

# Wissenschaftliche Artikel mit Scrivener 3

Thomas Hildmann

30th May 2018

## 1 Warum?

*Warum sollte man wissenschaftliche Artikel mit Scrivener 3 schreiben?*

Für mich besteht die Antwort in den in Scrivener<sup>1</sup> enthaltenen Workflow-Unterstützungen, in Trennung zwischen Inhalt und Form und in der Möglichkeit, alle relevanten Informationen in Form von Texten, Webseiten, Bildern, MindMaps etc. in ein Projekt zu integrieren und jeder Zeit wieder beieinander zu haben. Aber es gibt noch viele weitere Gründe z.B. in diesem Blogposts zu lesen ist Vellguth [9].

## 2 Wie?

*Wie funktioniert das nun?*

Auf das Schreiben und Arbeiten in Scrivener 3 möchte ich an dieser Stelle nicht weiter eingehen. Für mich bestand eine große Hürde in der Verwendung vor allem darin, Scrivener 3 dazu zu bringen, ...

- (für mich akzeptablen) LaTeX Output zu generieren
- meine BibTeX Literaturdatenbank einzubeziehen

Um (dem zukünftigen) mir und anderen die Arbeit zu ersparen, das Rad neu zu erfinden, stelle ich die Basis-Schritte im folgenden zusammen.

Ich besitze ein funktionierendes LaTeX Workflow basierend auf TeXpad<sup>2</sup> und TeXLive<sup>3</sup>, sowie BibTeX<sup>4</sup> via BibDesk<sup>5</sup>. Mein Scrivener 3 Workflow soll nun so aussehen, dass ich...

- auf die BibTeX Datenbank zugreifen kann,
- ich den WYSIWYG-Editor von Scrivener benutzen kann,

---

<sup>1</sup><https://www.literatureandlatte.com/scrivener/overview>

<sup>2</sup><https://www.texpad.com/>

<sup>3</sup><https://tug.org/texlive/>

<sup>4</sup><http://www.bibtex.org/>

<sup>5</sup><https://bibdesk.sourceforge.io/>

- keine Änderungen am LaTeX-Code mehr nötig sind und
- am Ende ein PDF rausfällt.

Prinzipiell soll der Ablauf so aussehen:

1. Scrivener 3 generiert MultiMarkdown (MMD) Code mmd [2]
2. Scrivener 3 übersetzt diesen MMD in LaTeX
3. Der LaTeX-Code wird in TeXPad geöffnet und in PDF übersetzt

Der letzte Schritt ist theoretisch auch aus der Kommandozeile, über ein AppleScript o.ä. durchführbar. Scrivomatic scm [4] treibt das ganze auf die Spitze. Hilfe bei meiner Konfiguration habe ich vor allem in folgenden Blogposts gefunden: [5, 6, 1]

### 3 Die Konfiguration

Im folgenden beschreibe ich nun die Konfiguration innerhalb von Scrivener 3, die zu diesem Ergebnis führt.

Einmal konfiguriert, kann man sich einen Fachartikel natürlich auch als Template Lit [7, 5.3 Project Templates] anlegen und den ganzen Konfigurationsaufwand für alle folgenden Artikel sparen.

Ich versuche meine Einstellungen passend zum Artikel am Ende auch bereitzustellen, freue mich aber auch über Zusendungen von Weiterentwicklungen oder eigenen Lösungen.

Ein Großteil der Magie geschieht wie bei Scrivener üblich im Menüpunkt **Ablage - Kompilieren** (Alt-CMD E).

Hier wähle ich, wie oben Beschrieben die Methode **MultiMarkdown -> LaTeX (.tex)**. Das Scrivener Manual widmet dem MultiMarkdown, Pandoc und LaTeX ein ganzes Kapitel Lit [7, Chapter 21].

Statt das Rad neu zu erfinden, kopiere ich den Standard und ändere ihn nur an einigen Stellen ab (Format duplizieren und bearbeiten).

Es erscheint ein Dialog, der in etwa wie folgt aussieht

Bei Formatbezeichnung gebe ich **Scientific Paper** ein. Selbstverständlich kann hier jeder das wählen, was ihm am sinnvollsten erscheint. Auch ist es wahrscheinlich, dass man im Laufe der Zeit verschiedene Detailformate erzeugt für bestimmte Anwendungszwecke.

#### 3.1 MultiMarkdown Optionen

Die entscheidenden Schritte folgen unter der Auswahl Formatvorlagen und MultiMarkdown (MMD) Optionen. Hier geht es darum Scrivener Format-Vorlagen zu erzeugen, die auf MMD Befehle abgebildet werden können.

