

# STUNIR API Versioning Guide

This document describes the API versioning system used in STUNIR to ensure backward compatibility and smooth migrations.

## Version Numbering Scheme

STUNIR follows [Semantic Versioning 2.0.0](https://semver.org/) (<https://semver.org/>):

MAJOR.MINOR.PATCH

- **MAJOR**: Incompatible API changes
- **MINOR**: Backward-compatible functionality additions
- **PATCH**: Backward-compatible bug fixes

## Current Version

- **STUNIR Core**: 1.0.0
- **IR Schema**: stunir.ir.v1
- **Manifest Schema**: stunir.manifest.v1

## API Stability Guarantees

### Stable APIs

APIs marked as stable will maintain backward compatibility within the same MAJOR version:

API	Status	Since	Notes
canonical_json()	Stable	1.0.0	RFC 8785 compliant
compute_sha256()	Stable	1.0.0	SHA-256 hex output
BaseManifestGenerator	Stable	1.0.0	Base class for manifests
BaseManifestVerifier	Stable	1.0.0	Base class for verification

### Experimental APIs

APIs marked as experimental may change without notice:

API	Status	Since	Notes
emit_dcbor()	Experimental	0.9.0	dCBOR encoding
TargetEmitter	Experimental	0.9.0	Code generation

# Schema Versioning

---

## IR Schema

```
{
  "schema": "stunir.ir.v1",
  "version": "1.0.0",
  "module": "example"
}
```

## Manifest Schema

```
{
  "schema": "stunir.manifest.ir.v1",
  "manifest_epoch": 1706400000,
  "manifest_hash": "abc123..."
}
```

# Deprecation Policy

---

## Timeline

1. **Deprecation Announcement:** Feature marked deprecated in release notes
2. **Warning Period:** 2 minor releases with deprecation warnings
3. **Removal:** Feature removed in next MAJOR release

## Deprecation Markers

### Python

```
import warnings
from typing import Any

def deprecated(message: str, version: str):
    """Decorator to mark functions as deprecated."""
    def decorator(func):
        def wrapper(*args, **kwargs):
            warnings.warn(
                f"{func.__name__} is deprecated since {version}: {message}",
                DeprecationWarning,
                stacklevel=2
            )
            return func(*args, **kwargs)
        wrapper.__doc__ = f"DEPRECATED ({version}): {message}\n\n{func.__doc__}"
        return wrapper
    return decorator

# Usage
@deprecated("Use compute_sha256() instead", "1.1.0")
def old_hash_function(data: bytes) -> str:
    """Compute hash using old algorithm."""
    pass
```

## Rust

```
/// Compute hash using old algorithm.
///
/// # Deprecated
///
/// This function is deprecated since 1.1.0.
/// Use [`compute_sha256`] instead.
#[deprecated(since = "1.1.0", note = "Use compute_sha256() instead")]
pub fn old_hash_function(data: &[u8]) -> String {
    // Implementation
}
```

## Haskell

```
{-# DEPRECATED oldHashFunction "Use computeSha256 instead (since 1.1.0)" #-}
oldHashFunction :: ByteString -> String
```

# Migration Guides

## Version Migration Template

```
# Migrating from X.Y to X.Z

## Breaking Changes

- `function_name()` signature changed
- `ClassName` renamed to `NewClassName`

## New Features

- Added `new_function()`
- New parameter `option` on `existing_function()`

## Migration Steps

1. Replace `old_function()` with `new_function()`
2. Update imports: `from new_module import ...`
3. Run tests to verify

## Compatibility Layer

For gradual migration, use the compatibility layer:

```python
from stunir.compat.v1 import old_function # Deprecated wrapper
```
```

```
## API Changelog

### Version 1.0.0 (2026-01-28)

#### Added

- `manifests.base.canonical_json()` - RFC 8785 JSON canonicalization
- `manifests.base.compute_sha256()` - SHA-256 hashing
- `manifests.base.BaseManifestGenerator` - Base manifest generator
- `manifests.base.BaseManifestVerifier` - Base manifest verifier
- IR manifest generation (`manifests.ir`)
- Receipts manifest generation (`manifests.receipts`)
- Targets manifest generation (`manifests.targets`)
- Pipeline manifest generation (`manifests.pipeline`)

#### Schema Versions

Schema	Version	Status
stunir.ir.v1	1.0.0	Stable
stunir.manifest.ir.v1	1.0.0	Stable
stunir.manifest.receipts.v1	1.0.0	Stable
stunir.manifest.targets.v1	1.0.0	Stable
stunir.manifest.pipeline.v1	1.0.0	Stable

## Version Compatibility Matrix

STUNIR Version	Python	Rust	Haskell	IR Schema
1.0.x	≥3.9	≥1.70	≥9.4	v1
0.9.x	≥3.8	≥1.65	≥9.2	v1 (beta)
```

## Detecting API Version

### Python

```
```python
import stunir
print(stunir.__version__) # "1.0.0"
print(stunir.API_VERSION) # "1"
```

## Rust

```
use stunir_native::VERSION;
println!("Version: {}", VERSION); // "0.1.0"
```

## Command Line

```
$ stunir --version
stunir 1.0.0

$ stunir-native --version
stunir-native 0.5.0.0
```