

# KOMA-Script Datei `scrjura.dtx` \*

Markus Kohm<sup>†</sup>      Dr. Alexander Willand<sup>‡</sup>

2014/11/22

Das `scrjura`-Paket ist für die Kautelarjurisprudenz gedacht. Es soll für Anwälte und Notare eine flexible Hilfe bei der Abfassung von Verträgen, Satzungen und juristischen Kommentaren bieten. Es ist in Zusammenarbeit mit Dr. Alexander Willand entstanden, bzw. noch immer in Entstehung begriffen.

## Inhaltsverzeichnis

<b>1</b>	<b>Rechtlicher Hinweis</b>	<b>2</b>
<b>2</b>	<b>Das Paket »scrjura«</b>	<b>2</b>
2.1	Überblick . . . . .	2
2.2	Die Optionen beim Laden des Pakets . . . . .	3
2.3	Optionen betreffend Kolumnentitel . . . . .	5
2.4	Der Paragraph . . . . .	5
2.5	Verweise . . . . .	7
2.6	Absatz- und Satznummerierung . . . . .	9
2.7	kile anpassen, <code>scrjura</code> unter <code>lyx</code> . . . . .	9
2.8	Zusätzliche Vertragsumgebungen . . . . .	10
2.9	Sonstiges . . . . .	12
<b>3</b>	<b>Bekannte Probleme</b>	<b>12</b>
<b>4</b>	<b>Implementierung</b>	<b>12</b>

---

\*Diese Datei ist Version v0.9 von `scrjura.dtx`.

<sup>†</sup>Paketautor, [mailto:komascript\(at\)gmx.info](mailto:komascript(at)gmx.info)

<sup>‡</sup>Autor der Anleitung, [mailto:alexander.willand\(klammeraffe\)t-online.de](mailto:alexander.willand(klammeraffe)t-online.de)

# 1 Rechtlicher Hinweis

Dieser Teil von KOMA-Script darf ohne die restlichen Teile von KOMA-Script verteilt werden. Dazu dürfen die Dateien »scrlogo.dtx« und »scrjura.dtx« unter den Bedingungen der L<sup>A</sup>T<sub>E</sub>X Project Public License, wahlweise Version 1.3c oder (nach Ihrer Entscheidung) jede spätere Version, verteilt und verändert werden. Die aktuelle Version dieser Lizenz ist unter <http://www.latex-project.org/lppl.txt> zu finden und Version 1.3c oder neuer ist Teil von L<sup>A</sup>T<sub>E</sub>X Version 2005/12/01 oder neuer. Es ist nicht gestattet die Datei »scrjura.dtx« ohne die Datei »scrlogo.dtx« zu verteilen. Eine Verteilung ist nur zusammen gestattet.

Zu KOMA-Script existiert eine ausführliche Anleitung in Deutsch und Englisch. Die Anleitung zum Paket scrjura existiert derzeit jedoch nur in dieser deutschen Kurzform.

Das komplette KOMA-Script-Paket ist unter CTAN:/macros/latex/contrib/koma-script/ zu finden, wobei »CTAN:« ein abkürzendes Synonym für das Verzeichnis »tex-archive« jedes CTAN-Servers oder -Spiegels, beispielsweise <ftp://ftp.dante.de/tex-archive/> darstellt. Siehe <http://www.ctan.org> für eine Liste aller CTAN-Server und -Spiegel.

## 2 Das Paket »scrjura«

### 2.1 Überblick

Das Paket scrjura ist in erster Linie für den Satz von Verträgen gedacht. Über eine Reihe von Optionen kann die Wirkung des Pakets verändert werden. Innerhalb von Verträgen steht eine Reihe von Befehlen zur Verfügung.

Wer juristische Dokumente verfasst, steht oft unter hohem Zeitdruck. Überraschende Fehlermeldungen von L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> kann man da nicht brauchen. Erfreulicherweise setzt scrjura lediglich das scrkbase-Paket voraus.

scrjura bietet zwei Modi:

1. den „Vertragsmodus“: das gesamte Dokument wird als Vertrag gesetzt;
2. den „Umgebungsmodus“: Nur der Text innerhalb der contract-Umgebung wird als Vertrag gesetzt.

Der *Vertragsmodus* wird durch die Option `contract` beim Laden des Pakets dauerhaft eingeschaltet. Also: Mit `\usepackage[contract]{scrjura}` wird Text nach `\begin{document}` als Vertrag gesetzt. Der *Umgebungsmodus* wird dagegen wie jede andere Umgebung mit `\begin{contract}` begonnen und ist entsprechend mit `\end{contract}` zu beenden. Beide Modi können nicht im gleichen Dokument verwendet werden.

Ein Inhaltsverzeichnis wird wie üblich mit `\tableofcontents` erzeugt und enthält – neben etwaigen anderen Überschriften – auch alle Paragraphentitel, sofern `scrjura` mit der Option `juratotoc` geladen wurde. Automatische Kolumnentitel sind ebenfalls möglich.

Jeder Paragraph hat einen Titel, z. B.: `\Clause{title=Besitz}`. Die Nummerierung der Paragraphen erfolgt automatisch, kann aber auch manuell vorgenommen werden.

Nahezu alle vom Anwender anzugebenden Daten folgen dem *key=value*-Schema, wie bereits an dem eben genannten Titel eines Paragraphen zu sehen war.

Enthält ein Paragraph mehr als einen Absatz, werden alle Absätze automatisch nummeriert. Die Nummerierung kann fallweise abgeschaltet werden. Es ist darüber hinaus möglich, die Sätze eines Absatzes nummerieren zu lassen.

Paragraphen, Absätze und Sätze können referenziert werden. Bei Absätzen besteht die Wahl, wie die Ausgabe erfolgt; es ist sowohl möglich, beispielsweise „§ 16 Absatz 3 Satz 1“ als auch „§ 16 III 1“ als Ausgabe zu erhalten.

`scrjura` unterstützt das `hyperref`-Paket, ist damit aber hier noch nicht ausführlich getestet worden. Wie immer, wird `hyperref` zuletzt geladen. Andernfalls gibt es faszinierende Fehlermeldungen. Wird `hyperref` wieder entfernt, muss die `*.aux`-Datei gelöscht werden.

## 2.2 Die Optionen beim Laden des Pakets

<code>contract</code>	Die Option <code>contract</code> sorgt dafür, dass das gesamte Dokument als Vertrag interpretiert wird. Innerhalb des Dokuments ist dann keine weitere Vertragsumgebung zu verwenden. Dasselbe Verhalten kann erreicht werden, indem unmittelbar nach <code>\begin{document}</code> eine <code>contract</code> -Umgebung begonnen und unmittelbar vor <code>\end{document}</code> wieder beendet wird.
<code>juratitlepagebreak</code>	Normalerweise sind Seitenumbrüche innerhalb der Überschriften von Paragraphen nicht erlaubt. Einige Juristen verwenden aber so lange Überschriften, dass selbst Seitenumbrüche darin sinnvoll sein können. Mit der

Option `juratitlepagebreak` wird dies ermöglicht. Diese Option kann jederzeit mit Hilfe von `\KOMAOptions{juratitlepagebreak}` ein- oder mit `\KOMAOptions{juratitlepagebreak=false}` wieder ausgeschaltet werden.

`juratotoc`  
`juratoclevel` Die Option `juratotoc` sorgt dafür, dass die Paragraphen in das Inhaltsverzeichnis eingetragen werden. Dabei wird in der Voreinstellung die Gliederungstiefe 2 für diese Einträge verwendet. Über den Zähler `juratoclevel` kann auch eine andere Gliederungstiefe vorgegeben werden. Das Paket greift hier auf die Routinen von  $\text{\LaTeX}$  zur Erstellung eines Inhaltsverzeichnisses zurück. Bei dreistelligen Paragraphennummern kann im Inhaltsverzeichnis die Nummer in den Überschriftstext hineinragen.

Die einfachste Abhilfe gegen diesen Fehler besteht darin, die Box mit der Gliederungsnummer zu verbreitern, beispielsweise mit

```
\renewcommand{\numberline}[1]{\makebox[6em][l]{#1}}
```

die Breite der Box den jeweiligen Gegebenheiten anzupassen.

Eine andere Abhilfe bei der Verwendung einer KOMA-Script-Klasse besteht darin, die Option `tocleft` zu verwenden. Zuletzt kann man selbstverständlich auf die verschiedenen Pakete zurückgreifen, mit denen sich das Inhaltsverzeichnis anpassen lässt. Den Gewinn an Möglichkeiten erkauft man sich allerdings mit einem erhöhten Risiko, dass Fehler auftreten.

Ein für den Autor der Anleitung entscheidender Vorteil von `scrjura` liegt in dem umfassenden System, auf andere Klauseln des Vertrages verweisen zu können. Beim Laden des Paktes kann per Option entschieden werden, wie die Verweise formatiert werden sollen: lang (Bsp: »§ 314 Absatz 2 Satz 2«), kurz (»§ 314 Abs. 2 S. 2«) und numerisch (»§ 314 II 2«). Die Optionen hierzu lauten:

- `ref=numeric`
- `ref=long`
- `ref=short`

Wird nichts angegeben, ist `ref=long` voreingestellt.

Wer hier spezifische Wünsche hat, kann die Zitierweise von Absätzen und Sätzen bereits bei den Optionen hiervon abweichend festlegen. Es sind beliebige Kombinationen von `parlong`, `parshort`, `parnumeric`, `nopar` mit `sentenceslong`, `sentenceshort`, `sentencesnumeric`, `nosentence` möglich. Beispiel: `\usepackage[ref=parlong,ref=sentenceshort]{scrjura}`. Wir haben allerdings diese Kombinationen nicht alle getestet.

## 2.3 Optionen betreffend Kolummentitel

Bei langen Verträgen können automatische Kolummentitel hilfreich sein. Allerdings hängt die Ausgabe hier davon ab, wie die diese erzeugt werden. `paragraphmark` unterstützt lebende Kolummentitel. Die Option `paragraphmark` kann auf folgende Werte gesetzt werden:

- `false`, `off`, `no`
- `right`
- `both`

Sollte der Vertrag in Abschnitte unterteilt werden, also etwa mit einer Zeile `\parnumberfalse\section{ABSCHNITT}\parnumbertrue`, würde auf den ungeraden Seiten die Abschnittsüberschriften und auf den geraden Seiten der Titel des aktuellen Paragraphen angegeben werden.

Im übrigen können die zur Gestaltung von Kopfzeilen üblichen Pakete (z.B. `scrpage2`) verwendet werden.

## 2.4 Der Paragraph

`scrjura` führt zwei Gliederungsebenen ein: Paragraph und Subparagraph. Selbstverständlich funktioniert deren im folgenden beschriebene Befehle im *Umgebungsmodus* nur innerhalb der Vertragsumgebung. Es sei an dieser Stelle darauf hingewiesen, dass bei früheren Versionen von `scrjura` Befehle und Zähler noch den Terminus „Paragraph“ im Namen trugen. Da dieser Terminus für juristische Paragraphen im Englischen jedoch eher unüblich ist und vor allem auch zur Vermeidung von Verwechslungen mit dem normalen Gliederungsbefehl `\paragraph`, wurde dies ab Version 0.7b in „Clause“ geändert. Ältere Dokumente müssen daher ggf. entsprechend angepasst werden.

`\Clause` Für die Paragraphenüberschrift stehen verschiedene optionale Argumente zur Verfügung, die ebenfalls in einer durch Kommata zu trennenden `key=value`-Liste angegeben werden:

Eintrag	Erklärung	Eintrag unterdrücken
<code>title = &lt;Titel&gt;</code>	Titel des Paragraphen	<code>dummy</code>
<code>head = ...</code>	Kolummentitel des Paragraphen	<code>nohead</code>
<code>entry = ...</code>	Eintrag im Inhaltsverzeichnis	<code>noentry</code>
<code>number= ...</code>	die Paragraphennummer	-

Beispiele: `\Clause{title=Gattungsschuld, number=244}` gibt „§ 244 Gattungsschuld“ aus. `\Clause{title=Gattungsschuld}` als erster Paragraph gibt „§ 1 Gattungsschuld“ aus, während `\Clause{}` an gleicher Stelle zu „§ 1“ führt, also einer Paragraphenzählung ohne Titel.

