

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

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

2015/05/01

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 v0.9b 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` 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 L<sup>A</sup>T<sub>E</sub>X 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 erkaufte 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 Kolumnentitel

Bei langen Verträgen können automatische Kolumnentitel hilfreich sein. Allerdings hängt die Ausgabe hier davon ab, wie die diese erzeugt werden. `scrjura` unterstützt lebende Kolumnentitel. 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>	Kolumnentitel 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` 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>	
<code>\refSentence</code>	So etwas wie <code>\ref</code> nur für den Satz.
<code>sentence</code>	Zähler für die Satznummer und dessen Ausgabe, sowie Formatierung der
<code>\thesentence</code>	Referenz. ACHTUNG: <code>\thesentence</code> muss bei unverändertem <code>\sentenceciteformat</code> eine arabische Zahl sein!
<code>\sentenceciteformat</code>	
<code>\newmaxpar</code>	Werden intern verwendet und müssen deshalb allenfalls für Paketautoren dokumentiert werden.
<code>\getmaxpar</code>	

## 3 Bekannte Probleme

Die Verweisteknik 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{tocbasic}
```

`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.

`juratoclevel`

`\toclevel@cpar`

```
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     \fi
```

```
28   \endgroup
```

```
29   \FamilyKeyStateProcessed
```

```
30   \KOMA@kav@xreplacevalue{.scrjura.sty}{juratotoc}{\value{juratoclevel}}%
```

```
31 }
```

```
32 \KOMA@kav@xadd{.scrjura.sty}{juratotoc}{\value{juratoclevel}}%
```

```
33 \providecommand*\toclevel@cpar{\arabic{juratoclevel}}
```

juratocnumberwidth	Der Einzug und die Breite der Nummer für Inhaltsverzeichniseinträge kann
juratocindent	über diese beiden Optionen bzw. Makros bestimmt werden.
\cpar@numberwidth	34 \newcommand*{\cpar@numberwidth}{2em}
\cpar@indent	35 \newcommand*{\cpar@indent}{1.5em}
	36 \KOMA@key{juratocnumberwidth}{%
	37 \begingroup\setlength{\@tempdima}{#1}\endgroup
	38 \renewcommand*{\cpar@numberwidth}{#1}%
	39 \FamilyKeyStateProcessed
	40 \KOMA@kav@replacevalue{.scrjura.sty}{\cpar@numberwidth}{%
	41 }
	42 \KOMA@kav@add{.scrjura.sty}{\cpar@numberwidth}{\cpar@numberwidth}
	43 \KOMA@key{juratocindent}{%
	44 \begingroup\setlength{\@tempdima}{#1}\endgroup
	45 \renewcommand*{\cpar@indent}{#1}%
	46 \FamilyKeyStateProcessed
	47 \KOMA@kav@replacevalue{.scrjura.sty}{\cpar@indent}{\cpar@indent}%
	48 }
	49 \KOMA@kav@add{.scrjura.sty}{\cpar@indent}{\cpar@indent}%
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.
	50 \KOMA@ifkey{juratitlepagebreak}{@juratitlepagebreak}
parnumber	Die (automatische) Absatznummerierung lässt sich nun auch per Option ein- und ausschalten.
	51 \newif\ifparnumber
	52 \KOMA@key{parnumber}[true]{%
	53 \ifstr{#1}{auto}{%
	54 \AutoPar
	55 \FamilyKeyStateProcessed
	56 \KOMA@kav@remove{.scrjura.sty}{parnumber}{manual}%
	57 \KOMA@kav@remove{.scrjura.sty}{parnumber}{auto}%
	58 \KOMA@kav@add{.scrjura.sty}{parnumber}{auto}%
	59 }{%
	60 \ifstr{#1}{manual}{%
	61 \ManualPar
	62 \FamilyKeyStateProcessed
	63 \KOMA@kav@remove{.scrjura.sty}{parnumber}{manual}%
	64 \KOMA@kav@remove{.scrjura.sty}{parnumber}{auto}%
	65 \KOMA@kav@add{.scrjura.sty}{parnumber}{manual}%
	66 }{%
	67 \KOMA@set@ifkey{parnumber}{parnumber}{#1}%

```

68      \KOMA@kav@replacebool{.scrjura.sty}{parnumber}{parnumber}%
69    }%
70  }%
71 }
72 \KOMA@kav@add{.scrjura.sty}{parnumber}{true}
73 \KOMA@kav@add{.scrjura.sty}{parnumber}{auto}

```

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!)

