

L^AT_EX 2_ε für Vereinsmeier: Minutes.sty

Knut Lickert

Mit dem hier beschriebenen Paket `Minutes.sty` können Protokolle für Vereine o. ä. erstellt werden. Zusammen mit der Klasse `scrartcl` werden einzelne Protokolle erstellt, mit `scrreprt`-artigen Klassen kann eine Zusammenfassung mehrerer Protokolle erstellt werden.

Das Paket ist im CTAN unter `tex_archive/macros/latex/contrib/supported/minutes/` verfügbar.

Einleitung

Wenn man wie ich in einem Verein aktiv tätig ist, wird man wohl das Problem kennen: Man hat Vereinssitzungen und will Protokolle über die Sitzungen haben. Erst muss ein Protokollant gefunden werden, und dann ist die Frage wie das Protokoll erstellt wird.

Dabei sammelt sich eine Vielzahl von einzelnen Protokollen in verschiedenen Dateiformaten (meist auf der Basis von winzigweich-Produkten, aber auch reine Handschriften) und unterschiedlichen Formatierungen an (mit oder ohne Inhaltsverzeichnis. . .). Diese Protokolle werden gesammelt und abgeheftet. Wenn man etwas sucht, müssen alle Protokolle durchgeblättert werden um das Gesuchte zu finden.

Dieses Chaos führte in einem Verein in dem ich Mitglied bin dazu, dass keine Protokolle mehr erstellt wurden und wenn, diese nicht auffindbar waren, und wenn man sie doch gefunden hat, dort die gesuchte Information nicht gefunden wurde. Darauf habe ich in den sauren Apfel gebissen und beschlossen, ab sofort die Protokollierung selbst zu übernehmen.

Meine Anforderungen an die Protokolle waren:

- Jedes Protokoll hat einen identischen Kopf mit Titel, Sitzungsleiter, Protokollant, Anwesenden und Fehlenden. Sitzungsleiter und Fehlende sind optional.
- Alle Themen der Sitzung werden in einem Inhaltsverzeichnis aufgelistet.

- Die einzelnen Protokolle können ohne Probleme in eine Protokollsammlung übernommen werden. Diese Sammlung hat ein Gesamtverzeichnis aller Themen und einen Index.
- Protokolle anderer, in einem beliebigen Dateiformat, können eingefügt werden. Dazu wird ein Eintrag in das Inhaltsverzeichnis eingefügt und leere Seiten zur Aufnahme des fremden Protokolls freigelassen.
- Thesenpapiere und andere Einlagen können eingefügt werden, entsprechende Leerseiten werden freigelassen. Es wird ein Verzeichnis aller Zusätze erstellt.
- Aufgaben sollten Leuten zugeordnet werden und in Vorlagen die Leute solange quälen, bis sie erledigt werden.
- *Keine* Anforderung war, eine vorbereitete Tagesordnung mit einer realen Sitzung abzugleichen.

Diese Anforderungen bewogen mich, ein Paket `Minutes.sty` zu schreiben. Eine erste Testversion hieß noch `protocol`. „Protocol“ existiert zwar auch im englischen, bezieht sich aber auf das diplomatische Protokoll, nicht auf Besprechungsprotokolle.

Alle Makros sind englisch definiert, für deutschsprachige Protokollanten sind deutschsprachige Makros verfügbar. Dieser Artikel verwendet die deutschsprachigen Makros.

Minutes.sty und Klassen

`Minutes.sty` ist als Paket implementiert und greift intern auf Gliederungskommandos wie `\section` zurück. Durch die Wahl einer Klasse kann so weitgehend das Aussehen der Protokolle bestimmt werden. `Minutes.sty` selbst hat zwei Möglichkeiten die das Aussehen beeinflussen. Mit den Optionen `ListTitle` und `TableTitle` kann das Aussehen der Protokollköpfe beeinflusst werden. Mit der Option `TwoColumn` wird der Textkörper zweispaltig gesetzt, Überschriften werden über beide Spalten gedruckt.