Neben **Beschriftung** für Bild- oder Tabellenunterschriften sind die folgenden Absatzformate relevant:

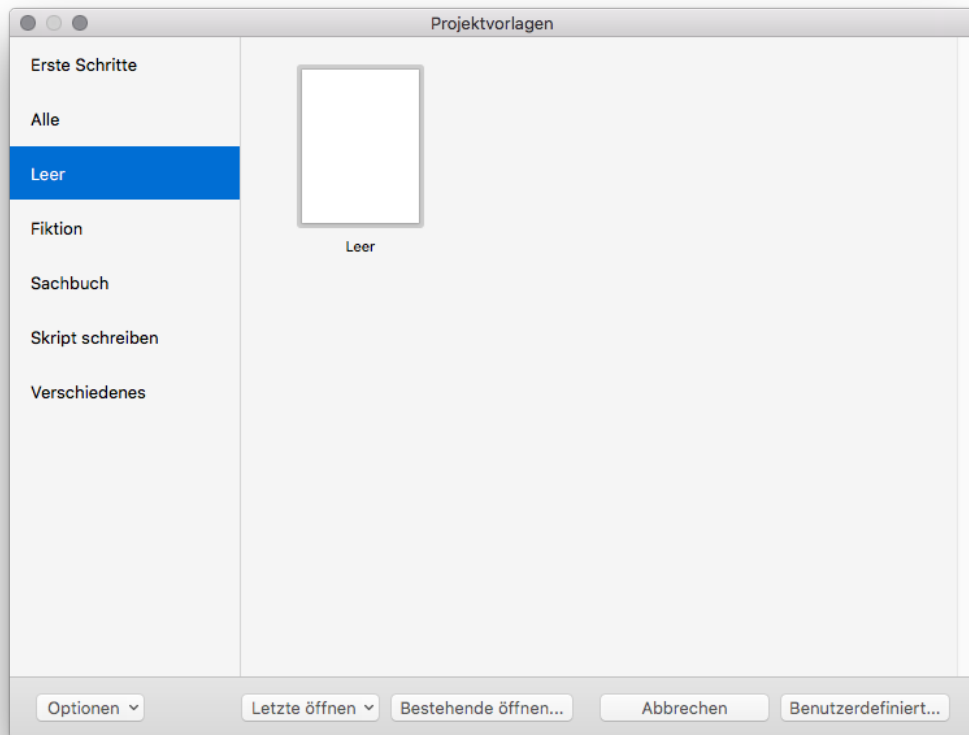


Figure 1: Projekt aus leerer Projektvorlage erstellen

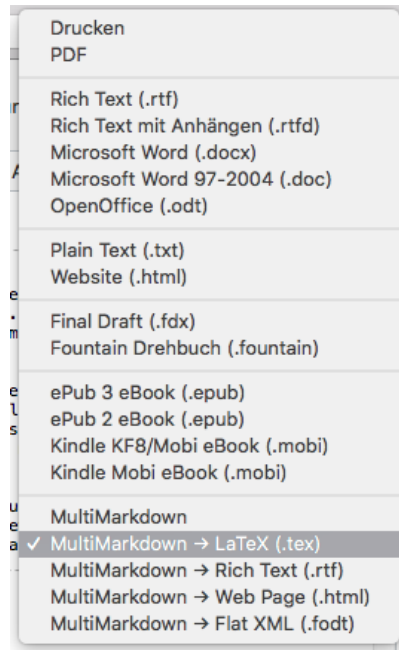


Figure 2: Kompilieren für MultiMarkdown -> LaTeX

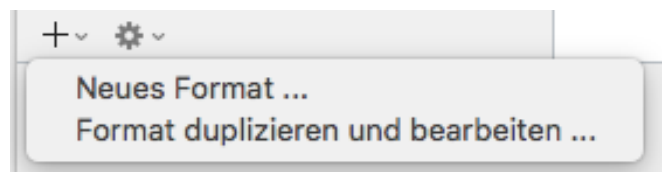


Figure 3: Format duplizieren und bearbeiten...