```

74 \newcommand*{\Clausemark}[1]{%
75 \KOMA@key{paragraphmark}{%
76   \begingroup
77     \KOMA@set@ncmdkey{paragraphmark}{@tempa}{%
78       {false}{0},{off}{0},{no}{0},%
79       {forceright}{1},%
80       {forceboth}{2},%
81       {right}{3},%
82       {both}{4}%
83     }{#1}%
84   \ifx\FamilyKeyState\FamilyKeyStateProcessed
85     \ifcase\number\@tempa
86       \endgroup
87       \let\Clausemark\@gobble
88     \or
89       \endgroup
90       \renewcommand*{\Clausemark}[1]{%
91         \markright{\csname MakeMarkcase\endcsname{##1}}}%
92     \or
93       \endgroup
94       \renewcommand*{\Clausemark}[1]{%
95         \markboth{\csname MakeMarkcase\endcsname{##1}}%
96           {\csname MakeMarkcase\endcsname{##1}}}%
97     \or
98       \endgroup
99       \renewcommand*{\Clausemark}[1]{%
100         \ifx
101           \@mkboth\@gobbletwo
102         \else
103           \markright{\csname MakeMarkcase\endcsname{##1}}%
104           \fi}%
105     \or
106       \endgroup
107       \renewcommand*{\Clausemark}[1]{%
108         \@mkboth{\csname MakeMarkcase\endcsname{##1}}%
109           {\csname MakeMarkcase\endcsname{##1}}}%
110     \else

```

```

111     \endgroup
112   \fi
113   \FamilyKeyStateProcessed
114 \else
115   \endgroup
116   \FamilyKeyStateUnknownValue
117 \fi
118 \KOMAC@kav@xreplacevalue{.scrjura.sty}{paragraphmark}{#1}%
119 }
120 \KOMAC@kav@add{.scrjura.sty}{paragraphmark}{false}
121 \KOMAC@DeclareDeprecatedOption[scrjura]{markright}{paragraphmark=forceright}
122 \KOMAC@DeclareDeprecatedOption[scrjura]{markboth}{paragraphmark=forceboth}

```

ref Die Form, in der Absätze und Sätze referenziert werden. Es gibt eine lange,  
 parccitename eine abgekürzte und eine numerische Form.  
 sentencecitename  
 \parccite@fromat In der Voreinstellung wird das lange Format verwendet. Es gilt: 0 = lang,  
 \sentencecite@fromat 1 = kurz, 2 = numerisch, -1 = gar nicht

```

123 \newcommand*{\parccite@format}{0}
124 \newcommand*{\sentencecite@format}{0}

```

Per Option kann die Voreinstellung verändert werden.

```

125 \KOMAC@key{ref}{%
126   \beginngroup
127     \KOMAC@set@ncmdkey{ref}{@tempa}{%
128       {parlong}{1},{longpar}{1},{ParL}{1},%
129       {parshort}{2},{shortpar}{2},{ParS}{2},%
130       {parnumeric}{3},{numericpar}{3},{ParN}{3},%
131       {paroff}{4},{nopar}{4},%
132       {sentencelong}{10},{longsentence}{10},{SentenceL}{10},%
133       {sentenceshort}{20},{shortsentence}{20},{SentenceS}{20},%
134       {sentencenumeric}{30},{numeric sentence}{30},{SentenceN}{30},%
135       {sentenceoff}{40},{nosentence}{40},%
136       {long}{11},%
137       {short}{22},%
138       {numeric}{33},%
139       {paragraphonly}{44},{onlyparagraph}{44},%
140       {ParagraphOnly}{44},{OnlyParagraph}{44}%
141     }{#1}%
142   \ifx\FamilyKeyState\FamilyKeyStateProcessed
143     \aftergroup\FamilyKeyStateProcessed
144     \@tempcnta=\@tempa\relax
145     \@tempcntb=\z@
146     \@whilenum \@tempcnta>9 \do{%
147       \advance\@tempcnta -10\relax
148       \advance\@tempcntb \@ne\relax

```