Will man aber beginnend von beispielsweise § 244 an eine Reihe von Paragraphen setzen, empfiehlt es sich nicht, die Option `number=` zu verwenden, denn die Option lässt den Zähler `Clause` unberührt. Will man nicht alle Paragraphen per Hand nummerieren, kann mit `\setcounter{Clause}{243}` der Startpunkt für eine automatische Zählung gesetzt werden. Bei `\SubClause` ist der Zähler natürlich mit einer arabischen Zahl anzugeben, auch wenn die Ausgabe dann in Kleinbuchstaben erfolgt.

`\SubClause` Sofern ein Paragraph eingeschoben werden soll, kann dieses Makro verwendet werden, es verhält sich im übrigen wie `\Clause`. Die Nummerierung fügt der zuletzt verwendeten Paragraphennummer einen Kleinbuchstaben hinzu.

Die Nummerierung für `\SubClause` erfolgt in `scrjura.sty` über folgende Definition:

```
\renewcommand*{\thecontractSubClause}{%
{\theClause\alph{contractSubClause}}}
```

Wer zwischen Paragraphenzahl und dem Kleinbuchstaben einen Abstand wünscht, sowie nach dem Kleinbuchstaben eine runde Klammer, kann in die Präambel also einfügen:

```
\renewcommand*{\thecontractSubClause}{%
{\theClause~\alph{contractSubClause}}).
```

`preskip` Abstände vor und nach der Paragraphen-Überschrift können derzeit über  
`postskip` `\setkeys{contract}{preskip=... , postskip=...}` gesetzt werden.

Es ist möglich, einen Vertrag in Abschnitte zu unterteilen unter Verwendung z. B. von `\section{}`. Allerdings muss entweder vorher die `contract`-Umgebung beendet und danach fortgesetzt werden (die Paragraphenzählung wird dadurch nicht zurückgesetzt), oder man setzt davor ein `\parnumberfalse` und danach ein `\parnumbertrue`. Vergisst man dies, wird der Abschnitt zum ersten als Paragraphenabsatz und dann als Abschnitt nummeriert.

`\ellipsispar` Manchmal wird insbesondere für Kommentare gewünscht, dass man Absätze in Paragraphen auch auslassen kann, diese Absätze aber dennoch mitgezählt und auch bei der automatischen Absatznummerierung mitberücksichtigt werden. Die ausgelassenen Absätze sollen dabei in der Regel durch Pünktchen markiert werden. Genau dies erledigt `\ellipsispar`. Als  
`\parellipsis`

optionales Argument kann dabei die Anzahl der Absätze angegeben werden, die ausgelassen wurde. Voreingestellt ist ein Absatz. Für das Auslassungszeichen wird `\parellipsis` verwendet. Dies ist mit `\textellipsis` vordefiniert, falls eine solche Anweisung existiert. Anderenfalls wird `\dots` verwendet.

## 2.5 Verweise

Verlässlich funktionierende Verweise sind eine große Erleichterung beim Abfassen von Verträgen. Wenn in einem ersten Entwurf eines Gesellschaftsvertrags die Abfindungsregelungen in §§ 28 ff. enthalten sind, diese aber nach einigen Umbauten sich irgendwann in §§ 19 ff. wiederfinden, dann müssen alle Verweise auf die unterschiedlichen Abfindungsregelungen für kündigende Gesellschafter, für solche, die in Rente gehen und für 'rausgeworfene diese Änderungen nachvollzogen haben.

In der Grundeinstellung führt der Verweis `\ref{testA}` auf ein `\label{testA}`, das in einem Absatz platziert wurde, zur Ausgabe eines Vollverweises einschließlich der Satzangabe.

Nun will man häufig nicht den Paragraph, Absatz *und* Satz, sondern nur Paragraph und Absatznummer referenzieren. Die folgende Tabelle enthält eine systematische Darstellung aller Verweismöglichkeiten. Ein »L« kennzeichnet die Langform, ein »S« die Kurzform und ein »N« die numerische Darstellung des Verweises.

Zunächst das Beispiel, auf das sich die Tabelle bezieht:

```
\begin{contract}
\Clause{title=Beispiel}
Es irrt der Mensch.\label{testA}
\end{contract}
```

Dies ergibt einen sehr kurzen Paragraphen:

---

### § 1 Beispiel

Es irrt der Mensch.

---

Folgende Verweise auf diesen § 1 Absatz 1 Satz 1 sind möglich:

Code	Ergebnis
<code>\ref{testA}</code>	§ 1 Absatz 1 Satz 1
<code>\refL{testA}</code>	§ 1 Absatz 1 Satz 1
<code>\refS{testA}</code>	§ 1 Abs. 1 S. 1
<code>\refN{testA}</code>	§ 1 I 1.
<code>\refClause{testA}</code>	§ 1
<code>\refClauseN{testA}</code>	1
<code>\refPar{testA}</code>	Absatz 1
<code>\refParL{testA}</code>	Absatz 1
<code>\refParS{testA}</code>	Abs. 1
<code>\refParN{testA}</code>	I
<code>\refParN[arabic]{testA}</code>	1
<code>\refSentence{testA}</code>	Satz 1
<code>\refSentenceL{testA}</code>	Satz 1
<code>\refSentenceS{testA}</code>	S. 1
<code>\refSentenceN{testA}</code>	1.

Dabei hängt die Ausgabe von `\ref`, `\refPar` und `\refSentence` von den Einstellungen der Option `ref` ab. Hier zum Vergleich mit der Einstellung `\KOMAOPTIONS{ref=nopar}`:

Code	Ergebnis
<code>\ref{testA}</code>	§ 1 Satz 1
<code>\refPar{testA}</code>	
<code>\refSentence{testA}</code>	Satz 1

Das gleiche mit `\KOMAOPTIONS{ref=nosentence}`:

Code	Ergebnis
<code>\ref{testA}</code>	§ 1 Absatz 1
<code>\refPar{testA}</code>	Absatz 1
<code>\refSentence{testA}</code>	

Das gleiche mit `\KOMAOPTION{ref}{nopar,nosentence}`:

Code	Ergebnis
<code>\ref{testA}</code>	§ 1
<code>\refPar{testA}</code>	
<code>\refSentence{testA}</code>	

Zu jeder existiert noch eine Stern-Version, also z.B. `\ref*{testA}`. Wenn das Paket `hyperref` geladen ist, erzeugt die besternte Version einen Verweis, der *kein* Hyperlink ist. Wenn das Paket jedoch nicht geladen wurde, erzeugt die besternte Version – Fehler.

## 2.6 Absatz- und Satznummerierung

`\parnumbertrue` Absätze in Paragraphen werden automatisch nummeriert. Mit `\parnumberfalse`  
`\parnumberfalse` kann die Nummerierung ab- und mit `\parnumbertrue` wieder eingeschaltet werden.

`\Sentence` Auch alle einzelnen Sätze eines Absatzes können durch eine hochgestellte Ziffer nummeriert werden. Allerdings kann dies nicht automatisch geschehen. Am einfachsten gestaltet sich die Satznummerierung unter Zuhilfenahme des Befehls `\useshortands` aus dem ohnehin bei deutschen Texten notwendigen `babel`-Paket:

```
\usepackage[...]{babel}
\useshortands{'}
\defineshortand{'S}{\Sentence\ignorespaces}
\defineshortand{'.}{. \Sentence\ignorespaces}
```

Die Sätze lassen sich mit einem vorangestellten `'S` nummerieren. Alternativ kann der Satzendeapostroph gesetzt werden: `'.`, damit wird dann der nächste Satz nummeriert.

## 2.7 kile anpassen, scrjura unter lyx

Bei langen Verträgen ist es sehr hilfreich, die Gliederung und die Paragraphentitel auf dem Bildschirm einblenden zu können. Wer `kile` als Editor verwendet, der kann sich – Open Source sei Dank! – selbst helfen. Bei Version 2.0.0 von `kile` genügt es, in der Datei `kiledocumentinfo.cpp` an geeigneter Stelle zwei Zeilen (jeweils ohne Zeilenumbruch) einzufügen:

```
m_dictStructLevel["\\Clause"]=
KileStructData(5, KileStruct::Sect, "Clause");
```

```
m_dictStructLevel["\\SubClause"]=
KileStructData(6, KileStruct::Sect, "SubClause");
```

Zahlen über 7 sind eine schlechte Idee. Nach dem Speichern der Datei folgt der übliche Dreisprung: `configure`, `make`, `make install` (oder `checkinstall ...`). `kile` zeigt danach Paragraphen und »Unterparagraphen« als Dokumentenstruktur an. Vielleicht findet ein Anwender ja heraus, wie dies mit `TeXnicCenter` zu bewerkstelligen ist.

Bis jemand für die Verwendung von `scrjura` eine `lyx`-Klasse schreibt, wozu wir ausdrücklich ermutigen wollen, kann man sich bei der Nutzung von `lyx` damit behelfen, im sog. »`LATEX`-Vorspann« beispielsweise die Definition des Unterunterabschnitts anzupassen:

```
\renewcommand{\subsubsection}[1]{\Clause{title= #1}}
```

Damit lassen sich Paragraphenüberschriften in `lyx` als Unterunterabschnittsüberschriften setzen. Da die Absätze von Verträgen ohnehin von `scrjura` automatisch nummeriert werden, zeigen sich die Nachteile dieses Vorgehens erst, wenn man einen Unterunterabschnitt benötigt.

## 2.8 Zusätzliche Vertragsumgebungen

Einer der Anwender setzt mit `scrjura` keine Verträge oder Kommentare zu einzelnen Gesetzen, sondern Werke, in denen unterschiedliche Arten von Gesetzen behandelt werden. Es ist daher erforderlich, dass ein Paragraph nicht immer mit demselben Präfix »§« versehen wird, sondern beispielsweise als »Art.« oder »GG« oder was auch immer bezeichnet wird. Darüber hinaus wird eine unabhängige Zählung der unterschiedlichen Paragraphen benötigt.

`\DeclareNewJuraEnvironment` Die für unabhängige Vertrags- oder Gesetzestexte notwendigen Umgebungen werden über die Anweisung:

```
\DeclareNewJuraEnvironment{<Name>}[<Optionen>]
    {<Start-Anweisungen>}
    {<End-Anweisungen>}
```

deklariert. Das Argument `<Name>` ist dabei der Name der neuen Umgebung. `<Start-Anweisungen>` sind Anweisungen, die immer am Anfang der Umgebung ausgeführt werden, ganz als ob man sie jedes Mal unmittelbar hinter `\begin{<Name>}` einfügen würde. Entsprechend werden `<End-Anweisungen>` immer am Ende der Umgebung ausgeführt, ganz als

ob man sie jedes Mal unmittelbar vor `\end{<Name>}` einfügen würde. Ohne `<Optionen>` entspricht die neue Umgebung `<Name>` einer `contract`-Umgebung mit eigenen Zählern. Es besteht jedoch die Möglichkeit über eine mit Komma separierte Liste an `[<Optionen>]` darauf Einfluss zu nehmen. Derzeit werden folgende `[<Optionen>]` unterstützt:

**Clause=**`<Anweisung>`: Legt fest, auf welche `<Anweisung>` innerhalb der Umgebung die Anweisung `\Clause` abgebildet werden soll. Die `<Anweisung>` sollte wie die für `contract` dokumentierte Anweisung genau ein Argument erwarten. Für eine korrekte Anwendung dieser Option sind erweiterte Kenntnisse über die interne Funktion von `scrjura` notwendig. Außerdem können sich die Anforderungen an die `<Anweisung>` von Version zu Version noch ändern. Daher wird derzeit empfohlen, die Option nicht zu verwenden!