Das Paket `Minutes.sty` unterscheidet zwischen Einzelprotokollen und Protokollsammlungen. Wird das Paket zusammen mit `scrartcl` oder einer ähnlichen Klasse verwendet, wird ein einzelnes Protokoll ausgegeben. Zusammen mit `scrreprt` ist eine Sammlung mehrerer Protokolle möglich. Unterscheidungskriterium ist die Existenz des Makros `\chapter`. Will man den op-

```

\begin{Protokoll}{Beispiel eines Protokoll in ...}
\untertitel{Dieses Protokoll ist mit \minutes\ erzeugt}
\moderation{Knut Lickert}
\protokollant{Knut Lickert}
\teilnehmer{Leser von "'\DTK"'}
\sitzungsdatum{\today}
\sitzungsbeginn{20:00}
\sitzungsende{23:00}
\verteiler{alle Interessierten}
\fehlend[alle Analphabeten]{Vereinsm{"u"}ller}
\protokollKopf
\topic{Top eins}
\subtopic{zu Top eins}
\addtopic{Einschub in Tagesordnung}
\topic{Top zwei}
\end{Protokoll}

```

Abbildung 1: Ein Protokoll mit `Minutes.sty`

tischen Eindruck einer neuen Klasse mit den verschiedenen Optionen von `minutes.sty` testen, liegt dem Paket eine Beispieldatei `MinutesStyleGuide` vor, die einem eine schnelle Übersicht über alle Optionen bietet.

Eine Ausgabe mehrerer Protokolle mit `scartcl` ist prinzipiell möglich, das Ergebnis wird aber kaum den Wünschen entsprechen, denn jedes Protokoll erhält ein vollständiges Inhaltsverzeichnis aller Protokolle.

Die Protokoll-Umgebung

Ein Protokoll wird mit einer Umgebung `Protokoll` definiert. Am Beginn der Umgebung werden Protokollant, Teilnehmer, ... definiert. Ein Beispiel ist in Abbildung ?? zu sehen. Per Option kann zwischen einem Protokollkopf in Listen- (mit der Option `ListTitle`, siehe Abbildung ??) oder Tabellenform (mit der Option `TableTitle`, siehe Abbildung ??) gewählt werden.

1 Beispiel eines Protokollkopfs in Listform

Dieses Protokoll ist mit `Minutes.sty` erzeugt

Moderation Knut Lickert

Protokollant Knut Lickert

Anwesend Leser von „T_EXnische Komödie“

Fehlend(entschuldigt) alle Analphabeten

Fehlend(unentschuldigt) Vereinsmüller

Datum 24. Juli 2000 20:00 – 23:00

Verteiler alle Interessierten

Tagesordnung

1.1 Top eins	2
1.1.1 zu Top eins	2
(1.1) Einschub in Tagesordnung	2
1.2 Top zwei	2

2 Beispiel eines Protokollkopfs in Tabellenform

Dieses Protokoll ist mit <code>Minutes.sty</code> erzeugt	
Datum: 24. Juli 2000 20:00– 23:00	
Moderation: Knut Lickert	Protokollant: Knut Lickert
Anwesend: Leser von „TeXnische Komödie“	Verteiler: alle Interessierten
Fehlend(entschuldigt): alle Analphabeten	Fehlend(unentschuldigt): Vereinsmüller

Tagesordnung

2.1 Top eins	4
2.1.1 zu Top eins	4
(2.1) Einschub in Tagesordnung	4
2.2 Top zwei	4

Tagesordnungspunkte

Ein Tagesordnungspunkt wird mit `\topic{Titel}` eingeleitet. Untergliederungen werden mit `\subtopic` und `\subsubtopic` definiert. Als Überschrift ohne Hierarchiestufe kann `\minitopic` verwendet werden.