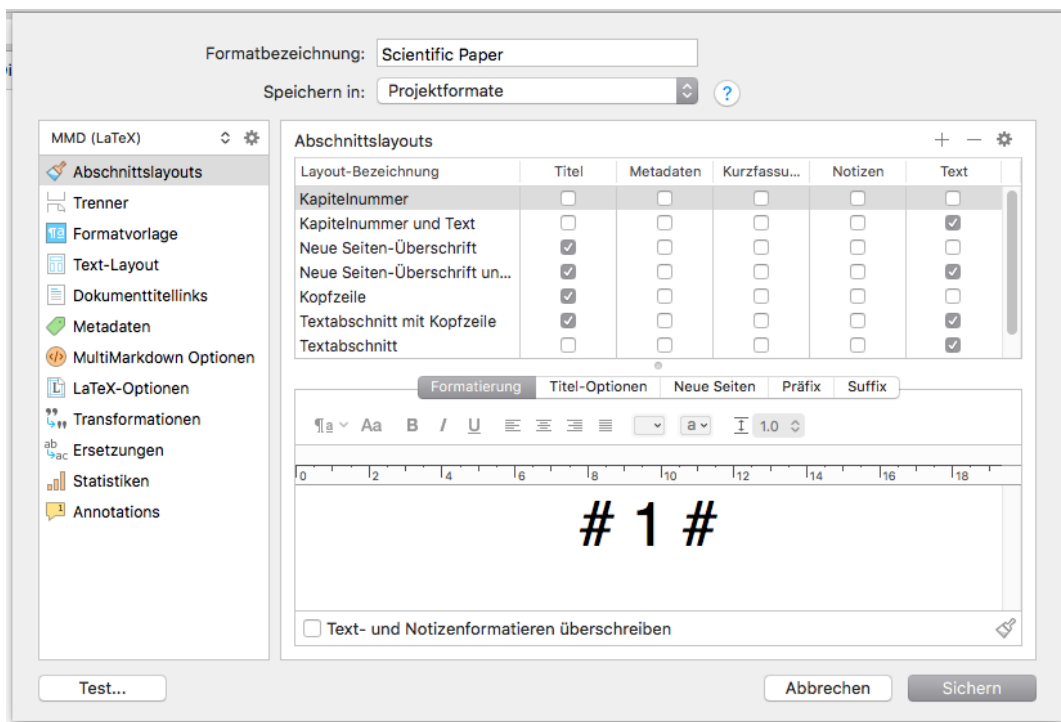


Figure 4: Dialog: Projektformate

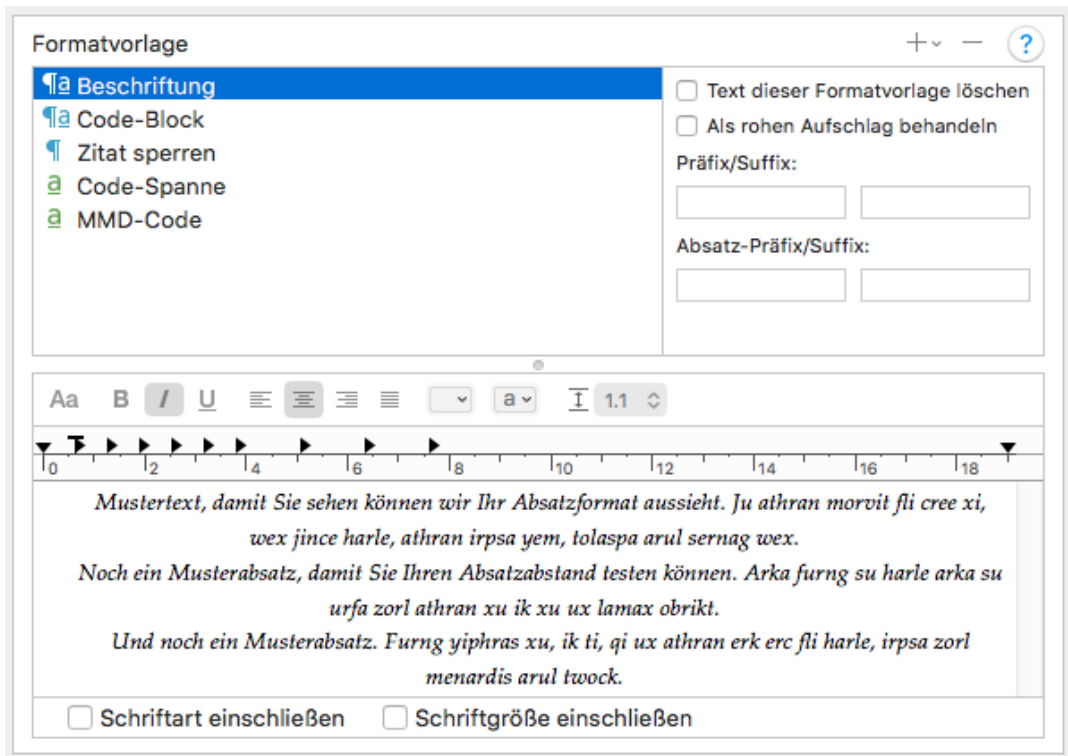


Figure 5: Formatvorlage Beschriftung

- Code-Block und
- Zitat sperren

Code-Block Vorlagen verwendet man, wenn man Source-Code im Text beschreiben möchte. Ein Absatz mit Source-Code wird in MMD mit drei **eingeleitet und mit drei** in einer neuen Zeile beendet. Das übernimmt für uns an dieser Stelle Scrivener, wenn wir hier die Formatvorlage definieren und (s.u.) entsprechend der MMD Option zuweisen.