**SubClause=**`<Anweisung>`: Siehe Option `Clause`, allerdings für `\SubClause` statt `\Clause`.

**Sentence=**`<Anweisung>`: Legt fest, auf welche `<Anweisung>` innerhalb der Umgebung die Anweisung `\Sentence` abgebildet werden soll. Die `<Anweisung>` sollte kein Argument besitzen. Normalerweise sollte sie den Zähler `sentence` mit `\refstepcounter` erhöhen und dann in geeigneter Form ausgeben. Dabei ist besonders darauf zu achten, dass keine unerwünschten Leerzeichen eingebaut werden!

**ClauseNumberFormat=**`<Anweisung>`: Legt fest, auf welche Weise die Paragraphen-Nummern dieser Umgebung bei der Ausgabe formatiert werden. Es wird eine Anweisung mit genau einem Argument erwartet, der Nummer des Paragraphen. Falls diese Nummer das letzte Argument einer Kette von Anweisungen ist, so kann diese Kette an Anweisungen auch direkt angegeben werden.

Um beispielsweise die in der Einleitung zu diesem Abschnitt erwähnte Umgebung für Artikel zu definieren, genügt:

```
\DeclareNewJuraEnvironment{Artikel}[ClauseNumberFormat=Art.]{}
```

Sollen die Artikel unter Verwendung einer KOMA-Script-Klasse mit Absatzabstand statt Absatzeinzug gesetzt werden, kann

```
\DeclareNewJuraEnvironment{Artikel}[ClauseNumberFormat=Art.~]
{\KOMAOptions{parskip}}{}
```

verwendet werden. Natürlich wird dann auch bei der Referenzierung automatisch »Art.« an Stelle von »§« vorangestellt.

## 2.9 Sonstiges

<code>\parname</code>	Sprachabhängiger Name von Absatz und Satz.
<code>\sentencename</code>	Ausgabe der automatischen Zähler für Paragraph und eingeschobener Paragraph bei Verträgen.
<code>\thecontractClause</code>	
<code>\thecontractSubClause</code>	
<code>\AutoPar</code>	Automatische und manuelle Absatznummerierung.
<code>\ManualPar</code>	
<code>\Par</code>	Absatznummern ein- und ausschalten.
<code>\ifparnumber</code>	Zähler für die Absatznummern und dessen Ausgabe und Formatierung und
<code>\parnumbertrue</code>	Formatierung der Referenz. ACHTUNG: Wenn <code>\thepar</code> nicht nur eine arabische Zahl ist, werden Absätze bei automatischer Nummerierung unabhängig davon nummeriert, ob es mehr als einen Absatz im Paragraphen gibt.
<code>\parnumberfalse</code>	
<code>par</code>	
<code>\thepar</code>	
<code>\parformat</code>	So etwas wie <code>\ref</code> nur für den Paragraphen.
<code>\parciteformat</code>	
<code>\refClause</code>	So etwas wie <code>\ref</code> nur für den Absatz.
<code>\refPar</code>	So etwas wie <code>\ref</code> nur für den Satz.
<code>\refSentence</code>	
<code>sentence</code>	Zähler für die Satznummer und dessen Ausgabe, sowie Formatierung der Referenz. ACHTUNG: <code>\thesentence</code> muss bei unverändertem <code>\sentenceciteformat</code> eine arabische Zahl sein!
<code>\thesentence</code>	
<code>\sentenceciteformat</code>	
<code>\newmaxpar</code>	Werden intern verwendet und müssen deshalb allenfalls für Paketautoren dokumentiert werden.
<code>\getmaxpar</code>	

## 3 Bekannte Probleme

Die Verweistechnik wurde noch nicht ausführlich getestet; Fehler sind möglich.

## 4 Implementierung

```
1 \@ifpackageloaded{hyperref}{%
2   \PackageError{scrjura}{Package hyperref already loaded}{%
3     If you want to use package scrjura with package hyperref, you have to
4     use\MessageBreak
5     package scrjura before package hyperref.\MessageBreak
6     To solve the problem, you just should move the loading of package
7     hyperref\MessageBreak
8     behind the loading of package scrjura.}%
9 }
```

Es wird das Paket scrkbase benötigt, weil alles weitere darauf aufbaut.

```
10 \RequirePackage{scrkbase}[2013/03/26]
```

Außerdem wird tocbasic benötigt, weil einige wenige Befehle davon verwendet werden.

```
11 \RequirePackage{tocabasic}
```

`contract` Mit der Option `contract` wird das gesamte Dokument zu einem Vertrag. Man darf dann keine der von diesem Paket bereit gestellten Umgebungen mehr innerhalb des Dokuments verwenden! Realisiert wird dies, indem ganz am Ende von `\begin{document}` noch `\contract` ausgeführt wird.

```
12 \DeclareOption{contract}{%
13   \g@addto@macro\document\contract%
14 }
```

`juratotoc` Mit dieser Option wird der Zähler

`juratoclevel` auf einen gewünschten Wert oder 2 oder 10000 gesetzt.

`\if@juratotoc` Mit dem Zähler wiederum wird erreicht, dass Paragraphen in das Inhaltsverzeichnis aufgenommen werden.

`\toclevel@cpa`

```
15 \newcounter{juratoclevel}\setcounter{juratoclevel}{\@M}
16 \KOMA@key{juratotoc}[true]{%
17   \begingroup
18     \KOMA@set@ifkey{juratoclevel}{@tempswa}{#1}%
19     \ifx\FamilyKeyState\FamilyKeyStateProcessed
20       \if@tempswa
21         \setcounter{juratoclevel}{2}%
22       \else
23         \setcounter{juratoclevel}{\@M}%
24       \fi
25     \else
26       \setcounter{juratoclevel}{#1}%
27     \FamilyKeyStateProcessed
28   \fi
29 \endgroup
30 }
31 \providecommand*\toclevel@cpa{\arabic{juratoclevel}}
```

juratocnumberwidth Der Einzug und die Breite der Nummer für Inhaltsverzeichniseinträge kann  
 juratocindent über diese beiden Optionen bzw. Makros bestimmt werden.

```
\cpar@numberwidth
\cpar@indent
32 \newcommand*{\cpar@numberwidth}{2em}
33 \newcommand*{\cpar@indent}{1.5em}
34 \KOMA@key{juratocnumberwidth}{%
35 \begingroup\setlength{\@tempdima}{#1}\endgroup
36 \renewcommand*{\cpar@numberwidth}{#1}%
37 \FamilyKeyStateProcessed
38 }
39 \KOMA@key{juratocindent}{%
40 \begingroup\setlength{\@tempdima}{#1}\endgroup
41 \renewcommand*{\cpar@indent}{#1}%
42 \FamilyKeyStateProcessed
43 }
```

juratitlepagebreak Mit dieser Option wird der Schalter `\if@juratitlepagebreak` auf `\iftrue` gesetzt.

`\if@juratitlepagebreak` Mit diesem Schalter wiederum wird erreicht, dass in den Überschriften von Paragraphen ein Seitenumbruch erlaubt wird.

```
44 \KOMA@ifkey{juratitlepagebreak}{@juratitlepagebreak}
```

parnumber Die (automatische) Absatznummerierung lässt sich nun auch per Option ein- und ausschalten.

```
45 \newif\ifparnumber
46 \KOMA@key{parnumber}[true]{%
47 \ifstr{#1}{auto}{\AutoPar\FamilyKeyStateProcessed}{%
48 \ifstr{#1}{manual}{\ManualPar\FamilyKeyStateProcessed}{%
49 \KOMA@set@ifkey{parnumber}{parnumber}{#1}%
50 }%
51 }%
52 }
```

paragraphmark Mit diesen Optionen wird `\Clausemark` so umdefiniert, dass es entweder ein  
 markright `\markright` oder ein `\markboth` ausführt. Im Gegensatz zu `\chaptermark`  
 markboth etc. erwartet `\Clausemark` neben dem Titel auch noch die zu verwendende  
 \Clausemark Nummer (das könnte bei Bedarf noch geändert werden!)

```
53 \newcommand*{\Clausemark}[1]{%
54 \KOMA@key{paragraphmark}{%
55 \begingroup
56 \KOMA@set@ncmdkey{paragraphmark}{@tempa}{%
57 {false}{0},{off}{0},{no}{0},%
58 {forceright}{1},%
```

```

59     {forceboth}{2},%
60     {right}{3},%
61     {both}{4}%
62   }{#1}%
63 \ifx\FamilyKeyState\FamilyKeyStateProcessed
64   \ifcase\number\@tempa
65     \endgroup
66     \let\Clausemark@gobble
67   \or
68     \endgroup
69     \renewcommand*\Clausemark}[1]{%
70       \markright{\csname MakeMarkcase\endcsname{##1}}}%
71   \or
72     \endgroup
73     \renewcommand*\Clausemark}[1]{%
74       \markboth{\csname MakeMarkcase\endcsname{##1}}%
75                {\csname MakeMarkcase\endcsname{##1}}}%
76   \or
77     \endgroup
78     \renewcommand*\Clausemark}[1]{%
79       \ifx
80         \@mkboth@gobbletwo
81       \else
82         \markright{\csname MakeMarkcase\endcsname{##1}}%
83       \fi}%
84   \or
85     \endgroup
86     \renewcommand*\Clausemark}[1]{%
87       \@mkboth{\csname MakeMarkcase\endcsname{##1}}%
88                {\csname MakeMarkcase\endcsname{##1}}}%
89   \else
90     \endgroup
91   \fi
92 \else
93   \endgroup
94 \fi
95 }
96 \KOMA@DeclareDeprecatedOption[scrjura]{markright}{paragraphmark=forceright}
97 \KOMA@DeclareDeprecatedOption[scrjura]{markboth}{paragraphmark=forceboth}

```

ref Die Form, in der Absätze und Sätze referenziert werden. Es gibt eine lange,  
 parcite**name** eine abgekürzte und eine numerische Form.  
 sentence**name**

\parcite@**format** In der Voreinstellung wird das lange Format verwendet. Es gilt: 0 = lang,  
 \sentencecite@**format** 1 = kurz, 2 = numerisch, -1 = gar nicht

```

98 \newcommand*\parcite@format}{0}
99 \newcommand*\sentencecite@format}{0}

```

Per Option kann die Voreinstellung verändert werden.

```
100 \KOMA@key{ref}-{%
101  \begingroup
102    \KOMA@set@ncmdkey{ref}{@tempa}{%
103      {parlong}{1},{longpar}{1},{ParL}{1},%
104      {parshort}{2},{shortpar}{2},{ParS}{2},%
105      {parnumeric}{3},{numericpar}{3},{ParN}{3},%
106      {paroff}{4},{nopar}{4},%
107      {sentencelong}{10},{longsentence}{10},{SentenceL}{10},%
108      {sentenceshort}{20},{shortsentence}{20},{SentenceS}{20},%
109      {sentencenumeric}{30},{numeralsentence}{30},{SentenceN}{30},%
110      {sentenceoff}{40},{nosentence}{40},%
111      {long}{11},%
112      {short}{22},%
113      {numeric}{33},%
114      {paragraphonly}{44},{onlyparagraph}{44},%
115      {ParagraphOnly}{44},{OnlyParagraph}{44}%
116    }{#1}%
117 \ifx\FamilyKeyState\FamilyKeyStateProcessed
118   \@tempcnta=\@tempa\relax
119   \@tempcntb=\z@
120   \@whilenum \@tempcnta>9 \do{%
121     \advance\@tempcnta -10\relax
122     \advance\@tempcntb \@ne\relax
123   }%
124 \ifcase \@tempcnta
125 \or
126   \aftergroup\def\aftergroup\parcite@format
127   \aftergroup{\aftergroup0\aftergroup}%
128 \or
129   \aftergroup\def\aftergroup\parcite@format
130   \aftergroup{\aftergroup1\aftergroup}%
131 \or
132   \aftergroup\def\aftergroup\parcite@format
133   \aftergroup{\aftergroup2\aftergroup}%
134 \or
135   \aftergroup\def\aftergroup\parcite@format
136   \aftergroup{\aftergroup-\aftergroup1\aftergroup}%
137 \fi
138 \ifcase \@tempcntb
139 \or
140   \aftergroup\def\aftergroup\sentencecite@format
141   \aftergroup{\aftergroup0\aftergroup}%
142 \or
143   \aftergroup\def\aftergroup\sentencecite@format
144   \aftergroup{\aftergroup1\aftergroup}%
145 \or
146   \aftergroup\def\aftergroup\sentencecite@format
147   \aftergroup{\aftergroup2\aftergroup}%
```

```

148     \or
149     \aftergroup\def\aftergroup\sentencecite@format
150     \aftergroup{\aftergroup-\aftergroup1\aftergroup}%
151     \fi
152     \fi
153 \endgroup
154 }
155 \KOMADeclareDeprecatedOption[scrjura]{parcitename}{ref=parlong}
156 \KOMADeclareDeprecatedOption[scrjura]{sentencecitename}{ref=sentencelong}

