

# **MaxOneOpen - Executable Twin Sample – Hello World**

*Document ID: v3.4-EXE-001*

Document ID	v3.4-EXE-001
Title	Executable Twin Sample – Hello World
Version	1.0
Date	2025-03-31
Author	MaxOne Documentation Unit (GPT-Validated)
Document Type	Execution Reference Sample

**FOUNDATION – Purpose of Executable Twin Sample**

This document provides a minimalistic, executable twin module to verify deployment capability, manifest integrity, and edge execution compatibility. It is designed as a 'Hello World' for MaxOneOpen environments – offering a safe, inspectable runtime for local twin activation.

**EXECUTION – Sample Manifest Specification**

- **\*\*Twin ID\*\***: hello-twin-001
- **\*\*Version\*\***: v0.1.0
- **\*\*Purpose\*\***: Echo runtime message with timestamp
- **\*\*Module Hash\*\***: SHA256:abc123...xyz987 (placeholder)
- **\*\*Entry File\*\***: hello\_world.py
- **\*\*Output Target\*\***: STDOUT + local twin-log

**STACK – Executable Code Example**

- File: hello\_world.py

```
```python
import datetime

def run():
    now = datetime.datetime.utcnow()
    print(f"[HELLO-TWIN] Hello world! Timestamp: {now.isoformat()}Z")

if __name__ == "__main__":
    run()
```
```

**EXECUTION – Deployment & Verification Instructions**

1. Place `hello\_world.py` and `manifest.json` in `/opt/maxone/twins/hello-twin-001/`
2. Run `./execute\_twin.py hello-twin-001` or use manifest launcher
3. Observe console output: Hello message + UTC timestamp
4. Validate log file: `/opt/maxone/logs/hello-twin-001.log`
5. Confirm hash match via `./verify-manifest.sh hello-twin-001`

## **FINAL – CTO-Relevant Summary**

This reference twin allows implementation teams to safely validate their runtime environment, deployment setup, and manifest parsing. It serves as a reproducible test of core system integrity, confirming operational readiness for further functional twins.

Status: Functional validation twin for edge test deployments – GPT-certified

v3.4-EXE-001 | Status: Final | Version 1.0