Hier ein Beispiel für Code-Block:

```
tell application "Mail"
  activate
  tell front window to set zoomed to true
end tell
```

Den Begriff **Zitat sperren** habe ich aus irgendeiner Formatvorlage übernommen. Gerne kann hier ein griffigerer Name gewählt werden. Ein Block-Zitat wird in MMD dargestellt mit einer Zeile, die mit > beginnt. Auch das konvertiert für uns Scrivener bei geeigneter Konfiguration (s.u.).

Ein Beispiel für ein Zitat:

```
> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut quis diam in lacus
> ullamcorper venenatis. Ut vitae elit ex. Etiam turpis ipsum, euismod vel
> porta eu, porta at eros. Suspendisse nec elit vitae enim rhoncus pretium in et
> lectus. Phasellus convallis ullamcorper ante, quis ultrices libero consectetur
> sed. Integer placerat, lacus eget porta tincidunt, odio nunc interdum massa,
> eget euismod tellus tortor vitae diam. Nulla a consequat libero. Pellentesque
> sagittis, metus vitae feugiat maximus, mauris massa interdum erat, faucibus
> vulputate nulla quam ut diam. Lorem ipsum dolor sit amet, consectetur
> adipiscing elit.
```

Es folgen zwei sog. **Zeichenstile**:

- Code-Spanne und
- MMD-Code

Hier müssen wir selbst für den richtigen Präfix/Suffix im MMD-Code sorgen. Scrivener soll den Text “Als ruhen Aufschlag behandeln”. D.h. der Text wird unverändert in MMD übertragen.

Die **Code-Spanne** ist die kleine Schwester vom **Code-Block**. Er findet Einsatz, wenn nur einzelne Wörter eines Textes als Code interpretiert werden sollen, wie in diesem Abschnitt hier jeweils die Begriffe **Code-Spanne** und **Code-Block**.

Die Größte “Magie” stellt jedoch die Vorlage MMD-Code dar. Sie findet überall dort Einsatz, wo Scrivener den eingegebenen Text 1:1 an den MMD Interpreter durchreichen