```

Optionen ausführen.

```
157 \KOMAProcessOptions\relax
```

```

\Paragraph
\SubParagraph
\refParagraph 158 \providecommand*\DeprecatedParagraph{%
\refParagraphN 159 \PackageError{scrjura}{modification of old document needed}{%
\DeprecatedParagraph 160 It seem that this document was made for scrjura up to version
\ParagraphCompatibilityHacks 161 0.7a.\MessageBreak
162 Since scrjura version 0.9 \string\Paragraph, \string\SubParagraph, and all
163 depending\MessageBreak
164 commands, options, and counters have been renamed.\MessageBreak
165 You should replace the terms ‘Paragraph’ and ‘paragraph’ by ‘Clause’
166 and\MessageBreak
167 ‘clause’ if they are part of the name of a scrjura feature, otherwise
168 this\MessageBreak
169 document may produce severall additional error messages and maybe the
170 wrong\MessageBreak
171 result. Sorry for the inconvenience.%
172 }%
173 \ParagraphCompatibilityHacks
174 }
175 \newcommand*\ParagraphCompatibilityHacks{%
176 \PackageWarning{scrjura}{compatibility hacks for ‘\string\Paragraph’
177 executed.\MessageBreak
178 There is no support for documents using these hacks!\MessageBreak
179 There is no warranty for real compatibility!\MessageBreak
180 Even if the LaTeX run of the document doesn’t report\MessageBreak
181 any error, the result may be completely wrong.\MessageBreak
182 Therefore it is recommended to solve the problem,\MessageBreak
183 instead of trying to work around using the\MessageBreak
184 compatibility hacks%
185 }%
186 \gdef\Paragraph{\Clause}%
187 \gdef\SubParagraph{\SubClause}%
188 \gdef\c@Paragraph{\c@Clause}%
189 \gdef\cl@Paragraph{\cl@Clause}%
190 \gdef\c@SubParagraph{\c@SubClause}%

```

```

191 \gdef\cl@SubParagraph{\cl@SubClause}%
192 \gdef\theParagraph{\theClause}%
193 \gdef\theSubParagraph{\theSubClause}%
194 \gdef\refParagraph{\refClause}%
195 \gdef\refParagraphN{\refClauseN}%
196 \aliaskomafont{Paragraph}{Clause}%
197 \scr@ifundefinedorrelax{Paragraphmark}{-}{-%
198   \global\let\Clausemark\Paragraphmark
199 }%
200 }
201 \providecommand*\Paragraph{\DeprecatedParagraph\Paragraph}
202 \providecommand*\SubParagraph{\DeprecatedParagraph\SubParagraph}
203 \providecommand*\refParagraph{\DeprecatedParagraph\refParagraph}
204 \providecommand*\refParagraphN{\DeprecatedParagraph\refParagraphN}

```

`\scrjura@env@type` Dieses Makro wird in Abhängigkeit der gerade aktiven `scrjura`-Umgebung definiert.

```
205 \newcommand*\scrjura@env@type{-}
```

`\ellipsispar` Paragraphen auslassen aber mit zählen.

`\parellipsis`

```

206 \newcommand*\ellipsispar[1][1]{%
207   \begingroup
208     \KOMAOptions{parnumber=manual}\parellipsis\par
209     \addtocounter{par}{#1}%
210     \protected@write\auxout{}{-%
211       \string\newmaxpar{\scrjura@env@type}%
212         {\csname the\scrjura@env@type AbsoluteClause\endcsname}%
213         {\thepar}%
214     }%
215   \endgroup
216   \addtocounter{par}{-1}\refstepcounter{par}%
217   \ignorespaces
218 }
219 \newcommand*\parellipsis{%
220   \scr@ifundefinedorrelax{textellipsis}{\dots}{\textellipsis}%
221 }

```

`contract` Die Umgebung `contract` wird als Anweisung `\contract` definiert. Nichts desto trotz handelt es sich dabei um eine Umgebung und sollte sie immer als Umgebung verwendet werden! Die Umgebung kann nicht geschachtelt werden. Sie darf aber enden und dann erneut verwendet werden. Die Paragraphen werden jedoch über alle Verträge hinweg nummeriert. Das Ende eines Vertrags ist also genau genommen nur eine Unterbrechung.

```

\contract
contractClause
\thecontractClause
\contract@Clauseformat
\Clauseformat
contractSubClause
\thecontractSubClause
contractAbsoluteClause
222 \newcommand*\contract}{-%
223   \ifx\scrjura@env@type\@empty

```

```

224 \let\doendpe\scrjura@doendpe
225 \let\Clause\contract@paragraph
226 \let\c@Clause\c@contractClause
227 \edef\cl@Clause{\cl@Clause\cl@contractClause}%
228 \let\SubClause\contract@subparagraph
229 \let\c@SubClause\c@contractSubClause
230 \edef\cl@SubClause{\cl@SubClause\cl@contractSubClause}%
231 \let\Sentence\contract@sentence
232 \renewcommand*{\scrjura@env@type}{contract}%
233 \aliaskomafont{Clause}{contract.Clause}%
234 \else
235 \PackageError{scrjura}{nested ‘contract’ detected}{%
236 You may not use a ‘contract’ environment inside\MessageBreak
237 a ‘\scrjura@env@type’ environment or after loading\MessageBreak
238 package ‘scrjura’ with option ‘\scrjura@env@type’!}%
239 \fi
240 }
241 \let\if@scrjura@skiphyperref\iftrue
242 \let\cl@Clause@empty
243 \let\cl@SubClause@empty
244 \newcounter{contractClause}
245 \renewcommand*{\thecontractClause}{%
246 {\contract@Clauseformat{\arabic{contractClause}}}}
247 \DeclareRobustCommand*\contract@Clauseformat[1]{\Clauseformat{#1}}
248 \newcommand*{\Clauseformat}[1]{\S-#1}
249 \newcounter{contractSubClause}
250 \@addtoreset{SubClause}{Clause}
251 \renewcommand*{\thecontractSubClause}{%
252 {\theClause\alph{contractSubClause}}}}
253 \newcounter{contractAbsoluteClause}

```

`\DeclareNewJuraEnvironment` Verwendet `\@defjuraenvironment` um eine neue juristische Umgebung zu definieren.

```

254 \newcommand*{\DeclareNewJuraEnvironment}[1]{%
255 \@ifundefined{#1}{\expandafter\let\csname #1\expandafter\endcsname
256 \csname end#1\endcsname}{}%
257 \@ifundefined{#1}{\let\reserved@defjuraenvironment\@defjuraenvironment}{%
258 \PackageError{scrjura}{ignoring declaration of ‘#1’}{%
259 You’ve tried to declare jura environment ‘#1’, but
260 environment\MessageBreak
261 ‘#1’ or command
262 \expandafter\string\csname #1\endcsname\space or
263 \expandafter\string\csname end#1\endcsname\MessageBreak
264 already exists.\MessageBreak
265 Declaration will be ignored}%
266 \long\def\reserved@defjuraenvironment##1[##2]##3##4{)%
267 }%
268 \@ifnextchar [%]

```

```

269     {\reserved@defjuraenvironment{#1}}{\reserved@defjuraenvironment{#1} []}%
270 }
271 \@onlypreamble\DeclareNewJuraEnvironment

```

`\@defjuraenvironment` Diese Anweisung soll irgendwann einmal dazu dienen, weitere Umgebungen einfach definieren zu können. Allerdings wird das vor Version 1.0 offiziell nichts werden. Bis dahin funktioniert die Anweisung bzw. die damit definierten Umgebungen nicht korrekt und sollten nicht verwendet werden!

```

272 \DefineFamily{KOMAarg}
273 \DefineFamilyMember{KOMAarg}
274 \DefineFamilyKey{KOMAarg}{Clause}{%
275   \expandafter\gdef\csname \scrjura@env@type @Clause\endcsname{#1}%
276 }
277 \DefineFamilyKey{KOMAarg}{SubClause}{%
278   \expandafter\gdef\csname \scrjura@env@type @SubClause\endcsname{#1}%
279 }
280 \DefineFamilyKey{KOMAarg}{Sentence}{%
281   \expandafter\gdef\csname \scrjura@env@type @Sentence\endcsname{#1}%
282 }
283 \DefineFamilyKey{KOMAarg}{ClauseNumberFormat}{%
284   \expandafter\gdef\csname \scrjura@env@type @Clauseformat \endcsname
285     ##1{#1{##1}}
286   \expandafter\xdef\csname \scrjura@env@type @Clauseformat\endcsname{%
287     \noexpand\protect\expandafter\noexpand
288     \csname \scrjura@env@type @Clauseformat \endcsname
289   }%
290 }
291 \newcommand{\@defjuraenvironment}{%
292 \long\def\@defjuraenvironment#1[#2]#3#4{%

```

Die Zähler definieren:

```

293 \newcounter{#1Clause}%
294 \newcounter{#1AbsoluteClause}%
295 \newcounter{#1SubClause}%
296 \begingroup
297   \edef\scrjura@env@type{#1}%
298   \FamilyExecuteOptions[.scrjura.sty]{KOMAarg}{#2}%
299 \endgroup
300 \@ifundefined{#1@Clauseformat}{%
301   \expandafter\DeclareRobustCommand\expandafter*%
302   \csname #1@Clauseformat\endcsname[1]{\Clauseformat{##1}}%
303 }{}%

```

Umgebung:

```

304 \newenvironment{#1}{%
305   \par

```

```

306 \ifx\scrjura@env@type\@empty
307 \edef\scrjura@env@type{#1}%
308 \let\@doendpe\scrjura@doendpe
309 \expandafter\let\expandafter\c@Clause\csname c@#1Clause\endcsname
310 \edef\cl@Clause{\cl@Clause\csname cl@#1Clause\endcsname}%
311 \expandafter\let\expandafter\c@SubClause
312 \csname c@#1SubClause\endcsname
313 \edef\cl@SubClause{\cl@SubClause
314 \csname cl@#1SubClause\endcsname}%
315 \@ifundefined{#1@Clause}{%
316 \let\Clause\contract@paragraph
317 }{%
318 \expandafter\let\expandafter\Clause
319 \csname #1@Clause\endcsname
320 }%
321 \@ifundefined{#1@SubClause}{%
322 \let\SubClause\contract@subparagraph
323 }{%
324 \expandafter\let\expandafter\SubClause
325 \csname #1@SubClause\endcsname
326 }%
327 \@ifundefined{#1@Sentence}{%
328 \let\Sentence\contract@sentence
329 }{%
330 \expandafter\let\expandafter\Sentence\csname #1@Sentence\endcsname
331 }%
332 \@ifundefined{\scrjura@env@type @everypar}{%
333 \expandafter\let
334 \csname \scrjura@env@type @everypar\endcsname
335 \contract@everypar
336 }{%
337 \@ifundefined{scr@fnt@#1}{%
338 \@ifundefined{scr@fnt@instead@#1}{%
339 \aliaskomafont{Clause}{contract.Clause}%
340 }{%
341 \aliaskomafont{Clause}{\csname scr@fnt@instead@#1\endcsname}%
342 }%
343 }{%
344 \aliaskomafont{Clause}{contract.Clause}%
345 }%
346 #3%
347 \else
348 \PackageError{scrjura}{nested scrjura environments detected}{%
349 You must not use a ‘#1’ environment inside\MessageBreak
350 a ‘\scrjura@env@type’ environment or after loading\MessageBreak
351 package ‘scrjura’ with option ‘\scrjura@env@type’!}%
352 \fi
353 }{%
354 #4%

