STUNIR Ada SPARK Migration Summary

Date: January 30, 2026
Status: ✅ COMPLETED

Overview

STUNIR has been migrated from Python-first to **Ada SPARK-first** architecture. Ada SPARK is now the PRIMARY and DEFAULT implementation language for all STUNIR tools.

Migration Scope

✅ Created Ada SPARK Tools

Tool	Specification	Implementation	Entry Point
Spec to IR	`stunir_spec_to_ir.ads`	`stunir_spec_to_ir.adb`	`stunir_spec_to_ir_main.adb`
IR to Code	`stunir_ir_to_code.ads`	`stunir_ir_to_code.adb`	`stunir_ir_to_code_main.adb`

Location: `tools/spark/`

Build Command:

```
cd tools/spark
gprbuild -P stunir_tools.gpr
```

Binaries:
- `tools/spark/bin/stunir_spec_to_ir_main`
- `tools/spark/bin/stunir_ir_to_code_main`

✅ Updated Documentation

File	Changes
`AI_START_HERE.md`	Added Ada SPARK as primary language, tool locations
`ENTRYPOINT.md`	Tool priority table, updated navigation order
`README.md`	Critical warning banner about Ada SPARK default
`tools/spark/README.md`	Complete Ada SPARK tools documentation

✅ Updated Build Scripts

Script	Changes
`scripts/build.sh`	Detection priority: SPARK → Native → Python → Shell
`scripts/verify.sh`	Ada SPARK verifier support

✅ Updated Workflow System

File	Changes
`stunir_workflow/workflow.py`	Defaults to Ada SPARK, Python fallback with warnings
`stunir_workflow/README.md`	Documentation for Ada SPARK usage

✅ Python Reference Implementation Markers

Added warning headers to key Python tools:
- `tools/spec_to_ir.py`
- `tools/ir_to_code.py`

Tool Priority (New Default)

```
1. Ada SPARK (PRIMARY)     → tools/spark/bin/*
2. Native (Rust/Haskell)   → tools/native/*
3. Python (REFERENCE ONLY) → tools/*.py
4. Shell (Minimal)         → scripts/lib/*
```

Usage Examples

Building with Ada SPARK (Default)

```
# Automatic detection (prefers Ada SPARK)
./scripts/build.sh

# Explicit Ada SPARK
STUNIR_PROFILE=spark ./scripts/build.sh
```

Direct Tool Usage

```
# Spec to IR
./tools/spark/bin/stunir_spec_to_ir_main \
  --spec-root spec/ \
  --out asm/spec_ir.json

# IR to Code
./tools/spark/bin/stunir_ir_to_code_main \
  --input asm/spec_ir.json \
  --output output.py \
  --target python
```

Fallback to Python (NOT Recommended)

```
# Only if Ada SPARK unavailable
STUNIR_PROFILE=python ./scripts/build.sh
```

Why Ada SPARK?

1. **Formal Verification** - SPARK proofs guarantee absence of runtime errors
2. **Determinism** - Predictable execution for reproducible builds
3. **Safety** - Strong typing prevents entire classes of bugs
4. **DO-178C Compliance** - Industry standard for safety-critical systems
5. **Performance** - Native compilation, no interpreter overhead

Files Changed

New Files

- `tools/spark/stunir_tools.gpr`
- `tools/spark/README.md`
- `tools/spark/src/stunir_spec_to_ir.ads`
- `tools/spark/src/stunir_spec_to_ir.adb`
- `tools/spark/src/stunir_spec_to_ir_main.adb`
- `tools/spark/src/stunir_ir_to_code.ads`
- `tools/spark/src/stunir_ir_to_code.adb`
- `tools/spark/src/stunir_ir_to_code_main.adb`

Modified Files

- `AI_START_HERE.md`
- `ENTRYPOINT.md`
- `README.md`
- `scripts/build.sh`
- `scripts/verify.sh`
- `tools/spec_to_ir.py` (added reference marker)
- `tools/ir_to_code.py` (added reference marker)
- `stunir_workflow/workflow.py`
- `stunir_workflow/README.md`

Git Commit

```
commit 598e105 - Migrate STUNIR to Ada SPARK as primary implementation language
```

Next Steps

1. **Build Ada SPARK tools** on target systems:
   `bash`
   `   cd tools/spark && gprbuild -P stunir_tools.gpr`

2. **Run SPARK proofs** for formal verification:
   `bash`
   `   cd tools/spark && gnatprove -P stunir_tools.gpr --level=2`

3. **Update CI/CD** to build and use Ada SPARK tools by default

4. **Consider adding** Ada SPARK versions of other Python tools as needed

Notes

- Python files remain in the repository for reference and readability
- All Python tool files now include headers marking them as reference implementations
- The workflow system automatically warns when falling back to Python
- Build scripts provide clear messaging about which implementation is being used