```

149     }%
150     \ifcase \@tempcnta
151     \or
152         \aftergroup\def\aftergroup\parcite@format
153         \aftergroup{\aftergroup0\aftergroup}%
154     \or
155         \aftergroup\def\aftergroup\parcite@format
156         \aftergroup{\aftergroup1\aftergroup}%
157     \or
158         \aftergroup\def\aftergroup\parcite@format
159         \aftergroup{\aftergroup2\aftergroup}%
160     \or
161         \aftergroup\def\aftergroup\parcite@format
162         \aftergroup{\aftergroup-\aftergroup1\aftergroup}%
163     \fi
164     \ifcase \@tempcntb
165     \or
166         \aftergroup\def\aftergroup\sentencecite@format
167         \aftergroup{\aftergroup0\aftergroup}%
168     \or
169         \aftergroup\def\aftergroup\sentencecite@format
170         \aftergroup{\aftergroup1\aftergroup}%
171     \or
172         \aftergroup\def\aftergroup\sentencecite@format
173         \aftergroup{\aftergroup2\aftergroup}%
174     \or
175         \aftergroup\def\aftergroup\sentencecite@format
176         \aftergroup{\aftergroup-\aftergroup1\aftergroup}%
177     \fi
178     \else
179         \aftergroup\FamilyKeyStateUnknownValue
180     \fi
181 \endgroup
182 \ifx\FamilyKeyState\FamilyKeyStateProcessed
183     \KOMA@kav@removekey{.scrjura.sty}{ref}%
184     \ifcase\parcite@format
185         \KOMA@kav@add{.scrjura.sty}{ref}{parlong}%
186     \or
187         \KOMA@kav@add{.scrjura.sty}{ref}{parshort}%
188     \or
189         \KOMA@kav@add{.scrjura.sty}{ref}{parnumeric}%
190     \or
191         \KOMA@kav@add{.scrjura.sty}{ref}{paroff}%
192     \fi
193     \ifcase\sentencecite@format
194         \KOMA@kav@add{.scrjura.sty}{ref}{sentencelong}%
195     \or
196         \KOMA@kav@add{.scrjura.sty}{ref}{sentenceshort}%
197     \or

```

```

198 \KOMA@kav@add{.scrjura.sty}{ref}{sentencenumeric}%
199 \or
200 \KOMA@kav@add{.scrjura.sty}{ref}{sentenceoff}%
201 \fi
202 \fi
203 }
204 \KOMA@kav@add{.scrjura.sty}{ref}{parlong}%
205 \KOMA@kav@add{.scrjura.sty}{ref}{sentencelong}%
206 \KOMA@DeclareDeprecatedOption[scrjura]{parcite}{ref=parlong}
207 \KOMA@DeclareDeprecatedOption[scrjura]{sentencecite}{ref=sentencelong}

```

Optionen ausführen.

```

208 \KOMAProcessOptions\relax

```

```

\Paragraph
\SubParagraph
\refParagraph
\refParagraphN
\DeprecatedParagraph
\ParagraphCompatibilityHacks
209 \providecommand*{\DeprecatedParagraph}{%
210 \PackageError{scrjura}{modification of old document needed}{%
211 It seem that this document was made for scrjura up to version
212 0.7a.\MessageBreak
213 Since scrjura version 0.9 \string\Paragraph, \string\SubParagraph, and all
214 depending\MessageBreak
215 commands, options, and counters have been renamed.\MessageBreak
216 You should replace the terms ‘Paragraph’ and ‘paragraph’ by ‘Clause’
217 and\MessageBreak
218 ‘clause’ if they are part of the name of a scrjura feature, otherwise
219 this\MessageBreak
220 document may produce severall additional error messages and maybe the
221 wrong\MessageBreak
222 result. Sorry for the inconvenience.%
223 }%
224 \ParagraphCompatibilityHacks
225 }
226 \newcommand*{\ParagraphCompatibilityHacks}{%
227 \PackageWarning{scrjura}{compatibility hacks for ‘\string\Paragraph’
228 executed.\MessageBreak
229 There is no support for documents using these hacks!\MessageBreak
230 There is no warranty for real compatibility!\MessageBreak
231 Even if the LaTeX run of the document doesn’t report\MessageBreak
232 any error, the result may be completely wrong.\MessageBreak
233 Therefore it is recommended to solve the problem,\MessageBreak
234 instead of trying to work around using the\MessageBreak
235 compatibility hacks%
236 }%
237 \gdef\Paragraph{\Clause}%
238 \gdef\SubParagraph{\SubClause}%
239 \gdef\c@Paragraph{\c@Clause}%
240 \gdef\cl@Paragraph{\cl@Clause}%

```