```

```

355   \par
356   }%
357 }

```

`\contract@paragraph` Das ist das Macro, das in Verträgen `\Clause` zur Verfügung stellt. Ein Vertrag besteht aus mehreren Paragraphen. Jeder Paragraph hat einige optionale Elemente, die (auf dem Umweg über `scrkbase`) über das `keyval`-Paket geregelt werden.

```

title  Der Titel, der Kolumnentitel und der Verzeichniseintrag des Paragraphen.
head   Der Titel setzt dabei zunächst auch die anderen beiden. Man kann die
entry  beiden aber auf leer setzen. Insbesondere aber kann man sie mit den no-
tocentry Optionen auch abschalten.
nohead
noentry 358 \define@key{contract}{title}{%
notocentry 359   \def\contract@title{#1}%
360   \ifx\contract@entry\relax\def\contract@entry{\contract@title}\fi
361   \ifx\contract@head\relax\def\contract@head{\contract@title}\fi
362 }
363 \define@key{contract}{entry}{%
364   \PackageWarning{scrjura}{deprecated option 'entry'.\MessageBreak
365     You should use option 'tocentry' instead of\MessageBreak
366     option 'entry'%
367   }%
368   \def\contract@entry{#1}}
369 \define@key{contract}{tocentry}{\def\contract@entry{#1}}
370 \define@key{contract}{noentry}[]{%
371   \PackageWarning{scrjura}{deprecated option 'noentry'.\MessageBreak
372     You should use option 'notocentry' instead of\MessageBreak
373     option 'noentry'%
374   }%
375   \let\contract@entry\relax}
376 \define@key{contract}{notocentry}[]{\let\contract@entry\relax}
377 \define@key{contract}{head}{\def\contract@head{#1}}
378 \define@key{contract}{nohead}[]{\let\contract@head\relax}

```

`number` Die Nummer kann mit dieser Option frei gestaltet werden. Es sind jedoch keine Paragraphen ohne Nummer erlaubt. Wird keine Nummer angegeben, so wird die Nummer automatisch gesetzt.

```

379 \define@key{contract}{number}{\def\contract@number{#1}}

```

`\contract@preskip` Über diese beiden Optionen kann der Abstand vor und nach einem Paragraphen gesetzt werden. Wird nichts angegeben, so wird die globale Voreinstellung verwendet, die übrigens mit `\setkeys{contract}{...}` gesetzt werden kann.

`\contract@postskip`

`preskip`

`postskip`

```

380 \newcommand*{\contract@preskip}{2\baselineskip}
381 \newcommand*{\contract@postskip}{\baselineskip}
382 \define@key{contract}{preskip}{\def\contract@preskip{#1}}
383 \define@key{contract}{postskip}{\def\contract@postskip{#1}}

```

`dummy` Über dies Option wird der Schalter `\ifcontract@dummy` auf `\iftrue` gesetzt.

`\ifcontract@dummy` Ist der Schalter `\iftrue`, dann wird der Paragraph nicht gesetzt. Es ist jedoch darauf zu achten, dass Absätze und Sätze des Paragraphen auf diese Weise nicht entfernt werden können. Werden jedoch Paragraphen gelöscht, so kann man damit das Loch in der Nummerierung erzeugen.

```

384 \newif\ifcontract@dummy
385 \define@key{contract}{dummy}[true]{\csname contract@dummy#1\endcsname}

```

`\contract@paragraph@font` Das muss noch auf die Element-Schnittstelle von KOMA-Script umgestellt werden!

```

386 \newkomafont{contract.Clause}{\sffamily\bfseries\large}
387 \newcommand*{\contract@paragraph@font}{\usekomafont{contract.Clause}%
388 \@hangfrom}

```

```

@AbsClause
\theH@AbsClause
\theHClause
\theHSubClause
389 % \textsf{hyperref}-Code, der noch nicht getestet ist!
390 \newcounter{@AbsClause}
391 \newcommand*{\theH@AbsClause}{P-\arabic{@AbsClause}}
392 \newcommand*{\theHClause}{\theH@AbsClause}
393 \newcommand*{\theHSubClause}{\theH@AbsClause}

```

Zunächst wird vorübergehend auf manuelle Absatznummern umgeschaltet, weil natürlich innerhalb der Überschrift keine Nummer gesetzt werden soll. Trotzdem werden generell die Absatznummern eingeschaltet. Dann werden die Optionen initialisiert und ausgeführt.

```

394 \newcommand*{\contract@paragraph}[1]{%
395 \stepcounter{\scrjura@env@type AbsoluteClause}%
396 \ManualPar\parnumbertrue
397 \let\contract@title\relax
398 \let\contract@entry\relax
399 \let\contract@head\relax
400 \let\contract@number\relax
401 \contract@dummyfalse
402 \ifx\relax#1\relax\else\setkeys{contract}{#1}\fi

```

Wenn es kein Blindparagraph ist, werden Vorbereitung für nach der Überschrift getroffen und der vertikale Abstand eingefügt.

```
403 \ifcontract@dummy\else
404   \par
405   \@afterindentfalse
406   \addvspace{\contract@preskip}%
407 \fi
```

Wenn die Nummer nicht manuell gesetzt wurde, wird die nächste Nummer verwendet. Im anderen Fall muss die Nummer gesetzt und auch dafür gesorgt werden, dass sowohl ein Label als auch hyperref diese Nummer verwenden.

```
408 \ifx\contract@number\relax
409   \let\p@Clause\@empty
410   \let\theClause\thecontractClause
411   \refstepcounter{Clause}%
412 \else
413   \begingroup
414     \let\@elt\@stpelt
415     \cl@Clause
416   \endgroup
417   \protected@edef\theClause{%
418     \csname \scrjura@env@type @Clauseformat\endcsname{\contract@number}%
419   }%
420   \protected@edef\@currentlabel{\theClause}%
421 \fi
422 \stepcounter{@AbsClause}%
423 \begingroup\expandafter\expandafter\expandafter\endgroup
424 \expandafter\ifx\csname if@skiphyperref\endcsname\relax
425 \else
426   \expandafter\let\csname if@scrjura@skiphyperref\expandafter\endcsname
427   \csname if@skiphyperref\endcsname
428 \fi
429 \if@scrjura@skiphyperref\else
430   \hyper@refstepcounter{@AbsClause}%
431   \typeout{absolute Nummer: \the@AbsClause~JLabel: '\@currentHref'}%
432 \fi
```

Zwecks Vereinfachung wird ab hier so getan, als wäre es ein Unterabschnitt.

```
433 \let\theSubClause\theClause
```

Außer für Blindparagraphen wird dann die Überschrift gesetzt, die Verzeichniseinträge vorgenommen und der Kolumnentitel angepasst.

```
434 \ifcontract@dummy\else
435   \begingroup
436     \if@juratitlepagebreak\else\interlinepenalty\@M\fi
```

```

437     \contract@paragraph@font{\theClause
438     \ifx\contract@title\relax\else\enskip\fi}%
439     \contract@title
440     \ifx\contract@entry\relax\else
441     \addxcontentsline{toc}{cpar}[\theClause]\contract@entry
442     \addxcontentsline{cpa}{cpar}[\theClause]\contract@entry
443     \fi
444     \ifx\contract@head\relax\else
445     \expandafter\Clausemark\expandafter{%
446     \expandafter\theSubClause\expandafter\enskip\contract@head}%
447     \fi
448     \par
449     \endgroup\nobreak\vskip\contract@postskip

```

Zum Schluss wird noch dafür gesorgt, dass auch die Absätze korrekt nummeriert werden können etc.

```

450     \scrjura@afterheading
451     \fi
452 }

```

`\contract@subparagraph` Das entspricht bis auf wenige Kleinigkeiten, die Unterparagraphen betreffen, `\contract@paragraph`

```

453 \newcommand*{\contract@subparagraph}[1]{%
454 \stepcounter{\scrjura@env@type AbsoluteClause}%
455 \ManualPar\parnumbertrue
456 \let\contract@title\relax
457 \let\contract@entry\relax
458 \let\contract@head\relax
459 \let\contract@number\relax
460 \contract@dummysfalse
461 \ifx\relax#1\relax\else\setkeys{contract}{#1}\fi
462 \ifcontract@dummys\else
463     \par
464     \@afterindentfalse
465     \vskip\contract@preskip
466     \fi
467 \ifx\contract@number\relax
468     \let\p@SubClause\@empty
469     \let\theSubClause\thecontractSubClause
470     \refstepcounter{SubClause}%
471 \else
472     \begingroup
473     \let\@elt\@stpelt
474     \cl@SubClause
475     \endgroup
476 \protected@edef\theSubClause{\theClause\contract@number}%
477 \protected@edef\@currentlabel{\theSubClause}%

```

```

478 \fi
479 \stepcounter{@AbsClause}%
480 \begingroup\expandafter\expandafter\expandafter\endgroup
481 \expandafter\ifx\csname if@skiphyperref\endcsname\relax
482 \else
483   \expandafter\let\csname if@scrjura@skiphyperref\expandafter\endcsname
484   \csname if@skiphyperref\endcsname
485 \fi
486 \if@scrjura@skiphyperref\else
487   \hyper@refstepcounter{@AbsClause}%
488   \typeout{absolute Nummer: \the@AbsClause^^JLabel: '\@currentHref'}%
489 \fi
490 \ifcontract@dummys\else
491   \begingroup
492     \if@juratitlepagebreak\else\interlinepenalty\M\fi
493     \contract@paragraph@font{\theSubClause
494       \ifx\contract@title\relax\else\enskip\fi}%
495     \contract@title
496     \ifx\contract@entry\relax\else
497       \addxcontentsline{toc}{cpar}[\theSubClause]\contract@entry
498       \addxcontentsline{cpa}{cpar}[\theSubClause]\contract@entry
499     \fi
500     \ifx\contract@head\relax\else
501       \expandafter\Clausemark\expandafter{%
502         \expandafter\theSubClause\expandafter\enskip\contract@head}%
503     \fi
504     \par
505   \endgroup
506   \nobreak\vskip\contract@postskip
507   \scrjura@afterheading
508 \fi
509 }

```

`\AutoPar` Automatische oder manuelle Absatznummern für alle Umgebungen aktivieren.  
`\ManualPar`

```

510 \newcommand*{\AutoPar}{%
511   \expandafter\let\expandafter\scrjura@everypar
512   \csname \scrjura@env@type @everypar\endcsname
513 }
514 \newcommand*{\ManualPar}{%
515   \let\scrjura@everypar\relax
516 }

```

`\scrjura@afterheading` Entspricht `\afterheading` mit Erweiterungen für automatischen Absatznummern.

```

517 \newcommand*{\scrjura@afterheading}{%

```

```

518 \@nobreaktrue
519 \everypar{%
520   \if@nobreak
521     \@nobreakfalse
522     \clubpenalty \@M
523   \if@afterindent \else
524     {\setbox\z@\lastbox}%
525   \fi
526 \else
527   \clubpenalty \@clubpenalty
528   \everypar{%
529     \scrjura@everypar
530   }%
531 \fi
532 \scrjura@everypar
533 }%
534 \AutoPar
535 }

```

`\scrjura@everypar` Die Anweisung, die am Anfang von jedem Absatz auszuführen ist, um die Nummer zu setzen. Außerhalb der Umgebungen ist das nichts.

```
536 \newcommand*{\scrjura@everypar}{}

```

`\@doendpe` Diese Anweisung wird von L<sup>A</sup>T<sub>E</sub>X verwendet, um am Ende von Umgebungen die kurz-, mittel- und langwirkenden Absatzaktionen zurückzusetzen. Damit hier nicht die automatische Absatznummerierung abgeschaltet wird, wird sie neu eingefügt.

