

# Datenerhaltung

## TextureSync

Version	1.2
Datum	25.04.19
Autor	Lukas Fürderer
Projektmitglieder	Hendrik Schutter, Lukas Fürderer, Robin Willmann, Jannik Seiler

# Inhaltsverzeichnis

1 Einleitung.....	3
2 Genereller Aufbau.....	3
3 Metadaten.....	3
3.1 Definitionen.....	3
4 Konsistenz.....	4
5 Changelog.....	5

# 1 Einleitung

Dieses Dokument beschreibt die Art und Weise, wie Texturen und deren Metadaten auf dem Server im Dateisystem abgelegt werden.

## 2 Genereller Aufbau

Alle Nutzdaten sind im aktuellen Arbeitsverzeichnis des Serverprozesses untergebracht. Auf diese Weise ist der Speicherort einfach konfigurierbar.

Die Texturdateien selbst liegen im Ordner `./textures/` und haben als Namen ihren SHA256-Hashwert ohne Dateiendung. Der Hashwert ist wie üblich hexadezimal angegeben und die enthaltenen Buchstaben sind klein geschrieben.

Alle Metadaten der Texturen sind in einer Datei unter `./collection.json` abgelegt.

## 3 Metadaten

Die Textur-Metadaten sind in der Collections-Datei so abgelegt, wie sie auch über das Netzwerk übertragen werden.

### 3.1 Definitionen

Für Basisdefinitionen siehe *Netzwerk-Protokoll 2.1*.

```
CollectionFile ::= {  
    textures: <Array von <Texture>>  
}
```

Die Datei `./collection.json` enthält genau ein Json-Objekt vom Typ `CollectionFile`.

## 4 Konsistenz

Um die Daten bei einem Serverabsturz konsistent zu halten und auch die Konsistenz eines einfachen Datei-Backups möglichst zu gewährleisten, muss die Collection-Datei atomar überschrieben werden. Hierzu erstellt der Server zunächst eine neue Datei unter dem Namen `./collection_new.json` und füllt diese mit allen notwendigen Daten. Als letzter Schritt wird mit einem `rename(2)` Syscall die `./collection.json` atomar durch die neue Datei ersetzt.

Beim Hinzufügen und Löschen von Texturen können die Datei selbst und der zugehörige Metadaten-Eintrag nicht gemeinsam atomar erstellt bzw. gelöscht werden. An dieser Stelle gilt die Grundregel: Eine Datei darf ohne Metadaten-Eintrag existieren, jedoch nicht umgekehrt.

Beim Hinzufügen einer Textur muss also zuerst die Textur-Datei selbst geschrieben werden, danach darf der Server die `collection.json` aktualisieren.

Umgekehrt muss der Server beim Löschen einer Textur zuerst den Eintrag aus der `collection.json` entfernen und diese neu schreiben, danach darf er die Textur-Datei löschen.

Um Datenmüll aufzuräumen muss der Server zusätzlich nach dem Start das Verzeichnis aller Texturen auslesen und alle nicht mehr referenzierten Texturdateien löschen. Umgekehrt löscht er auch Metadaten-Einträge, bei denen die Textur-Datei fehlt. Dieser Fall kann bei Backups im laufenden Betrieb auftreten.

## 5 Changelog

Version	Änderung
1.0	Dokument Datenerhaltung erstellt
1.1	Daten liegen im CWD statt unter /var/texturesync/
1.2	Entferne Basisdefinitionen, Verweise stattdessen bei Definitionen auf Netzwerkprotokoll, um Inkonsistenz zu vermeiden.