```

241 \gdef\c@SubParagraph{\c@SubClause}%
242 \gdef\cl@SubParagraph{\cl@SubClause}%
243 \gdef\theParagraph{\theClause}%
244 \gdef\theSubParagraph{\theSubClause}%
245 \gdef\refParagraph{\refClause}%
246 \gdef\refParagraphN{\refClauseN}%
247 \aliaskomafont{Paragraph}{Clause}%
248 \scr@ifundefinedorrelax{Paragraphmark}{-}{%
249   \global\let\Clausemark\Paragraphmark
250 }%
251 }
252 \providecommand*\{Paragraph}{\DeprecatedParagraph\Paragraph}
253 \providecommand*\{SubParagraph}{\DeprecatedParagraph\SubParagraph}
254 \providecommand*\{refParagraph}{\DeprecatedParagraph\refParagraph}
255 \providecommand*\{refParagraphN}{\DeprecatedParagraph\refParagraphN}

```

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

```

256 \newcommand*\{scrjura@env@type}{-}

```

`\ellipsispar` Paragraphen auslassen aber mit zählen.

`\parellipsis`

```

257 \newcommand*\{ellipsispar}[1][1]{%
258   \begingroup
259     \KOMAOptions{parnumber=manual}\parellipsis\par
260     \addtocounter{par}{#1}%
261     \protected@write\@auxout{}{%
262       \string\newmaxpar{\scrjura@env@type}%
263       {\csname the\scrjura@env@type AbsoluteClause\endcsname}%
264       {\thepar}%
265     }%
266   \endgroup
267   \addtocounter{par}{-1}\refstepcounter{par}%
268   \ignorespaces
269 }
270 \newcommand*\{parellipsis}{%
271   \scr@ifundefinedorrelax{textellipsis}{\dots}{\textellipsis}%
272 }

```

`contract` Die Umgebung `contract` wird als Anweisung `\contract` definiert. Nichts

`\contract` desto trotz handelt es sich dabei um eine Umgebung und sollte sie immer

`contractClause` als Umgebung verwendet werden! Die Umgebung kann nicht geschachtelt

`\thecontractClause` werden. Sie darf aber enden und dann erneut verwendet werden. Die Pa-

`\contract@Clauseformat` ragraphen werden jedoch über alle Verträge hinweg nummeriert. Das Ende

`\Clauseformat` eines Vertrags ist also genau genommen nur eine Unterbrechung.

`contractSubClause`

`\thecontractSubClause`

`contractAbsoluteClause`

```

273 \newcommand*\{contract}{-}

```

```

274 \ifx\scrjura@env@type\@empty
275   \let\@doendpe\scrjura@doendpe
276   \let\Clause\contract@paragraph
277   \let\c@Clause\c@contractClause
278   \edef\cl@Clause{\cl@Clause\cl@contractClause}%
279   \let\SubClause\contract@subparagraph
280   \let\c@SubClause\c@contractSubClause
281   \edef\cl@SubClause{\cl@SubClause\cl@contractSubClause}%
282   \let\Sentence\contract@sentence
283   \renewcommand*{\scrjura@env@type}{contract}%
284   \aliaskomafont{Clause}{contract.Clause}%
285 \else
286   \PackageError{scrjura}{nested 'contract' detected}{%
287     You may not use a 'contract' environment inside\MessageBreak
288     a '\scrjura@env@type' environment or after loading\MessageBreak
289     package 'scrjura' with option '\scrjura@env@type'!}%
290 \fi
291 }
292 \let\if@scrjura@skiphyperrref\iftrue
293 \let\cl@Clause\@empty
294 \let\cl@SubClause\@empty
295 \newcounter{contractClause}
296 \renewcommand*{\thecontractClause}{%
297   {\contract@Clauseformat{\arabic{Clause}}}}
298 \DeclareRobustCommand*{\contract@Clauseformat}[1]{\Clauseformat{#1}}
299 \newcommand*{\Clauseformat}[1]{\S~#1}
300 \newcounter{contractSubClause}
301 \@addtoreset{SubClause}{Clause}
302 \renewcommand*{\thecontractSubClause}{%
303   {\theClause\alph{SubClause}}}
304 \newcounter{contractAbsoluteClause}

```

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