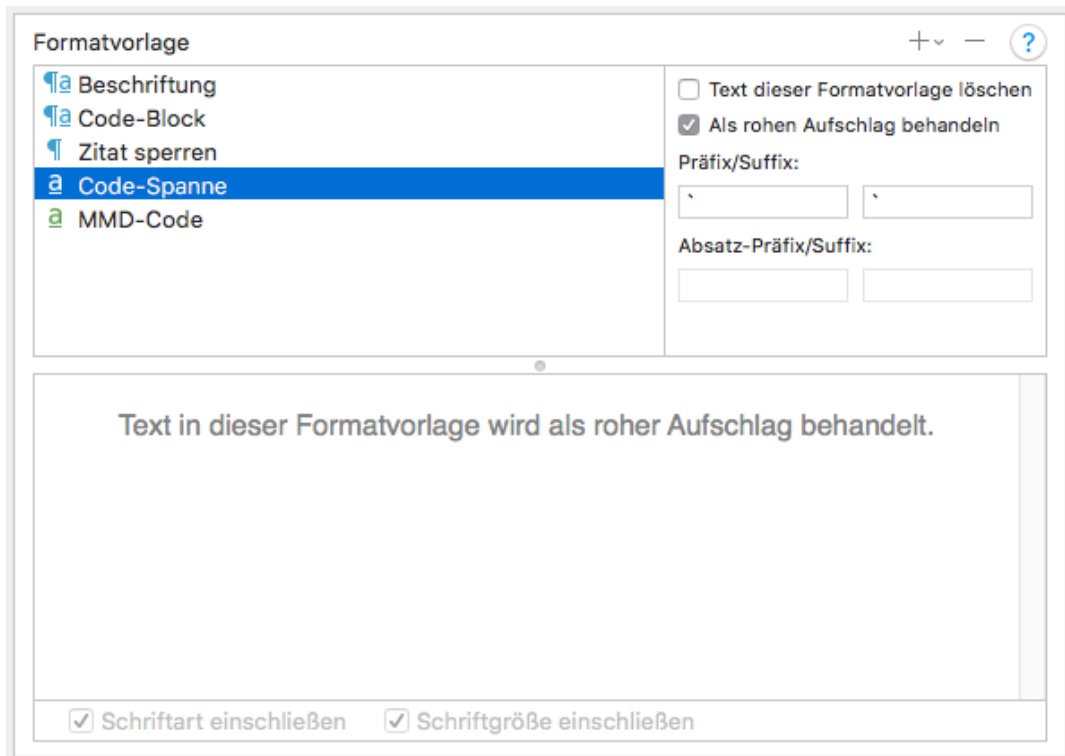


Figure 6: Formatvorlage Code-Spanne



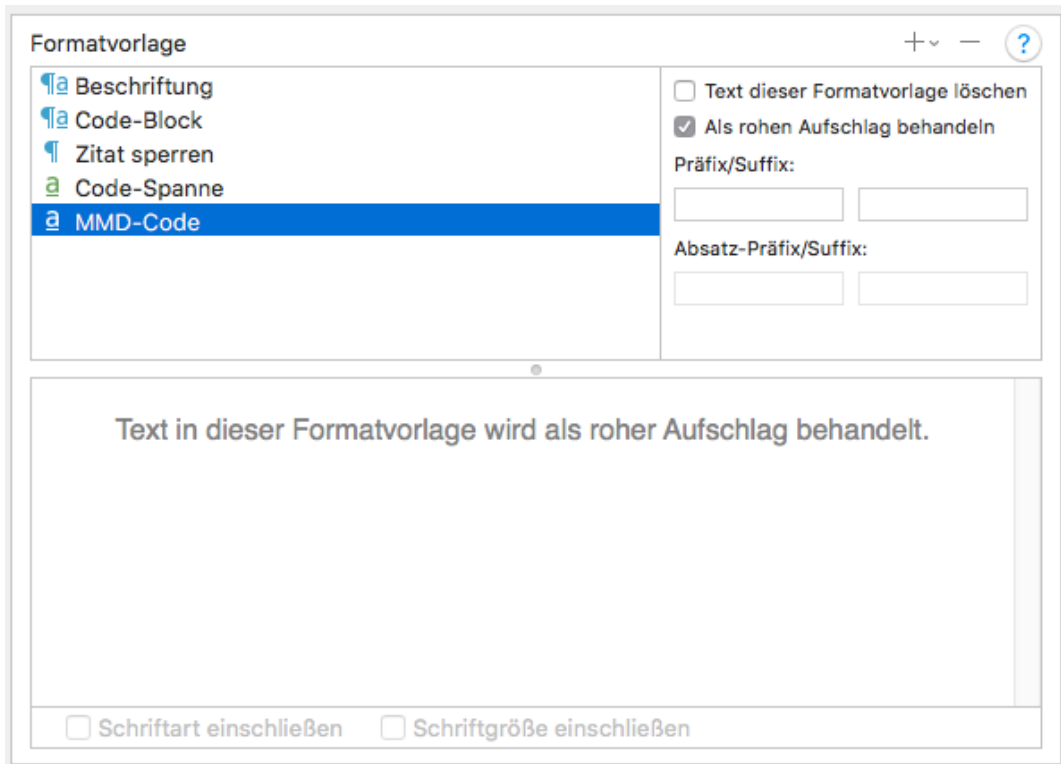


Figure 7: Formatvorlage MMD-Code

soll. Über diesen Trick kommt man dann auch an Funktion, wie Bibliotheksverweise heran aber prinzipiell auch alles andere, was MMD so im Werkzeugkasten hat, was Scrivener aber nicht automatisch generiert.

Bei MMD-Code benutze ich dabei die Funktion “Markierfeld um Text ziehen”, damit ich auch wirklich sehe, dass hier MMD-Code eingebettet ist und mich nicht über die Ausgaben wundere.

### 3.2 Formatvorlagen zwei Mal anlegen

Zu jeder Vorlage muss auch eine Formatvorlage im Editor erzeugt werden, damit die Formate auch im Text angesprochen werden können. Diese müssen *exakt genauso* heißen, weil sie sonst nicht richtig zugeordnet werden können.

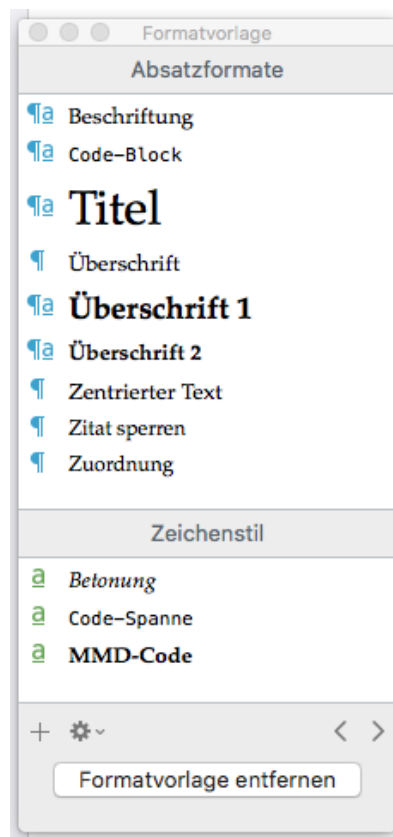


Figure 8: Definition der Formatvorlagen

### 3.3 MultiMarkdown Optionen

Die vorangegangenen Schritte sind eigentlich nur Vorbereitung für die Zuordnung der speziellen Formate in den MMD Optionen.