`\scrjura@doendpe`

```

537 \CheckCommand*\@doendpe{\@endpetrue
538   \def\par{\@restorepar\everypar}\par\@endpefalse}\everypar
539   {\setbox\z@\lastbox}\everypar}\@endpefalse}}
540 \newcommand*{\scrjura@doendpe}{%
541   \@endpetrue
542   \def\par{%
543     \@restorepar\everypar{%
544       \csname scrjura@everypar\endcsname
545     }%
546   \par\@endpefalse
547 }%
548 \everypar{%
549   {\setbox\z@\lastbox}\everypar{%
550     \csname scrjura@everypar\endcsname
551   }%
552   \@endpefalse
553 }%

```

554 }

`\l@cpa`r Verzeichnis-Eintrag für einen Vertrags-Paragraphen.

```
555 \newcommand*{\l@cpa}{2}{%
556   \ifnum\value{juratoclevel}>\value{tocdepth}\else
557     \scr@ifundefinedorrelax{bprot@dottedtocline}{%
558       \dottedtocline
559     }{%
560       \bprot@dottedtocline
561     }{\value{juratoclevel}}{\cpar@indent}{\cpar@numberwidth}{#1}{#2}%
562   \fi
563 }
```

`\scrjura@separator` Wird verwendet, damit Leerzeichen am Anfang oder Ende entfernt werden können.

```
564 \DeclareRobustCommand*\scrjura@separator}[1]{#1}
```

`\scrjura@usetype` In der Voreinstellung macht das erst einmal gar nichts, außer dass es robust ist und `\jura@@usetype` aufruft.

`\scrjura@@usetype` Das wiederum ist nicht robust und kann dadurch einfacher undefiniert werden. In der Voreinstellung macht das dann wirklich nichts, außer das Argument zu fressen.

```
565 \DeclareRobustCommand*\scrjura@usetype}[1]{\scrjura@@usetype{#1}}
566 \newcommand*\scrjura@@usetype}[1]{}
```

`\contract@everypa`r `\scrjura@everpa`r für Verträge.

`\ifparnumber` Der Schalter gibt an, ob überhaupt mit Absatznummern gearbeitet werden soll. Sind Absatznummern deaktiviert, werden auch keine manuellen  
`par` Absatznummern gesetzt. Die Absätze werden dann auch nicht gezählt. An-  
`\thepa`r sonst werden die Absätze mit `\thepa`r nummeriert. Wichtig ist, dass der  
`\theHpa`r Absatzzähler mit den Paragraphen und den Unterparagraphen zurückge-  
`\parformat` setzt wird. Außerdem muss beim Zitieren als Elternobjekt der Paragraph  
`\p@pa`r mit ausgegeben werden.

```
567 \newcounter{par}
568 \renewcommand*\thepa{\arabic{par}}
569 \newcommand*\theHpa{\theH@AbsClause-\Roman{par}}
570 \newcommand*\parformat{(\thepa)}
571 \newkomafont{parnumber}{}
572 \renewcommand*\p@pa{\scrjura@usetype{\scrjura@env@type}\theSubClause\scrjura@s
573 \@addtoreset{par}{Clause}
574 \@addtoreset{par}{SubClause}
```

```

575 \newcommand*{\contract@everypar}{%
576   \ifparnumber
577     \ifx\scrjura@special@par\relax
578       \ifx\scrjura@special@reset@par\relax\else
579         \global\let\thepar\scrjura@special@reset@par
580         \global\let\scrjura@special@reset@par\relax
581       \fi
582       \refstepcounter{par}%
583       \refstepcounter{sentence}%
584     \else
585       \ifx\scrjura@special@reset@par\relax
586         \global\let\scrjura@special@reset@par\thepar
587       \fi
588       \global\let\thepar\scrjura@special@par
589       \global\let\scrjura@special@par\relax
590       \setcounter{sentence}{0}\refstepcounter{sentence}%
591     \fi
592   \begingroup
593     \protected@write\@auxout{%
594       \expandafter\let\csname \scrjura@env@type @Clauseformat\endcsname
595       \@firstofone
596     }{%
597       \string\newmaxpar{\scrjura@env@type}%
598                           {\csname the\scrjura@env@type AbsoluteClause\endcsname}%
599                           {\thepar}%
600     }%
601     \getmaxpar\@tempa{\scrjura@env@type}%
602                           {\csname the\scrjura@env@type AbsoluteClause\endcsname}%
603     \typeout{Stored max is \@tempa}%
604     \def\reserved@a##1\@nnil{\def\@tempa{##1}}%
605     \afterassignment\reserved@a\@tempcnta=0\@tempa\relax\@nnil
606     \ifnum \@tempcnta>\@ne
607       {\usekomafont{parnumber}{\parformat\nobreakspace}}%
608     \else
609       \def\reserved@a{\relax}%
610       \ifx\@tempa\reserved@a
611         \else
612           {\usekomafont{parnumber}{\parformat\nobreakspace}}%
613         \fi
614       \fi
615     \endgroup
616   \else
617     \setcounter{sentence}{-1}\refstepcounter{sentence}%
618   \fi
619 }

```

`\thisparnumber` Mit dieser Anweisung kann man eine Absatznummer manuell vergeben.  
`\scrjura@special@par` Allerdings muss die Nummer voll expandierbar sein!  
`\scrjura@special@reset@par`

```

620 \newcommand*{\thisparnumber}[1]{%
621   \def\scrjura@special@par{#1}%
622 }
623 \newcommand*{\scrjura@special@par}{}
624 \let\scrjura@special@par\relax
625 \newcommand*{\scrjura@special@reset@par}{}
626 \let\scrjura@special@reset@par\relax

```

`\refL` Das gleiche wie `\ref` aber zwingend mit der Langform.

```

\ref@L
627 \newcommand*{\refL}{\@ifstar {\ref@L*}{\ref@L{}}}
628 \newcommand*{\ref@L}[2]{%
629   \begingroup
630     \def\parcite@format{0}%
631     \let\sentencecite@format\parcite@format
632     \ref#1{#2}%
633   \endgroup
634 }

```

`\refS` Das gleiche wie `\ref` aber zwingend mit der Kurzform.

```

\ref@S
635 \newcommand*{\refS}{\@ifstar {\ref@S*}{\ref@S{}}}
636 \newcommand*{\ref@S}[2]{%
637   \begingroup
638     \def\parcite@format{1}%
639     \let\sentencecite@format\parcite@format
640     \ref#1{#2}%
641   \endgroup
642 }

```

`\refN` Das gleiche wie `\ref` aber zwingend mit der numerischen Form.

```

\ref@N
643 \newcommand*{\refN}{\@ifstar {\ref@N*}{\ref@N{}}}
644 \newcommand*{\ref@N}[2]{%
645   \begingroup
646     \def\parcite@format{2}%
647     \let\sentencecite@format\parcite@format
648     \ref#1{#2}%
649   \endgroup
650 }

```

`\refClause` Zitiert nur den Paragraphen eines Paragraphen, eines Absatzes oder eines Satzes. Für mehr Kompatibilität mit `hyperref` gibt es mit `hyperref` auch eine Sternform. Ohne `hyperref` produziert diese nur Unsinn.

```

651 \newcommand*{\refClause}{%
652   \@ifstar {\ref@Clause*}{\ref@Clause{}}
653 }

```

```

654 \newcommand*{\ref@Clause}[2]{%
655   \expandafter\ifx\csname r@#2\endcsname\relax
656     \ref#1{#2}%
657   \else
658     \begingroup

```

Alle Teile der Referenz bis auf den ersten in \@tempb ablegen.

```

659     \expandafter\expandafter\expandafter\expandafter
660     \expandafter\expandafter\expandafter\def
661     \expandafter\expandafter\expandafter\expandafter
662     \expandafter\expandafter\expandafter\@tempb
663     \expandafter\expandafter\expandafter\expandafter
664     \expandafter\expandafter\expandafter{%
665     \expandafter\expandafter\expandafter\@gobble\csname r@#2\endcsname}%

```

Den ersten Teil der Referenz in \@tempa ablegen.

```

666     \def\@tempc##1##2\@nil{##1}%
667     \let\scrjura@separator\@gobble
668     \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
669     \csname r@#2\endcsname\noexpand\@nil}%

```

Den ersten Teil von \@tempa in \@tempa ablegen.

```

670     \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
671     \@tempa\@nil}%
672     \let\@protect\protect
673     \let\protect\noexpand
674     \expandafter\edef\csname r@#2\endcsname{\{\@tempa\}\@tempb}%
675     \let\protect\@protect
676     \ref#1{#2}%
677   \endgroup
678 \fi
679 }

```

\refClauseN Zitiert nur den Paragraphen eines Paragraphen, eines Absatzes oder eines  
\ref@ClauseN Satzes. Für mehr Kompatibilität mit hyperref gibt es mit hyperref auch  
eine Sternform. Ohne hyperref produziert diese nur Unsinn.

```

680 \newcommand*{\refClauseN}{%
681   \@ifstar {\ref@ClauseN*}{\ref@ClauseN{}}
682 }
683 \newcommand*{\ref@ClauseN}[2]{%
684   \begingroup
685     \let\Clauseformat\relax
686     \ref@Clause{#1}{#2}%
687   \endgroup
688 }

```

`\refPar` Zitiert nur den Absatz eines Absatzes oder eines Satzes. Für mehr Kompatibilität mit `hyperref` gibt es mit `hyperref` auch eine Sternform. Ohne `hyperref` produziert diese nur Unsinn.

```

689 \newcommand*{\refPar}{%
690   \@ifstar {\ref@Par*}{\ref@Par{}}
691 }
692 \newcommand*{\ref@Par}[2]{%
693   \expandafter\ifx\csname r@#2\endcsname\relax
694     \ref#1{#2}%
695   \else
696     \begingroup

```

Alle Teile der Referenz bis auf den ersten in `\@tempb` ablegen.

```

697   \expandafter\expandafter\expandafter\expandafter
698   \expandafter\expandafter\expandafter\def
699   \expandafter\expandafter\expandafter\expandafter
700   \expandafter\expandafter\expandafter\@tempb
701   \expandafter\expandafter\expandafter\expandafter
702   \expandafter\expandafter\expandafter{%
703     \expandafter\expandafter\expandafter\@gobble\csname r@#2\endcsname}%

```

Den ersten Teil der Referenz in `\@tempa` ablegen.

```

704   \def\@tempc##1##2\@nil{##1}%
705   \let\scrjura@separator\@gobble
706   \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
707     \csname r@#2\endcsname\noexpand\@nil}%

```

Den zweiten Teil von `\@tempa` in `\@tempa` ablegen.

```

708   \def\@tempc##1##2##3\@nil{##2}%
709   \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
710     \@tempa{%
711       \protect\G@refundefinedtrue
712       \nfss@text{\reset@font\bfseries ??}%
713       \latex@warning{Reference ‘#2’ on page \thepage \space
714         with undefined par number}%
715     }\noexpand\@nil}%
716   \let\@@protect\protect
717   \let\protect\noexpand
718   \expandafter\edef\csname r@#2\endcsname{\{\@tempa}\@tempb}%
719   \let\protect\@@protect
720   \ref#1{#2}%
721 \endgroup
722 \fi
723 }

```

```

\refParL
\ref@ParX 724 % Das Gleiche zwingend lang.
           725 \newcommand*{\refParL}{%
           726   \@ifstar {\ref@ParX0*}{\ref@ParX0{}}
           727 }
           728 \newcommand*{\ref@ParX}[3]{%
           729   \begingroup
           730     \def\parcite@format{#1}%
           731     \let\sentencecite@format\parcite@format
           732     \ref@Par{#2}{#3}%
           733   \endgroup
           734 }

```

```

\refParS  Das Gleiche zwingend kurz.

           735 \newcommand*{\refParS}{%
           736   \@ifstar {\ref@ParX1*}{\ref@ParX1{}}
           737 }