```

305 \newcommand*{\DeclareNewJuraEnvironment}[1]{%
306   \@ifundefined{#1}{\expandafter\let\csname #1\expandafter\endcsname
307     \csname end#1\endcsname}{}%
308   \@ifundefined{#1}{\let\reserved@defjuraenvironment\@defjuraenvironment}{%
309     \PackageError{scrjura}{ignoring declaration of '#1'}{%
310       You've tried to declare jura environment '#1', but
311       environment\MessageBreak
312       '#1' or command
313       \expandafter\string\csname #1\endcsname\space or
314       \expandafter\string\csname end#1\endcsname\MessageBreak
315       already exists.\MessageBreak
316       Declaration will be ignored}%
317   \long\def\reserved@defjuraenvironment##1[##2]##3##4{%
318     }%

```

```

319 \@ifnextchar [%]
320   {\reserved@defjuraenvironment{#1}}{\reserved@defjuraenvironment{#1}[]}%
321 }
322 \@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!

```

323 \DefineFamily{KOMAAarg}
324 \DefineFamilyMember{KOMAAarg}
325 \DefineFamilyKey{KOMAAarg}{Clause}{%
326   \expandafter\gdef\csname \scrjura@env@type @Clause\endcsname{#1}%
327 }
328 \DefineFamilyKey{KOMAAarg}{SubClause}{%
329   \expandafter\gdef\csname \scrjura@env@type @SubClause\endcsname{#1}%
330 }
331 \DefineFamilyKey{KOMAAarg}{Sentence}{%
332   \expandafter\gdef\csname \scrjura@env@type @Sentence\endcsname{#1}%
333 }
334 \DefineFamilyKey{KOMAAarg}{ClauseNumberFormat}{%
335   \expandafter\gdef\csname \scrjura@env@type @Clauseformat \endcsname
336   ##1{#1{##1}}
337   \expandafter\xdef\csname \scrjura@env@type @Clauseformat\endcsname{%
338     \noexpand\protect\expandafter\noexpand
339     \csname \scrjura@env@type @Clauseformat \endcsname
340   }%
341 }
342 \newcommand{\@defjuraenvironment}{%
343 \long\def\@defjuraenvironment#1[#2]#3#4{%

```

Die Zähler definieren:

```

344 \newcounter{#1Clause}%
345 \newcounter{#1AbsoluteClause}%
346 \newcounter{#1SubClause}%
347 \begingroup
348   \edef\scrjura@env@type{#1}%
349   \FamilyExecuteOptions[.scrjura.sty]{KOMAAarg}{#2}%
350 \endgroup
351 \@ifundefined{#1@Clauseformat}{%
352   \expandafter\DeclareRobustCommand\expandafter*%
353   \csname #1@Clauseformat\endcsname[1]{\@Clauseformat{##1}}%
354 }{}%