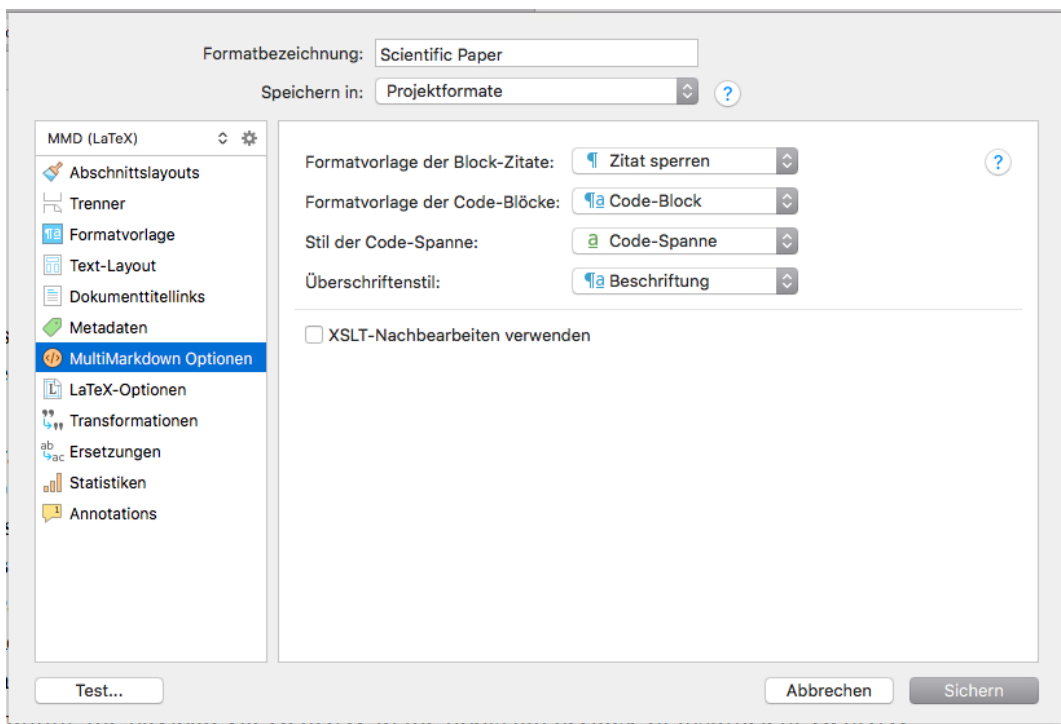


Figure 9: MultiMarkdown Optionen

Wir müssen nun die verschiedenen Formatvorlagen dann auch in den MMD Optionen zugewiesen werden. Wie, erklärt sich durch die fast Begriffe von selbst.

MMD Option	Formatvorlage
Formatvorlage Block-Zitate	Zitat sperren
Formatvorlage der Code-Blöcke	Code-Block
Stil der Code-Spanne	Code-Spanne
Überschriftenstil	Beschriftung

### 3.4 LaTeX - Optionen

Die vorherigen Einstellungen dienten dazu, dass Scrivener MMD Code erzeugen kann und über Formatvorlagen gesteuert werden kann, welche Textbereiche wie in MMD dargestellt werden. Wäre unser Ausgabeformat MMD, wären wir nun fertig. Da aus dem MMD Code jedoch in einem weiterem Schritt LaTeX Code generiert werden soll, geht es wie folgt weiter.

