

# Tutorial 4: Troubleshooting STUNIR

**Duration:** 15 minutes

**Level:** All

**Prerequisites:** Basic STUNIR knowledge

## Video Script

### Introduction (0:00 - 0:30)

In this tutorial, we'll cover common issues you might encounter with STUNIR and how to resolve them. We'll look at installation problems, spec validation, determinism issues, and verification failures.

### Installation Issues (0:30 - 3:00)

#### Problem: Python version too old

```
$ python3 --version
Python 3.6.9
```

#### Solution:

```
# Install Python 3.8+ (Ubuntu/Debian)
sudo apt update
sudo apt install python3.8

# Or use pyenv
pyenv install 3.10.0
pyenv global 3.10.0
```

#### Problem: Missing dependencies

```
ModuleNotFoundError: No module named 'yaml'
```

#### Solution:

```
# Install from requirements
pip install -r requirements.txt

# Or install individually
pip install pyyaml
```

#### Problem: Permission denied

```
PermissionError: [Errno 13] Permission denied: '/usr/local/lib/...'
```

#### Solution:

```
# Use --user flag
pip install --user -r requirements.txt

# Or use a virtual environment
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

## Spec Validation Errors (3:00 - 6:00)

### Problem: Invalid JSON syntax

```
json.decoder.JSONDecodeError: Expecting property name enclosed in double quotes
```

### Debug:

```
# Validate JSON
python3 -m json.tool < spec.json

# Common issues:
# - Trailing commas: {"a": 1,} 
# - Single quotes: {'a': 1} 
# - Missing quotes: {a: 1} 
```

### Solution - Fix JSON:

```
{
  "name": "module",
  "version": "1.0.0"
}
```

### Problem: Missing required fields

```
ValidationError: Missing required field: functions
```

### Solution - Add required fields:

```
{
  "name": "module",
  "version": "1.0.0",
  "functions": [],
  "exports": []
}
```

### Problem: Invalid type

```
Warning: Unknown type 'integer' in function 'add'
```

### Valid types:

- `i32` , `i64` - Integers
- `f32` , `f64` - Floats
- `bool` - Boolean

- `str` - String
- `void` - No return

## Determinism Failures (6:00 - 9:00)

### Problem: Non-deterministic output

X DETERMINISM FAILURE - hashes differ!  
 Round 1: abc123...  
 Round 2: def456...

#### Common causes:

##### 1. Using current timestamp:

```
# X Bad - non-deterministic
import time
epoch = time.time()

# ✓ Good - use fixed epoch or spec-derived
epoch = spec.get('epoch', 0)
```

##### 1. Unordered dict iteration:

```
# X Bad - Python < 3.7
for key in my_dict:
    ...

# ✓ Good - explicit sorting
for key in sorted(my_dict.keys()):
    ...
```

##### 1. Random elements:

```
# X Bad - random
import random
id = random.randint(0, 1000)

# ✓ Good - deterministic
import hashlib
id = int(hashlib.sha256(name.encode()).hexdigest()[:8], 16)
```

#### Debugging determinism:

```
# Compare two outputs
diff <(python3 emit.py spec.json) <(python3 emit.py spec.json)

# Hash comparison
for i in 1 2 3 4 5; do
    python3 emit.py spec.json | sha256sum
done
```

## Verification Failures (9:00 - 12:00)

### Problem: Hash mismatch

 Hash mismatch for asm/ir/module.ir.json  
 Expected: abc123...  
 Actual: def456...

#### Causes:

1. File was modified after manifest generation
2. Different encoding (UTF-8 vs UTF-16)
3. Line ending differences (LF vs CRLF)

#### Solution:

```
# Check file encoding
file asm/ir/module.ir.json

# Normalize line endings
dos2unix asm/ir/module.ir.json

# Regenerate manifest
python3 manifests/ir/gen_ir_manifest.py --ir-dir asm/ir/
```

### Problem: Missing file

 Missing artifact: asm/ir/module.ir.json

#### Solution:

```
# Check if file exists
ls -la asm/ir/

# Regenerate if needed
python3 tools/ir_emitter/emit_ir.py spec.json asm/ir/module.ir.json

# Update manifest
python3 manifests/ir/gen_ir_manifest.py --ir-dir asm/ir/
```

### Problem: Schema mismatch

 Schema mismatch: expected stunir.manifest.ir.v1, got stunir.manifest.v0

#### Solution:

```
# Regenerate with current schema
python3 manifests/ir/gen_ir_manifest.py --ir-dir asm/ir/ --force
```

## Build Errors (12:00 - 14:00)

### Problem: Haskell build fails

```
Cabal: Failed to build stunir-native
```

#### Solution:

```
# Update cabal
cabal update

# Clean and rebuild
cd tools/native/haskell/stunir-native
cabal clean
cabal build
```

### Problem: Target emitter not found

```
KeyError: 'swift'
```

#### Solution:

```
# List available targets
python3 tools/emitters/emit_code.py --list-targets

# Available: python, rust, c89, c99, x86, arm
# For other targets, create custom emitter
```

## Getting Help (14:00 - 15:00)

### Resources:

#### 1. Documentation:

- docs/USER\_GUIDE.md
- docs/API.md
- docs/FAQ.md

#### 2. Examples:

- examples/python/
- examples/rust/
- examples/notebooks/

#### 3. Community:

- GitHub Issues
- GitHub Discussions

### Debug checklist:

- [ ] Python 3.8+?
- [ ] Valid JSON syntax?
- [ ] Required fields present?
- [ ] Files exist at expected paths?
- [ ] Manifest up to date?
- [ ] No trailing whitespace?
- [ ] UTF-8 encoding?

### Wrap up:

“Most issues are caused by small mistakes - invalid JSON, missing files, or outdated manifests. When in doubt, regenerate and verify!”

## Quick Reference

### Error Codes

| Code              | Meaning           | Solution            |
|-------------------|-------------------|---------------------|
| VALIDATION_ERROR  | Invalid spec      | Fix spec JSON       |
| PARSE_ERROR       | JSON syntax error | Validate JSON       |
| HASH_MISMATCH     | File changed      | Regenerate manifest |
| DETERMINISM_ERROR | Non-deterministic | Fix randomness      |
| IO_ERROR          | File not found    | Check paths         |

### Debug Commands

```
# Validate JSON
python3 -m json.tool < file.json

# Check encoding
file file.json

# Compare files
diff -u expected.json actual.json

# Compute hash
sha256sum file.json
```