```

Umgebung:

```

355 \newenvironment{#1}{%

```

```

356 \par
357 \ifx\scrjura@env@type\@empty
358 \edef\scrjura@env@type{#1}%
359 \let\@doendpe\scrjura@doendpe
360 \expandafter\let\expandafter\c@Clause\csname c@#1Clause\endcsname
361 \edef\cl@Clause{\cl@Clause\csname cl@#1Clause\endcsname}%
362 \expandafter\let\expandafter\c@SubClause
363 \csname c@#1SubClause\endcsname
364 \edef\cl@SubClause{\cl@SubClause
365 \csname cl@#1SubClause\endcsname}%
366 \@ifundefined{#1@Clause}{%
367 \let\Clause\contract@paragraph
368 }{%
369 \expandafter\let\expandafter\Clause
370 \csname #1@Clause\endcsname
371 }%
372 \@ifundefined{#1@SubClause}{%
373 \let\SubClause\contract@subparagraph
374 }{%
375 \expandafter\let\expandafter\SubClause
376 \csname #1@SubClause\endcsname
377 }%
378 \@ifundefined{#1@Sentence}{%
379 \let\Sentence\contract@sentence
380 }{%
381 \expandafter\let\expandafter\Sentence\csname #1@Sentence\endcsname
382 }%
383 \@ifundefined{\scrjura@env@type @everypar}{%
384 \expandafter\let
385 \csname \scrjura@env@type @everypar\endcsname
386 \contract@everypar
387 }{%
388 \@ifundefined{scr@fnt@#1}{%
389 \@ifundefined{scr@fnt@instead@#1}{%
390 \aliaskomafont{Clause}{contract.Clause}%
391 }{%
392 \aliaskomafont{Clause}{\csname scr@fnt@instead@#1\endcsname}%
393 }%
394 }{%
395 \aliaskomafont{Clause}{contract.Clause}%
396 }%
397 #3%
398 \else
399 \PackageError{scrjura}{nested scrjura environments detected}{%
400 You must not use a ‘#1’ environment inside\MessageBreak
401 a ‘\scrjura@env@type’ environment or after loading\MessageBreak
402 package ‘scrjura’ with option ‘\scrjura@env@type’!}%
403 \fi
404 }{%

```

```

405     #4%
406     \par
407 }%
408 }

```

`\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
notocentry
409 \define@key{contract}{title}{%
410 \def\contract@title{#1}%
411 \ifx\contract@entry\relax\def\contract@entry{\contract@title}\fi
412 \ifx\contract@head\relax\def\contract@head{\contract@title}\fi
413 }
414 \define@key{contract}{entry}{%
415 \PackageWarning{scrjura}{deprecated option 'entry'.\MessageBreak
416 You should use option 'tocentry' instead of\MessageBreak
417 option 'entry'%
418 }%
419 \def\contract@entry{#1}}
420 \define@key{contract}{tocentry}{\def\contract@entry{#1}}
421 \define@key{contract}{noentry}[]{%
422 \PackageWarning{scrjura}{deprecated option 'noentry'.\MessageBreak
423 You should use option 'notocentry' instead of\MessageBreak
424 option 'noentry'%
425 }%
426 \let\contract@entry\relax}
427 \define@key{contract}{notocentry}[]{\let\contract@entry\relax}
428 \define@key{contract}{head}{\def\contract@head{#1}}
429 \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.

```

430 \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`

```

431 \newcommand*{\contract@preskip}{2\baselineskip}
432 \newcommand*{\contract@postskip}{\baselineskip}
433 \define@key{contract}{preskip}{\def\contract@preskip{#1}}
434 \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.

```

435 \newif\ifcontract@dummy
436 \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!

```

437 \newkomafont{contract.Clause}{\sffamily\bfseries\large}
438 \newcommand*{\contract@paragraph@font}{\usekomafont{contract.Clause}%
439 \@hangfrom}

```

```

@AbsClause
\theH@AbsClause
\theHClause
\theHSubClause
440 % \textsf{hyperref}-Code, der noch nicht getestet ist!
441 \newcounter{@AbsClause}
442 \newcommand*{\theH@AbsClause}{P-\arabic{@AbsClause}}
443 \newcommand*{\theHClause}{\theH@AbsClause}
444 \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.

```

445 \newcommand*{\contract@paragraph}[1]{%
446 \stepcounter{\scrjura@env@type AbsoluteClause}%
447 \ManualPar\parnumbertrue
448 \let\contract@title\relax
449 \let\contract@entry\relax
450 \let\contract@head\relax
451 \let\contract@number\relax
452 \contract@dummyfalse
453 \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.