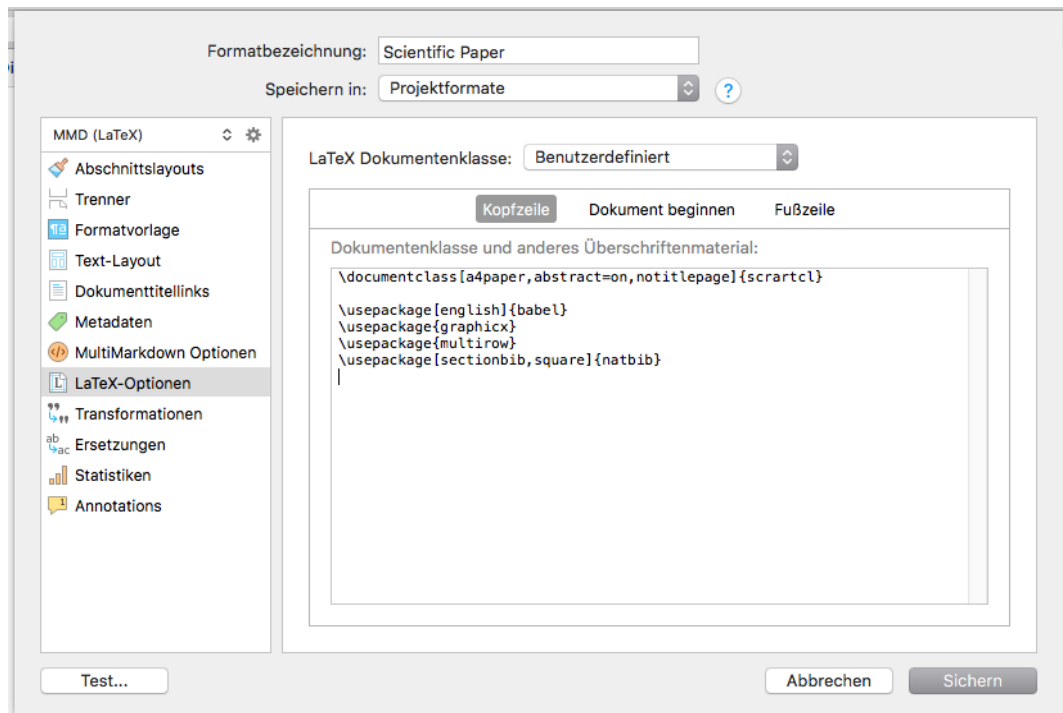


Figure 10: LaTeX-Optionen

Unter LaTeX-Optionen können fertige LaTeX-Einstellungen benutzt werden. Für meine wissenschaftlichen Artikel war jedoch leider nichts dabei, das ich so verwenden wollte, weshalb ich über die LaTeX Dokumentenklasse: Benutzerdefiniert eigenen LaTeX-Code verwende.

### 3.5 Kopfzeile

Alles, was unter Kopfzeile geführt wird, wird dem LaTeX Dokument vorangestellt. Hier kommen Befehle, wie `\documentclass` und die verwendeten `\usepacke` Befehle rein. Meine vollständige “Kopfzeile” sieht wie folgt aus:

```
\documentclass[a4paper,abstract=on,notitlepage]{scrartcl}

\usepackage[utf8]{inputenc}
\usepackage[ngerman,english]{babel}
\usepackage{ngerman}
\usepackage{graphicx}
\usepackage{multirow}
\usepackage[square,authoryear]{natbib}
\usepackage{multirow,tabulary}
\usepackage{booktabs}
\usepackage[breaklinks]{hyperref}
```

### 3.6 Dokument beginnen

```
\begin{document}

\title{\mytitle}
\author{\myauthor}

\maketitle
```

Wie man hier sieht, habe ich auf umständliche Art und Weise mich als Autoren an dieser Stelle fest encodiert. Besser wäre, wenn der Name aus den Projekteigenschaften übernommen werden würde. Da gab es aber irgendwelche Probleme. Wenn ich die beseitigt habe, reiche ich eine Korrektur nach. Für Hinweise bin ich natürlich auch sehr dankbar!

Wer möchte kann übrigens an dieser Stelle auch eine `abstract` Umgebung einbauen. Man kann dann unter Metadaten ein Feld `Abstract` einfügen, dessen Inhalt dann im Dokumentenkopf eingesetzt wird.

### 3.7 Fußzeile

```
\bibliographystyle{plainnat}
\bibliocommand

\end{document}
```

In der Fußzeile muss dann lediglich noch die BibTeX Referenzierung angehängt werden.

## 4 Für jedes Projekt

Das waren die Einstellungen, die man eigentlich nur einmal machen muss und die man dann unter `Format exportieren` im `Kompilieren...` Dialog speichern und wiederverwenden kann.

Für jedes Projekt [7, 5. All About Projects] müssen dann unter `Kompilieren...` immer noch folgende Metadaten erfasst werden:

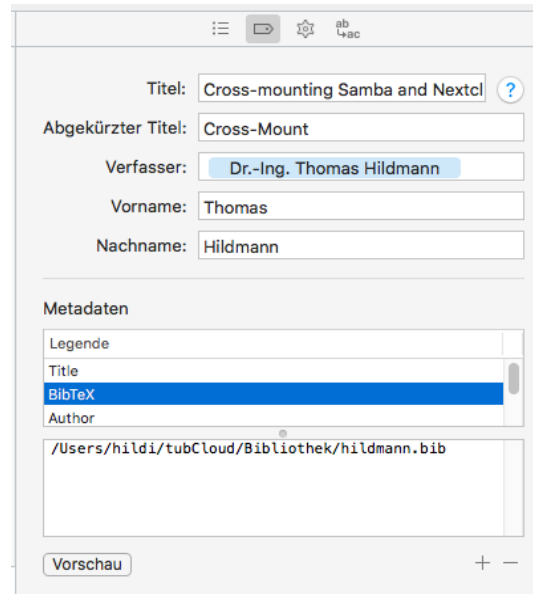


Figure 11: Kompilieren: Metadaten

Legende	Inhalt
Titel	<code>&lt; \$projecttitle &gt;</code>
BibTeX	<code>/Pfad/beispiel.bib</code>
Base Header Level	<code>3</code>
Author	<code>&lt; \$author &gt;</code>

BibTeX gibt den Pfad zur `.bib` Datei an, die für die Referenzen verwendet werden sollen. Hier kann man der Einfachheit halber auch relative Pfade, wie `../` etc. angeben.

Base Header Level gibt an, wieviele Überschriftenebenen verwendet werden sollen. Der Base Header Level muss zur `\documentclass` passen. So hat z.B. `scrartcl` kein `\chapter` definiert, so dass nur mit `\section`, `\subsection` etc. gearbeitet werden kann.

Dabei bedeuten die folgenden Level die dahinterstehenden LaTeX-Kommandos:

## 5 Tipps zur Nutzung

Hier noch einige Tipps für die Nutzung:

Level	LaTeX-Command
1	<code>\part{}</code>
2	<code>\chapter{}</code>
3	<code>\section{}</code>

- Bilder alle über Doppelklick noch mal neu benennen und in den Namen keine Leerzeichen, Umlaute oder andere Schweinereien verwenden
- Wenn man Bilder gleich beschriftet, würfelt Scrivener die Bilder bei der Ausgabe durcheinander (dies gilt in der Regel zu vermeiden).
- Nutzt man in Überschriften Umlaute, erzeugt Scrivener automatisch Labels in LaTeX, die ebenfalls diese Umlaute enthalten, was nicht zulässig ist. Workaround: “MMD-Code” benutzen und Überschrift in `# Überschrift #` Schreibweise einsetzen.
- Martin beschreibt in seinem Artikel [8], wie man sich das Leben mit BibTeX und LaTeX durch die Verwendung von Symlinks etwas leichter machen kann.
- Die App Store Version von Scrivener wird auf Grund der Regeln von Apple ohne Pandoc Unterstützung ausgeliefert [3]. Deshalb würde ich auf jeden Fall empfehlen, die Version direkt on Literature and Latte zu kaufen.

## References

- [1] Configuring scrivener to output latex. AbnormalData, January 2015. URL <https://abnormaldata.wordpress.com/2015/01/14/configuring-scrivener-latex/>.
- [2] Multimarkdown, December 2017. URL <http://fletcherpenney.net/multimarkdown/>.
- [3] Should i buy from your web store or from the mac app store? Literature and Latte Support, May 2018. URL <https://scrivener.tenderapp.com/help/kb/purchasing-and-installation/should-i-buy-from-your-web-store-or-from-the-mac-app-store>.
- [4] Scrivomatic: Scrivener and pandocomatic, May 2018. URL <https://github.com/iandol/scrivomatic>.
- [5] Tim Brandes. How to write your thesis in latex with scrivener 2, multimarkdown 3 and bibdesk, February 2012. URL <https://timbrandes.com/guide/2012/02/28/howto-write-your-thesis-in-latex-using-scrivener-2-multimarkdown-3-and-bibdesk/>.
- [6] Tim Brandes. Optimize bibdesk, multimarkdown and scrivener for a nice scientific bibliography and citation workflow, April

2014. URL <https://timbrandes.com/guide/2014/04/08/optimize-bibdesk-multimarkdown-and-scrivener-for-a-nice-scientific-bibliography-and->
- [7] *SCRIVENER 3 USER MANUAL for macOS*. Literature and Latte LTD., revision 3.0.1-01b edition, 12 2017. URL <https://www.literatureandlatte.com/learn-and-support/user-guides>.
- [8] Martin. Scrivener, bibdesk, latex, and symlinks. Ursecta, June 2014. URL <https://www.ursecta.com/2014/06/30/scrivener-bibdesk-latex-and-symlinks/>.
- [9] Jacqueline Vellguth. 15 gute Gründe für das ultimative Schreibwerkzeug – Scrivener 3.0. Schriftsteller-werden.de, April 2018. URL <https://www.schriftsteller-werden.de/schreibprogramm/15-gute-gruende-fuer-das-ultimative-schreibwerkzeug-scrivener-3-0/>.