So definierte Tagesordnungspunkte werden dem Protokoll als Tagesordnung vorangestellt. `\topic`-Einträge werden zusätzlich in das Inhaltsverzeichnis der Protokollsammlung übernommen.

Wird eine Einladung zu einer Sitzung versendet, so soll die versendete Tagesordnung identisch sein mit der Tagesordnung aus dem Protokoll. Trotzdem kommt es vor, dass Zusatzpunkte auftauchen. `minutes.sty` unterstützt solche Wünsche mit dem Makro `\zusatztopic`. Diese Einträge erhalten eine eigenständige eingeklammerte römische Nummerierung.

Geheimes

Sollen Teile eines Protokolls nicht veröffentlicht werden, kann der entsprechenden Bereich zwischen mit `\begin{Secret}` und `\end{Secret}` geschachtelt werden. Diese geheimen Teile werden nur übernommen, falls die Optionen `Secret` gewählt wurde.

Kurze Abschnitte können mit `\secret` verborgen werden.

Diskussionspapiere

Gibt es zusätzliche Papiere, die einem Protokoll hinzugefügt werden sollen (Thesenpapiere, Verträge etc.) kann man das Makro `\attachment` verwenden. Das Makro hat drei Parameter:

- eine optionale Kurzbezeichnung, der zum Setzen eines Labels benutzt wird,
- ein Titel, der im Inhaltsverzeichnis erscheint,
- Seitenzahl des Anhangs.

```
\anhang[verweis]{Thesenpapier mit zwei Seiten}{2}
\anhang{Thesenpapier mit drei Seiten}{3}
```

ergibt im laufenden Text:

0.i	Thesenpapier mit zwei Seiten	8 [2]
0.ii	Thesenpapier mit drei Seiten	10 [3]

Abbildung 4: Anhangsliste

Anhang 0.i: Thesenpapier mit zwei Seiten Seite ??[2]

Anhang 0.ii: Thesenpapier mit drei Seiten [3]

Die Anhänge werden römisch nummeriert. Mit `\ref{verweis}` kann in diesem Beispiel auf das erste Thesenpapier verwiesen werden. („Anhang ?? auf Seite ??“).

Mit `\anhangsliste` wird ein Verzeichnis mit allen Anhängen erstellt, siehe Abbildung ???. Nach Ausgabe der Anhangsliste wird ein Seitenumbruch durchgeführt, die Anhänge müssen dann manuell eingelegt und nummeriert werden. In der Anhangsliste wird die Seitennummer und die Anzahl der Seiten angegeben. Das Verzeichnis erhöht die weitere Seitennummerierung gemäß der Seitenanzahl der Anhänge, für Die T_EXnische Komödie wird das unterdrückt.

Die Möglichkeit eines Anhangsverzeichnisses pro Protokoll ist für eine spätere Version geplant. Die Ausgabe ohne eine Änderung der Seitennummerierung soll per Option an- und ausschaltbar sein.

Aufgaben und Listen offener Aufgaben

Mit dem Makro `\aufgabe` werden Aufgaben definiert. Das Makro hat vier Parameter:

1. Erledigt-Datum (optional).
Ist dieser Parameter verwendet, wird das Datum in einer Fußnote vermerkt und die Aufgabe wird nicht mehr in der Liste der offenen Aufgaben aufgelistet. Erledigte Aufgaben müssen immer am Ort ihrer Definition als bearbeitet gesetzt werden.
2. Verantwortlicher. Ist niemand spezielles verantwortlich, kann `*` verwendet werden.
3. Datum zu dem es erledigt werden muss (optional)

Liste offener Aufgaben

Irgendwer	Korrektur lesen	0/8
Redaktion	↔ vor Drucktermin		
	Die T _E Xnische Komödie fertigstellen	0/8

