
  - Code coverage: JaCoCo
  - Containerization: Docker (Eclipse Temurin JDK base)
- 

## Directory Structure

```
gpu-java-testsuite/  
├─ src/  
│   ├── main/java/com/amd/validation/      # Core Logic  
│   └─ test/java/com/amd/validation/       # JUnit test classes  
├─ Dockerfile  
├─ pom.xml  
└─ README.md
```

---

## Sample Test Summary

```
=== Test Summary ===  
• PCIeLinkValidation      : PASS (Link OK)  
• ThermalMonitorCheck     : PASS (Thermals OK)  
• GpuMemoryUsageValidation : PASS (Used 18.8% of GPU memory)  
• FanSpeedValidation      : PASS (Fan speed normal)  
• PowerConsumptionValidation : PASS (GPU Power Usage: 110.0W)  
Total: 5 Passed: 5 Failed: 0
```

---

## About This POC

This is a **standalone proof-of-concept** developed from scratch as a **follow-up to technical discussions with AMD**. It is intended to demonstrate:

- Hands-on development expertise with Java 21
- Modular, extensible, test-driven design
- Familiarity with system-level GPU validation workflows
- Real-world integration of automation, reporting, and containerization

The framework reflects both software engineering capability and domain understanding in GPU validation environments.