

Migration Guide: v1.x to v2.x

This guide covers migrating from STUNIR 1.x to 2.x.

Overview

STUNIR 2.0 is a major release with significant improvements:

- Multi-target support (Python, Rust, C, Assembly)
- Manifest system for artifact tracking
- Provenance tracking
- Improved determinism guarantees

Breaking Changes

1. Spec Schema Changes

Before (v1.x):

```
{
  "module": "mymodule",
  "funcs": [
    {"name": "add", "args": ["int", "int"], "ret": "int"}
  ]
}
```

After (v2.x):

```
{
  "name": "mymodule",
  "version": "1.0.0",
  "functions": [
    {
      "name": "add",
      "params": [{"name": "a", "type": "i32"}, {"name": "b", "type": "i32"}],
      "returns": "i32"
    }
  ],
  "exports": ["add"]
}
```

Migration script:

```

import json

def migrate_spec(old_spec):
    """Migrate v1 spec to v2 format."""
    new_spec = {
        "name": old_spec.get("module", "unnamed"),
        "version": "1.0.0",
        "functions": [],
        "exports": []
    }

    for func in old_spec.get("funcs", []):
        new_func = {
            "name": func["name"],
            "params": [
                {"name": f"arg{i}", "type": map_type(t)}
                for i, t in enumerate(func.get("args", []))
            ],
            "returns": map_type(func.get("ret", "void"))
        }
        new_spec["functions"].append(new_func)
        new_spec["exports"].append(func["name"])

    return new_spec

def map_type(old_type):
    """Map v1 types to v2 types."""
    mapping = {
        "int": "i32",
        "long": "i64",
        "float": "f32",
        "double": "f64",
        "boolean": "bool",
        "string": "str"
    }
    return mapping.get(old_type, old_type)

```

2. IR Format Changes

Before (v1.x):

```

{
  "ir": {...},
  "hash": "abc123"
}

```

After (v2.x):

```

{
  "ir_version": "1.0.0",
  "ir_epoch": 1738000000,
  "ir_spec_hash": "abc123...",
  "module": {...},
  "functions": [...]
}

```

Migration: Regenerate all IR files from updated specs.

3. Receipt Format Changes

Receipts now include more metadata and use a different schema.

Migration: Regenerate all receipts and manifests.

4. CLI Changes

v1.x Command	v2.x Command
<code>stunir gen</code>	<code>python3 tools/ir_emitter/emit_ir.py</code>
<code>stunir emit</code>	<code>python3 tools/emitters/emit_code.py</code>
<code>stunir verify</code>	<code>./scripts/verify_strict.sh</code>

5. Type System Changes

v1.x Type	v2.x Type
<code>int</code>	<code>i32</code>
<code>long</code>	<code>i64</code>
<code>float</code>	<code>f32</code>
<code>double</code>	<code>f64</code>
<code>boolean</code>	<code>bool</code>
<code>string</code>	<code>str</code>

Step-by-Step Migration

Step 1: Backup Current Setup

```
# Create backup
cp -r stunir stunir-v1-backup
cp -r specs specs-v1-backup
cp -r output output-v1-backup
```

Step 2: Update STUNIR

```
# Get v2.x
git fetch origin
git checkout v2.0.0

# Or clone fresh
git clone -b v2.0.0 https://github.com/stunir/stunir.git stunir-v2
```

Step 3: Migrate Specs

```
# Run migration script
python3 scripts/migrate_specs.py specs-v1/ specs-v2/

# Verify
for f in specs-v2/*.json; do
    python3 -m json.tool < "$f" > /dev/null && echo "OK: $f" || echo "ERROR: $f"
done
```

Step 4: Regenerate IR

```
# Generate new IR
for spec in specs-v2/*.json; do
    name=$(basename "$spec" .json)
    python3 tools/ir_emitter/emit_ir.py "$spec" "output/ir/${name}.ir.json"
done
```

Step 5: Generate Manifests

```
# Generate manifests
python3 manifests/ir/gen_ir_manifest.py --ir-dir output/ir/
```

Step 6: Verify

```
# Run verification
./scripts/verify_strict.sh --strict
```

Common Issues

Issue: “Unknown type ‘int’”

Cause: Using v1 type names.

Fix: Update to v2 type names (see Type System Changes).

Issue: “Missing required field: version”

Cause: Spec missing version field.

Fix: Add `"version": "1.0.0"` to spec.

Issue: Hash mismatch after migration

Cause: v2 uses different canonicalization.

Fix: Regenerate all IR and manifests. This is expected.

Verification Checklist

- [] All specs converted to v2 format
- [] All IR files regenerated
- [] All manifests regenerated
- [] Verification passes (`verify_strict.sh --strict`)
- [] Determinism verified (same input = same output)

- [] Tests passing

Rollback

If migration fails:

```
# Restore backup
rm -rf stunir
mv stunir-v1-backup stunir
mv specs-v1-backup specs
mv output-v1-backup output
```

Getting Help

If you encounter issues:

1. Check the [FAQ](#) (../FAQ.md)
2. Search [GitHub Issues](https://github.com/stunir/stunir/issues) (https://github.com/stunir/stunir/issues)
3. Open a new issue with the `migration` label