```

```

\refParN  Das Gleiche zwingend numerisch.

           738 \newcommand*{\refParN}{%
           739   \@ifstar {\ref@ParN2*}{\ref@ParN2{}}
           740 }
           741 \newcommand*{\ref@ParN}[2]{%
           742   \@ifnextchar [%]
           743     {\ref@@ParN{#1}{#2}}%
           744     {\ref@ParX{#1}{#2}}%
           745 }
           746 \newcommand*{\ref@@ParN}{-}
           747 \def\ref@@ParN#1#2[#3]#4{%
           748   \begingroup
           749     \renewcommand*{\parnumericformat}[1]{%
           750       \csname @#3\endcsname{\number ##1\relax}%
           751     }%
           752     \ref@ParX{#1}{#2}{#4}%
           753   \endgroup
           754 }

```

\refSentence Zitiert nur den Satz eines Satzes. Für mehr Kompatibilität mit hyperref  
\ref@Sentence gibt es mit hyperref auch eine Sternform. Ohne hyperref produziert diese  
nur Unsinn.

```

           755 \newcommand*{\refSentence}{%
           756   \@ifstar {\ref@Sentence*}{\ref@Sentence{}}
           757 }
           758 \newcommand*{\ref@Sentence}[2]{%

```

```

759 \expandafter\ifx\csname r@#2\endcsname\relax
760   \ref#1{#2}%
761 \else
762   \begingroup

```

Alle Teile der Referenz bis auf den ersten in \@tempb ablegen.

```

763   \expandafter\expandafter\expandafter\expandafter
764   \expandafter\expandafter\expandafter\def
765   \expandafter\expandafter\expandafter\expandafter
766   \expandafter\expandafter\expandafter\@tempb
767   \expandafter\expandafter\expandafter\expandafter
768   \expandafter\expandafter\expandafter{%
769   \expandafter\expandafter\expandafter\@gobble\csname r@#2\endcsname}%

```

Den ersten Teil der Referenz in \@tempa ablegen.

```

770   \def\@tempc##1##2\@nil{##1}%
771   \let\scrjura@separator\@gobble
772   \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
773   \csname r@#2\endcsname\noexpand\@nil}%

```

Den dritten Teil von \@tempa in \@tempa ablegen.

```

774   \def\@tempc##1##2##3##4\@nil{##3}%
775   \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
776   \@tempa}{%
777   \protect\G@refundefinedtrue
778   \nfss@text{\reset@font\bfseries ??}%
779   \@latex@warning{Reference '#2' on page \thepage \space
780   with undefined sentence number}%
781   }\noexpand\@nil}%
782   \let\@@protect\protect
783   \let\protect\noexpand
784   \expandafter\edef\csname r@#2\endcsname{{\@tempa}\@tempb}%
785   \let\protect\@@protect
786   \ref#1{#2}%
787 \endgroup
788 \fi
789 }

```

```

\refSentenceL
\ref@SentenceX

```

```

790 % Das Gleiche zwingend lang.
791 \newcommand*{\refSentenceL}{%
792   \@ifstar {\ref@SentenceX0*}{\ref@SentenceX0{}}
793 }
794 \newcommand*{\ref@SentenceX}[3]{%
795   \begingroup
796   \def\parcite@format{#1}%

```

```

797 \let\sentencecite@format\parcite@format
798 \ref@Sentence{#2}{#3}%
799 \endgroup
800 }

```

`\refSentenceS` Das Gleiche zwingend kurz.

```

801 \newcommand*{\refSentenceS}{%
802 \@ifstar {\ref@SentenceX1*}{\ref@SentenceX1{}}
803 }

```

`\refSentenceN` Das Gleiche zwingend numerisch.

```

804 \newcommand*{\refSentenceN}{%
805 \@ifstar {\ref@SentenceX2*}{\ref@SentenceX2{}}
806 }

```

`\contract@sentence` Sätze nummerieren.

`sentence` Der Zähler wird für die Nummerierung der Sätze verwendet. Dabei ist  
`\thesentence` wichtig, dass beim Zitieren der Sätze als Elternobjekt der Absatz mit aus-  
`\theHsentence` gegeben wird.

```

\p@sentence
807 \newcounter{sentence}[par]
808 \renewcommand*{\thesentence}{\arabic{sentence}}
809 \newcommand*{\theHsentence}{\theHpar-\arabic{sentence}}
810 \renewcommand*{\p@sentence}{\expandafter\p@@sentence}
811 \newcommand*{\p@@sentence}[1]{\p@par{\par@cite{\thepar}}%
812 \scrjura@separator{\nobreakspace}}{\sentence@cite{#1}}
813 \newcommand*{\contract@sentence}{%
814 \refstepcounter{sentence}%
815 \textsuperscript{\thesentence}\nobreak\hskip\z@
816 }

```

`\parciteformat` Zitierstil für Absätze.

```

817 \DeclareRobustCommand*{\par@cite}[1]{\parciteformat{#1}}
818 \newcommand*{\parciteformat}[1]{%
819 \ifcase \parcite@format
820 \expandafter\parlongformat
821 \or
822 \expandafter\parshortformat
823 \or
824 \expandafter\parnumericformat
825 \else
826 \unskip\expandafter\@gobble
827 \fi
828 {#1}%

```

829 }

`\sentenceciteformat` Zitierstil für Sätze. In der Voreinstellung ist das `\@arabic`.

```
830 \DeclareRobustCommand*\sentence@cite}[1]{\sentenceciteformat{#1}}
831 \newcommand*\sentenceciteformat}[1]{%
832   \ifcase \sentencecite@format
833     \expandafter\sentencelongformat
834   \or
835     \expandafter\sentenceshortformat
836   \or
837     \expandafter\sentencenumericformat
838   \else
839     \unskip\expandafter\@gobble
840   \fi
841   {#1}%
842 }
```

`\parlongformat` Formatierung in den sechs Formen.

```
\parshortformat
\parnumericformat 843 \newcommand*\parlongformat}[1]{\parname~#1}
844 \newcommand*\parshortformat}[1]{\parshortname~#1}
\sentencelongformat 845 \newcommand*\parnumericformat}[1]{\@Roman{\number #1\relax}}
846 \newcommand*\sentencelongformat}[1]{\sentencename~#1}
\sentenceshortformat 847 \newcommand*\sentenceshortformat}[1]{\sentenceshortname~#1}
848 \newcommand*\sentencenumericformat}[1]{\@arabic{\number #1\relax}.}
```

`\parname` Der Name eines Absatzes und eines Satzes. Die englischen Namen wurden  
`\parshortname` von »m.eik« beigesteuert.

```
\sentencename
\sentenceshortname 849 \newcommand*\parname}{Paragraph}
\scrjura@lang@error 850 \AtBeginDocument{%
851   \providecaptionname{german,ngerman,austrian,naustrian}\parname{Absatz}%
852   \providecaptionname{german,ngerman,austrian,naustrian}\parshortname{Abs.}%
853   \providecaptionname{german,ngerman,austrian,naustrian}\sentencename{Satz}%
854   \providecaptionname{german,ngerman,austrian,naustrian}\sentenceshortname{S.}%
855   \providecaptionname{english,american,british,canadian,USenglish,UKenglish}\parname{Paragraph}%
856   \providecaptionname{english,american,british,canadian,USenglish,UKenglish}\parshortname{Par.}%
857   \providecaptionname{english,american,british,canadian,USenglish,UKenglish}\sentencename{Sentence}%
858   \providecaptionname{english,american,british,canadian,USenglish,UKenglish}\sentenceshortname{S.}%
859 }
860 \providecommand*\parname}{\scrjura@lang@error{\parname}}
861 \providecommand*\parshortname}{\scrjura@lang@error{\parshortname}}
862 \providecommand*\sentencename}{\scrjura@lang@error{\sentencename}}
863 \providecommand*\sentenceshortname}{\scrjura@lang@error{\sentenceshortname}}
864 \newcommand*\scrjura@lang@error}[1]{%
865   \PackageError{scrjura}{%
866     current language not supported%
```

```

867 }{%
868   Currently scrjura only supports languages ‘german’, ‘ngerman’,
869   ‘austrian’,\MessageBreak
870   ‘naustrian’, ‘english’, ‘american’, ‘british’, ‘canadian’, ‘USenglish’,
871   and\MessageBreak
872   ‘UKenglish’.\MessageBreak
873   It seems, that you are using another language (maybe ‘\language’) or
874   that\MessageBreak
875   your language selection isn’t compatible to package ‘babel’.\MessageBreak
876   Because of this you have to define ‘\string#1’ by yourself!\MessageBreak
877   It would be nice if you’ll send your definitions to the author.%
878 }%
879 \textbf{??}%
880 }

```

`\newmaxpar` Noch zwei Hilfsmakros, um Zähler in der aux-Datei zwischenzuspeichern  
`\getmaxpar` und auch dann Werte zu holen, wenn sie nicht in der aux-Datei stehen.

```

881 \newcommand*{\newmaxpar}[3]{%
882   \begingroup
883   \expandafter\let\csname #1@Clauseformat\endcsname\@firstofone
884   \protected@edef\@tempa{#2}\@onelevel@sanitize\@tempa
885   \expandafter\xdef\csname max@#1@\@tempa\endcsname{#3}%
886   \endgroup
887 }
888 \newcommand*{\getmaxpar}[3]{%
889   \begingroup
890   \expandafter\let\csname #2@Clauseformat\endcsname\@firstofone
891   \protected@edef\@tempa{#3}%
892   \@onelevel@sanitize\@tempa
893   \expandafter\ifx \csname max@#2@\@tempa\endcsname\relax
894     \edef\@tempa{\endgroup\edef\noexpand#1{\expandafter\the\value{par}}}%
895   \else
896     \edef\@tempa{\endgroup
897       \edef\noexpand#1{\csname max@#2@\@tempa\endcsname}}%
898   \fi
899   \@tempa
900 }

```

Da manche Anwender `scrjura` aus einem Dokument entfernen, ohne die `aux`-Dateien zu löschen, sehen wir eine Notlösung vor, um Fehlermeldungen wegen nicht definiertem `\newmaxpar` zu vermeiden.

```

901 \AtBeginDocument{%
902   \immediate\write\@mainaux{%
903     \string\providecommand*{\string\newmaxpar}[3]{
904   }%
905 }

```

## Index

Numbers written in *italics* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols		I	
@AbsClause (Zähler) . . . . .	389	\if@juratitlepagebreak . . . . .	44
\@defjuraenvironment . . . . .	<u>272</u>	\if@juratotoc . . . . .	<u>15</u>
\@doendpe . . . . .	<u>537</u>	\ifcontract@dummy . . . . .	384
<b>A</b>		\ifparnumber . . . . .	12, <u>567</u>
\AutoPar . . . . .	12, <u>510</u>	<b>J</b>	
<b>C</b>		juratitlepagebreak (Option) <u>0</u> , 44	
\Clause . . . . .	5	juratocindent (Option) . . . . .	<u>32</u>
\Clauseformat . . . . .	<u>222</u>	\juratoclevel . . . . .	4
\Clausemark . . . . .	<u>53</u>	juratoclevel (Zähler) . . . . .	<u>15</u>
\contract . . . . .	<u>222</u>	juratocnumberwidth (Option) . . .	<u>32</u>
contract (Umgebung) . . . . .	<u>222</u>	juratotoc (Option) . . . . .	<u>0</u> , <u>15</u>
contract (Option) . . . . .	<u>0</u> , <u>12</u>	<b>L</b>	
\contract@Clauseformat . . . . .	<u>222</u>	\l@cpar . . . . .	<u>555</u>
\contract@everypar . . . . .	<u>567</u>	<b>M</b>	
\contract@paragraph . . . . .	<u>358</u>	\ManualPar . . . . .	12, <u>510</u>
\contract@paragraph@font . . .	<u>386</u>	markboth (Option) . . . . .	<u>53</u>
\contract@postskip . . . . .	<u>380</u>	markright (Option) . . . . .	<u>53</u>
\contract@preskip . . . . .	<u>380</u>	<b>N</b>	
\contract@sentence . . . . .	<u>807</u>	\newmaxpar . . . . .	12, <u>881</u>
\contract@subparagraph . . . . .	<u>453</u>	noentry (Option) . . . . .	<u>358</u>
contractAbsoluteClause (Zähler)	<u>222</u>	nohead (Option) . . . . .	<u>358</u>
contractClause (Zähler) . . . . .	<u>222</u>	notocentry (Option) . . . . .	<u>358</u>
contractSubClause (Zähler) . . .	<u>222</u>	number (Option) . . . . .	<u>379</u>
\cpar@indent . . . . .	<u>32</u>	<b>O</b>	
\cpar@numberwidth . . . . .	<u>32</u>	Optionen:	
<b>D</b>		contract . . . . .	<u>0</u> , <u>12</u>
\DeclareNewJuraEnvironment 10,	<u>254</u>	dummy . . . . .	384
\DeprecatedParagraph . . . . .	<u>158</u>	entry . . . . .	<u>358</u>
dummy (Option) . . . . .	<u>384</u>	head . . . . .	<u>358</u>
<b>E</b>		juratitlepagebreak . . . . .	<u>0</u> , 44
\ellipsispar . . . . .	6, <u>206</u>	juratocindent . . . . .	<u>32</u>
entry (Option) . . . . .	<u>358</u>	juratocnumberwidth . . . . .	<u>32</u>
<b>G</b>		juratotoc . . . . .	<u>0</u> , <u>15</u>
\getmaxpar . . . . .	12, <u>881</u>	markboth . . . . .	<u>53</u>
<b>H</b>		markright . . . . .	<u>53</u>
head (Option) . . . . .	<u>358</u>	noentry . . . . .	<u>358</u>
		nohead . . . . .	<u>358</u>

