

# Fehleranalyse

## Versions-Mismatch Isaac Sim / Isaac Lab

### Problemstellung

Isaac Lab und Isaac Sim entwickeln sich unabhängig voneinander weiter. Wenn eine Szene oder ein Task mit einer neueren API-Version erstellt wurde, aber mit einer älteren Version von Isaac Sim oder Isaac Lab gestartet wird, brechen die Imports sofort ab. Umgekehrt können alte Szenen mit zu neuer API ebenfalls nicht geladen werden. Dieses Problem tritt oft nach Updates oder beim Wechsel zwischen Projekten auf.

### Typische Symptome

- `AttributeError: module 'omni.isaac.core' has no attribute 'World'`
- `ImportError: cannot import name ... from omni.isaac...`
- Abbruch direkt beim Start, bevor die Stage geladen wird.
- Tasks, die vorher funktionierten, laufen nach einem Update plötzlich nicht mehr.

### Häufige Ursachen

- Szene wurde für Isaac Lab 1.0+ gebaut, gestartet aber mit älterem Isaac Sim.
- Bestimmte Klassen oder Funktionsaufrufe existieren in der verwendeten Version nicht mehr oder heißen anders.
- Versionsnummern von Isaac Lab, Isaac Sim und PyPI-Paketen sind nicht aufeinander abgestimmt.
- Unterschiedliche Environments: Szene läuft in Conda-Env A, gestartet wird aber Env B mit älteren Paketen.

### Minimalbeispiel

```
# Funktioniert in neuer Version, bricht aber in älterer sofort ab
from omni.isaac.core import World

world = World() # AttributeError in älteren Versionen
```

### Fix-Patterns

**Pattern A: Versionen pinnen**

- Exakte Versionen in `requirements.txt` oder `environment.yml` festhalten.
- Dokumentieren: Welche Isaac Sim und Isaac Lab Version wurde genutzt?

### Pattern B: Gemeinsame Updates

- Isaac Sim und Isaac Lab gleichzeitig auf kompatible Versionen aktualisieren.
- Release Notes der NVIDIA-Docs prüfen.

### Pattern C: API-Anpassungen

- Imports auf neue Namensräume anpassen, wenn sich API geändert hat.
- Alte Aufrufe durch neue ersetzen (`SimulationContext` statt `World` in älteren Labs).

### Pattern D: Environment-Management

- Nur ein aktives Env pro Projekt nutzen.
- Vor Start immer prüfen: `which python, pip list`.

## Debug-Playbook

1. Mit `pip show omni.isaac.lab` und `pip show omni.isaac.core` Versionen prüfen.
2. Isaac Sim Launcher öffnen und Version notieren.
3. Abgleich mit Dokumentation: Welche Versionen sind offiziell kompatibel?
4. Import-Test in Python-Shell:

```
import omni.isaac.core
print(dir(omni.isaac.core))
```

5. Falls Attribut fehlt -> API-Änderung oder falsche Version.
6. Lösung: Alle Versionen synchronisieren und Caches leeren.

## Quick-Checklist

- Isaac Lab und Isaac Sim Versionen dokumentiert und festgepinnt?
- Updates immer für beide Komponenten gleichzeitig?
- Imports an API-Änderungen angepasst?
- Richtiges Python-Env aktiv?
- Nach Änderungen Caches löschen und sauberen Restart machen?