```

454 \ifcontract@dummy\else
455   \par
456   \@afterindentfalse
457   \addvspace{\contract@preskip}%
458 \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.

```

459 \ifx\contract@number\relax
460   \let\p@Clause\@empty
461   \let\theClause\thecontractClause
462   \refstepcounter{Clause}%
463 \else
464   \begingroup
465     \let\@elt\@stpelt
466     \cl@Clause
467   \endgroup
468   \protected@edef\theClause{%
469     \csname \scrjura@env@type @Clauseformat\endcsname{\contract@number}%
470   }%
471   \protected@edef\@currentlabel{\theClause}%
472 \fi
473 \stepcounter{@AbsClause}%
474 \begingroup\expandafter\expandafter\expandafter\endgroup
475 \expandafter\ifx\csname if@skiphyperref\endcsname\relax
476 \else
477   \expandafter\let\csname if@scrjura@skiphyperref\expandafter\endcsname
478   \csname if@skiphyperref\endcsname
479 \fi
480 \if@scrjura@skiphyperref\else
481   \hyper@refstepcounter{@AbsClause}%
482   \typeout{absolute Nummer: \the@AbsClause^^JLabel: '\@currentHref'}%
483 \fi

```

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

```

484 \let\theSubClause\theClause

```

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

```

485 \ifcontract@dummy\else
486   \begingroup
487     \if@juratitlepagebreak\else\interlinepenalty\@M\fi

```

```

488     \contract@paragraph@font{\theClause
489     \ifx\contract@title\relax\else\enskip\fi}%
490     \contract@title
491     \ifx\contract@entry\relax\else
492     \addxcontentsline{toc}{cpar}[\theClause]\contract@entry
493     \addxcontentsline{cpa}{cpar}[\theClause]\contract@entry
494     \fi
495     \ifx\contract@head\relax\else
496     \expandafter\Clausemark\expandafter{%
497     \expandafter\theSubClause\expandafter\enskip\contract@head}%
498     \fi
499     \par
500 \endgroup\nobreak\vskip\contract@postskip

```

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

```

501     \scrjura@afterheading
502     \fi
503 }

```

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

```

504 \newcommand*{\contract@subparagraph}[1]{%
505   \stepcounter{\scrjura@env@type AbsoluteClause}%
506   \ManualPar\parnumbertrue
507   \let\contract@title\relax
508   \let\contract@entry\relax
509   \let\contract@head\relax
510   \let\contract@number\relax
511   \contract@dummysfalse
512   \ifx\relax#1\relax\else\setkeys{contract}{#1}\fi
513   \ifcontract@dummys\else
514     \par
515     \@afterindentfalse
516     \vskip\contract@preskip
517   \fi
518   \ifx\contract@number\relax
519     \let\p@SubClause\@empty
520     \let\theSubClause\thecontractSubClause
521     \refstepcounter{SubClause}%
522   \else
523     \begingroup
524     \let\@elt\@stpelt
525     \cl@SubClause
526   \endgroup
527   \protected@edef\theSubClause{\theClause\contract@number}%
528   \protected@edef\@currentlabel{\theSubClause}%

```

```

529 \fi
530 \stepcounter{@AbsClause}%
531 \begingroup\expandafter\expandafter\expandafter\endgroup
532 \expandafter\ifx\csname if@skiphyperref\endcsname\relax
533 \else
534   \expandafter\let\csname if@scrjura@skiphyperref\expandafter\endcsname
535   \csname if@skiphyperref\endcsname
536 \fi
537 \if@scrjura@skiphyperref\else
538   \hyper@refstepcounter{@AbsClause}%
539   \typeout{absolute Nummer: \the@AbsClause^^JLabel: '\@currentHref'}%
540 \fi
541 \ifcontract@dummys\else
542   \begingroup
543     \if@juratitlepagebreak\else\interlinepenalty\@M\fi
544     \contract@paragraph@font{\theSubClause
545       \ifx\contract@title\relax\else\enskip\fi}%
546     \contract@title
547     \ifx\contract@entry\relax\else
548       \addxcontentsline{toc}{cpar}{[\theSubClause]\contract@entry
549       \addxcontentsline{cpa}{cpar}{[\theSubClause]\contract@entry
550     \fi
551     \ifx\contract@head\relax\else
552       \expandafter\Clausemark\expandafter{%
553         \expandafter\theSubClause\expandafter\enskip\contract@head}%
554     \fi
555     \par
556   \endgroup
557   \nobreak\vskip\contract@postskip
558   \scrjura@afterheading
559 \fi
560 }

```

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

```

561 \newcommand*{\AutoPar}{%
562   \expandafter\let\expandafter\scrjura@everypar
563   \csname \scrjura@env@type @everypar\endcsname
564 }
565 \newcommand*{\ManualPar}{%
566   \let\scrjura@everypar\relax
567 }