Abbildung 5: Liste der offenen Aufgaben

4. Was erledigt werden muss.

```
\aufgabe[7. Juni]{Knut}[vor Redaktionsschluss]%
  {Artikel schreiben}
\aufgabe*{Korrektur lesen}
\aufgabe{Redaktion}[vor Drucktermin]{\DTK\ fertigstellen}
\aufgabe[wird gerade erledigt]*{\DTK\ lesen}
```

ergibt:

Aufgabe (vor Redaktionsschluss)¹: Artikel schreiben (Knut)

Aufgabe: Korrektur lesen

Aufgabe (vor Drucktermin): Die T_EXnische Komödie fertigstellen
(Redaktion)

Aufgabe²: Die T_EXnische Komödie lesen

Das Makro `\aufgabenliste` erzeugt eine Liste aller noch offenen Aufgaben (siehe Abbildung ??). Als optionales Argument kann der Dateiname einer anderen Protokollsammlung angegeben werden. So kann man beispielsweise offene Aufgaben aus dem Vorjahr (`\aufgabenliste[Protokolle1999]`) übernehmen. Die so ausgegebenen Seitenzahlen beziehen sich auf die Seiten aus der Originaldatei.

Termine und Vereinskalendar

Termine können mit dem Makro `\termin` definiert werden.

Aus dem Beispiel

¹7. Juni 2000

²wird gerade erledigt


```
\termin{2000/12/24}{Heilig Abend}
\termin{2000/12/24}[20:00]{Bescherung}[
  Gibt es \DTK\ auch in gebundener Lederfassung?
]
```

ergibt sich

24. Dezember 2000: Heilig Abend

24. Dezember 2000 20:00: Bescherung

Gibt es Die $\text{T}_{\text{E}}\text{X}$ nische Komödie auch in gebundener Lederfassung?

Die normierte Eingabe mit Jahr, Monat und Tag, jeweils getrennt mit „/“ wird in eine sprachabhängige Ausgabe umgesetzt. Per optionalen Parametern können den Terminen zusätzlich eine Uhrzeit und eine längere Beschreibung übergeben werden.

Aus diesen Terminen kann direkt eine Eingabedatei für das Paket `Calendar.sty` erzeugt werden. Das Paket `Calendar.sty` ermöglicht die Erstellung von Kalendern in verschiedenen Versionen und ist im CTAN unter `macros/latex/contrib/supported/calendar` zu finden. So ist es unter anderem möglich monatliche Kalenderblätter oder eine Terminliste zu erzeugen. Bei der Erzeugung eines Kalenders reagiert `\termin` sehr empfindlich auf zerbrechliche Befehle. Der Langtext wird bislang nicht weitergegeben. Soll keine Eintragung in einen Kalender stattfinden, ist `\termin*` vorgesehen.

Ausblick und zukünftige Versionen

Es liegt schon ein Makro `\beschluss` für Beschlüsse vor. Eine rudimentäre Beschlussliste ist schon machbar, muss aber noch von mir überarbeitet werden. Mein Wunsch ist eine Gruppierung von Beschlüssen nach Themen. Vermutlich ist dies nur mit einer Nachbearbeitung mit `makeindex` oder ähnlichem möglich.

Aufgaben sollen analog wie Termine direkt in einen Kalender eingefügt werden können. Das Makro `\aufgabe` soll so erhalten bleiben wie es jetzt ist, genau datierbare Aufgaben sollen einem neuen Makro vorbehalten bleiben.

Bekannte Fehler sollen selbstverständlich beseitigt werden. Dies sind momentan die fehlende Möglichkeit von Langtexten in Terminen in Kombination mit Kalendern und Instabilität der `Secret`-Umgebung zusammen mit der mehrspaltigen Ausgabe (Option `TwoColumn`).

Ich würde mich freuen, wenn das Paket auch von anderen „Vereinsmeiern“ genutzt wird. Über Rückmeldungen wo das Paket genutzt wird, freue ich mich besonders. Kommentare zu Sinn und Unsinn der Makros, sowie weitere Verbesserungsvorschläge nehme ich gerne entgegen.