

Technische Spezifikation der DAT-Datei zur Repräsentation von Bibliothek-Projektabhängigkeiten

1. Einleitung

Die vorliegende Spezifikation beschreibt das Format und die Struktur einer **DAT-Datei**, die zur textuellen Darstellung von Abhängigkeiten innerhalb eines Projekts dient. Die Datei folgt einem **hierarchischen Aufbau**, um eine klare und strukturierte Repräsentation von Software-Dependencies und zugehörigen Lizenzinformationen zu gewährleisten. Die einheitliche Struktur ermöglicht eine einfache maschinelle Verarbeitung sowie eine intuitive menschliche Lesbarkeit.

2. Dateiformat und allgemeine Struktur

Die Datei besteht aus einer zeilenbasierten Darstellung von Abhängigkeiten und zugehörigen Lizenzinformationen. **Jede Zeile** enthält **genau eine Information** und folgt einem vordefinierten Schema. **Hierarchische Beziehungen** zwischen den einzelnen Elementen werden durch **Einrückungen** ausgedrückt.

2.1 Formatierungskonventionen

- Jede Zeile entspricht einem logischen Element der Struktur.
- Trennzeichen innerhalb einer Zeile sind Doppelpunkte **(:)** oder doppelte Doppelpunkte **(::)**.
- **Hierarchische Strukturen** werden durch **Einrückungen** von Leerzeichen visualisiert.

3. Hierarchische Struktur und Einrückungsrichtlinien

- **Hauptelemente** (Dependency-Knoten): Diese Einträge befinden sich auf oberster Ebene und enthalten **keine führenden Leerzeichen**.
- **Verschachtelte Abhängigkeiten**: Jede tiefere Hierarchieebene wird durch eine Einrückung von genau **6 Leerzeichen** dargestellt.
- **Lizenzinformationen** (License-Knoten): Lizenzen sind untergeordnete Elemente von Abhängigkeiten und werden um **3 zusätzliche Leerzeichen** gegenüber dem zugehörigen Dependency-Knoten eingerückt.

4. Zeileninhalte und Formatierungsrichtlinien

4.1 Dependency-Deklaration

Jede Abhängigkeit wird durch eine Zeile mit folgender Struktur repräsentiert:

```
<groupId>:<artifactId>:<version>
```

Beispiel:

```
com.example:sample-library:1.2.3
```

4.2 Lizenzinformationen

Falls eine Abhängigkeit **Lizenzen** besitzt, werden diese in separaten, eingerückten Zeilen mit folgender Struktur aufgelistet:

```
<Lizenzname>::<Lizenz-URL>
```

Beispiel:

```
MIT License::https://opensource.org/licenses/MIT
```

5. Beispielstruktur einer DAT-Datei

```
3rdp_libs:sap-idoc:3.1.1
com.h2database:h2:2.2.220
  MPL 2.0::https://www.mozilla.org/en-US/MPL/2.0/
  EPL 1.0::https://opensource.org/licenses/eclipse-1.0.php
org.glassfish.jersey.core:jersey-container-jdk-http:3.1.2
  jakarta.inject:jakarta.inject-api:2.0.1
    Apache License,
    2.0::http://www.apache.org/licenses/LICENSE-2.0.html
```

6. Vorteile und Anwendungsfälle

- **Maschinelle Verarbeitung:** Dank der konsistenten Struktur können Software-Tools die Datei effizient einlesen und verarbeiten.
- **Einfache menschliche Lesbarkeit:** Die Einrückungen und klare Trennung von Informationen erleichtern die visuelle Erfassung der Abhängigkeiten.
- **Flexibilität:** Das Format unterstützt eine beliebige Verschachtelungstiefe und kann damit auch komplexe Dependency-Strukturen abbilden.