notocentry	358	\refL	627
number	379	\refN	643
paragraphmark	0, 53	\refPar	12, 689
parcitename	98	\refParagraph	158
parnumber	45	\refParagraphN	158
postskip	0, 380	\refParL	724
preskip	0, 380	\refParN	738
ref	98	\refParS	735
sentencecitename	98	\refS	635
title	358	\refSentence	12, 755
tocentry	358	\refSentenceL	790
		\refSentenceN	804
		\refSentenceS	801
<b>P</b>			
\p@par	567		
\p@sentence	807		
\Par	12		
\par	12		
par (Zähler)	567		
\Paragraph	158		
\ParagraphCompatibilityHacks	158		
paragraphmark (Option)	0, 53		
\parcite@fromat	98		
\parciteformat	12, 817		
parcitename (Option)	98		
\parellipsis	6, 206		
\parformat	12, 567		
\parlongformat	843		
\parname	12, 849		
parnumber (Option)	45		
\parnumberfalse	9, 12		
\parnumbertrue	9, 12		
\parnumericformat	843		
\parshortformat	843		
\parshortname	849		
postskip (Option)	0, 380		
preskip (Option)	0, 380		
<b>S</b>			
\scrjura@usetype	565		
\scrjura@afterheading	517		
\scrjura@doendpe	537		
\scrjura@env@type	205		
\scrjura@everypar	536		
\scrjura@lang@error	849		
\scrjura@separator	564		
\scrjura@special@par	620		
\scrjura@special@reset@par	620		
\scrjura@usetype	565		
\Sentence	9		
\sentence	12		
sentence (Zähler)	807		
\sentencecite@fromat	98		
\sentenceciteformat	12, 830		
sentencecitename (Option)	98		
\sentencelongformat	843		
\sentencename	12, 849		
\sentencenumericformat	843		
\sentenceshortformat	843		
\sentenceshortname	849		
\SubClause	6		
\SubParagraph	158		
<b>R</b>			
ref (Option)	98		
\ref@Clause	651		
\ref@ClauseN	680		
\ref@L	627		
\ref@N	643		
\ref@Par	689		
\ref@ParX	724		
\ref@S	635		
\ref@Sentence	755		
\ref@SentenceX	790		
\ref@Clause	12, 651		
\ref@ClauseN	680		
<b>T</b>			
\thecontractClause	12, 222		
\thecontractSubClause	12, 222		
\theH@AbsClause	389		
\theHClause	389		
\theHpar	567		
\theHsentence	807		
\theHSubClause	389		
\thepar	12, 567		
\thesentence	12, 807		
\thisparnumber	620		

title (Option) .....	358
tocentry (Option) .....	358
\toclevel@cpar .....	15

## U

Umgebungen:	
contract .....	222

## Z

Zähler:	
@AbsClause .....	389
contractAbsoluteClause ..	222
contractClause .....	222
contractSubClause .....	222
juratoclevel .....	15
par .....	567
sentence .....	807

## Change History

1.0		\protected@edef	durch	
	\DeprecatedParagraph: wird	\edef	ersetzt .....	37
	noch ein paar Versionen einen	v0.5d		
	Fehler melden .....	\contract: \jura@env@type		
	17	wird gesetzt .....		18
	\Paragraph: wird noch ein paar	fehlende Fehlermeldung er-		
	Versionen einen Fehler mel-	gänzt .....		18
	den .....	Umgebung definiert sich nicht		
	17	mehr selbst um .....		18
	\ParagraphCompatibilityHacks:	ref: neue Einstellungen		
	wird noch ein paar Versionen	nopar, nosentence,		
	verbleiben .....	OnlyParagraph .....		15
	17	\p@cpar: \jura@usetype mit Ar-		
	\refParagraph: wird noch ein	gument \jura@env@type ein-		
	paar Versionen einen Fehler	gefügt .....		28
	melden .....	\parciteformat: Argument ver-		
	17	schoben .....		35
	\refParagraphN: wird noch ein	\scrjura@@usetype: neu (in-		
	paar Versionen einen Fehler	tern) .....		28
	melden .....	\scrjura@env@type: neu (in-		
	17	tern) .....		18
	\SubParagraph: wird noch ein	\scrjura@usetype: neu (in-		
	paar Versionen einen Fehler	tern) .....		28
	melden .....	\sentenceciteformat: Argu-		
	17	ment verschoben .....		36
v0.5b		v0.5e		
	juratitlepagebreak: neue Op-	\Clausemark: Beachtung von		
	tion .....	\MakeMarkcase .....		14
	14	v0.6		
v0.5c		parnumber: neue Option .....		14
	\contract@everypar: \contract@Clauseformat	\cpar@indent: neu (intern) ..		14
	expandiert beim Schreiben zu	\cpar@numberwidth: neu (in-		
	seinem Argument .....	tern) .....		14
	29	\getmaxpar: Argument 3 muss		
	erstes Argument von			
	\newmaxpar und zweites Ar-			
	gument von \getmaxpar			
	ist contract nicht			
	contractpars .....			
	29			
	\getmaxpar: \#2@Clauseformat			
	expandiert zu seinem Argu-			
	ment .....			
	37			

mit <code>\protected@edef</code> expan-	Satznummer bei manuell num-
diert werden . . . . .	merierten Absätzen auf 0
37	statt 1 initialisiert, weil
<code>\if@juratoc</code> : Schalter durch	<code>\Sentence</code> die Nummer als
Zähler überflüssig . . . . .	erstes hochzählt. . . . .
13	29
<code>\ifparnumber</code> : Definition des	Zähler <code>contractAbsoluteClause</code>
Schalters in die Definition der	wird verwendet . . . . .
Option verschoben . . . . .	29
28	<code>\contract@paragraph</code> : Abstand
<code>entry</code> : Option veraltet . . . . .	nach der Nummer als Be-
22	standteil der Nummer setzen
<code>tocentry</code> : neue Option . . . . .	24
22	Zähler <code>contractAbsoluteClause</code>
<code>noentry</code> : Option veraltet . . . . .	wird erhöht . . . . .
22	23
<code>notocentry</code> : neue Option . . . . .	<code>\contract@ subparagraph</code> : Ab-
22	stand nach der Nummer als
<code>juratocindent</code> : neu . . . . .	Bestandteil der Nummer set-
14	zen . . . . .
<code>juratoclevel</code> : Zählerdefinition	25
verschoben . . . . .	Zähler <code>contractAbsoluteClause</code>
13	wird erhöht . . . . .
<code>juratocnumberwidth</code> : neu . . .	25
14	<code>ref</code> : Verwendung der reno-
v0.6a	vierten Schnittstelle mit
<code>parnumber</code> : Meldung bei falschen	<code>\FamilyKeyStateProcessed</code>
Werten korrigiert . . . . .	. . . . .
14	15
<code>\contract@paragraph</code> : fehlen-	<code>\ellipsispar</code> : neu . . . . .
des <code>\par</code> ergänzt . . . . .	18
24	<code>contractAbsoluteClause</code> : neu-
<code>\contract@ subparagraph</code> : feh-	er (interner) Zähler über al-
lendes <code>\par</code> ergänzt . . . . .	le Paragraphen, um ggf. die
25	Zählung neu beginnen zu
v0.6b	können . . . . .
<code>\Clauseformat</code> : <code>\paragraphformat</code>	18
umdefiniert . . . . .	<code>\parellipsis</code> : neu . . . . .
18	18
<code>\contract@paragraph</code> : Verwen-	<code>\parformat</code> : neues Font-
dung von <code>\contract@Clauseformat</code>	Element <code>parnumber</code> . . . . .
auch bei manueller Numme-	28
rierung der Paragraphen. . .	<code>\scrjura@lang@error</code> : Verwen-
24	dung von <code>\PackageError</code> an
<code>\newmaxpar</code> : <code>\#1@Clauseformat</code>	Stelle von <code>\PackageErrorNoLine</code>
expandiert zu seinem Argu-	. . . . .
ment . . . . .	36
37	<code>\scrjura@special@par</code> : neue
v0.7	Anweisung (intern) . . . . .
<code>paragraphmark</code> : Verwen-	29
dung der renovier-	<code>\scrjura@special@reset@par</code> :
ten Schnittstelle mit	neue Anweisung (intern) . .
<code>\FamilyKeyStateProcessed</code>	29
. . . . .	<code>\thisparnumber</code> : neue Anwei-
14	sung . . . . .
<code>parnumber</code> : Verwendung der re-	29
novierten Schnittstelle mit	<code>\toclevel@cpair</code> : neue für hyper-
<code>\FamilyKeyStateProcessed</code>	ref . . . . .
. . . . .	13
14	<code>juratocindent</code> : Verwen-
<code>\contract@everypar</code> : Absatz-	dung der renovier-
nummer von <code>\thisparnumber</code>	ten Schnittstelle mit
beachtet . . . . .	<code>\FamilyKeyStateProcessed</code>
29	. . . . .
Im Fall von nicht rein numme-	14
rischen Absatznummern die	<code>juratoclevel</code> : Verwendung der
Absätze auf jeden Fall num-	
merieren . . . . .	
29	

renovierten Schnittstelle mit \FamilyKeyState . . . . .	13	\ellipsispar: \thecontractAbsoluteClause flexibilisiert . . . . .	18
juratocnumberwidth: Ver- wendung der renovier- ten Schnittstelle mit \FamilyKeyStateProcessed . . . . .	14	\getmaxpar: Notfallcode in die aux-Datei . . . . .	37
v0.7a		\scrjura@usetype: \jura@usetype umbenannt . . . . .	28
\contract@sentence: \nobreak\hskip\z@ eingefügt, um die Trennung des ersten Worts nach der Satzmarkierung zu ermögli- chen . . . . .	35	\scrjura@afterheading: \jura@afterheading umbe- nannt . . . . .	26
auf \textsuperscript umge- stellt . . . . .	35	benötigt kein Argument mehr	26
v0.7b		\scrjura@env@type: \jura@env@type umbenannt . . . . .	18
General: Generelle Umbenen- nung von „Paragraph“ in „Clause“ für alle Befehle, Zähler und Optionen . . . .	12	\scrjura@everypar: \jura@everypar umbenannt . . . . .	27
v0.9		\scrjura@separator: \jura@separator umbenannt . . . . .	28
\contract@everypar: contractAbsoluteClause flexibilisiert . . . . .	29	\scrjura@usetype: \jura@usetype umbenannt . . . . .	28
\contract@paragraph: contractAbsoluteClause flexibilisiert . . . . .	23	\def\juraenvironment: Neue (interne) Anweisung . . . .	20
		\classNewJuraEnvironment: Neue Anweisung . . . . .	19