```

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

```

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

```

```

569 \@nobreaktrue
570 \everypar{%
571   \if@nobreak
572     \@nobreakfalse
573     \clubpenalty \@M
574     \if@afterindent \else
575       {\setbox\z@\lastbox}%
576     \fi
577   \else
578     \clubpenalty \@clubpenalty
579     \everypar{%
580       \scrjura@everypar
581     }%
582   \fi
583   \scrjura@everypar
584 }%
585 \AutoPar
586 }

```

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

```
587 \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`

```

588 \CheckCommand*\@doendpe{\@endpetrue
589   \def\par{\@restorepar\everypar{}\par\@endpefalse}\everypar
590   {\setbox\z@\lastbox}\everypar{}\@endpefalse}}
591 \newcommand*{\scrjura@doendpe}{%
592   \@endpetrue
593   \def\par{%
594     \@restorepar\everypar{%
595       \csname scrjura@everypar\endcsname
596     }%
597     \par\@endpefalse
598   }%
599   \everypar{%
600     {\setbox\z@\lastbox}\everypar{%
601       \csname scrjura@everypar\endcsname
602     }%
603     \@endpefalse
604   }%

```

605 }

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

```

606 \newcommand*{\l@cpa}{2}{%
607   \ifnum\value{juratoclevel}>\value{tocdepth}\else
608     \scr@ifundefinedorrelax{bprot@dottedtocline}{%
609       \@dottedtocline
610     }{%
611       \bprot@dottedtocline
612     }{\value{juratoclevel}}{\cpar@indent}{\cpar@numberwidth}{#1}{#2}%
613   \fi
614 }

```

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

```

615 \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.

```

616 \DeclareRobustCommand*{\scrjura@usetype}[1]{\scrjura@@usetype{#1}}
617 \newcommand*{\scrjura@@usetype}[1]{}

```

`\contract@everpar`    `\scrjura@everpar` 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-

`\thepar`    sonsten werden die Absätze mit `\thepar` nummeriert. Wichtig ist, dass der

`\theHpar`    Absatzzähler mit den Paragraphen und den Unterparagraphen zurückge-

`\parformat`    setzt wird. Außerdem muss beim Zitieren als Elternobjekt der Paragraph

`\p@par`    mit ausgegeben werden.

```

618 \newcounter{par}
619 \renewcommand*{\thepar}{\arabic{par}}
620 \newcommand*{\theHpar}{\theH@AbsClause-\Roman{par}}
621 \newcommand*{\parformat}{(\thepar)}
622 \newkomafont{parnumber}{}
623 \renewcommand*\p@par{{\scrjura@usetype{\scrjura@env@type}\theSubClause\scrjura@s
624 \@addtoreset{par}{Clause}
625 \@addtoreset{par}{SubClause}

```

```

626 \newcommand*{\contract@everypar}{%
627   \ifparnumber
628     \ifx\scrjura@special@par\relax
629       \ifx\scrjura@special@reset@par\relax\else
630         \global\let\thepar\scrjura@special@reset@par
631         \global\let\scrjura@special@reset@par\relax
632       \fi
633       \refstepcounter{par}%
634       \refstepcounter{sentence}%
635     \else
636       \ifx\scrjura@special@reset@par\relax
637         \global\let\scrjura@special@reset@par\thepar
638       \fi
639       \global\let\thepar\scrjura@special@par
640       \global\let\scrjura@special@par\relax
641       \setcounter{sentence}{0}\refstepcounter{sentence}%
642     \fi
643   \begingroup
644     \protected@write\@auxout{%
645       \expandafter\let\csname \scrjura@env@type @Clauseformat\endcsname
646       \@firstofone
647     }{%
648       \string\newmaxpar{\scrjura@env@type}%
649       {\csname the\scrjura@env@type AbsoluteClause\endcsname}%
650       {\thepar}%
651     }%
652     \getmaxpar\@tempa{\scrjura@env@type}%
653     {\csname the\scrjura@env@type AbsoluteClause\endcsname}%
654     \typeout{Stored max is \@tempa}%
655     \def\reserved@a##1\@nnil{\def\@tempa{##1}}%
656     \afterassignment\reserved@a\@tempcnta=0\@tempa\relax\@nnil
657     \ifnum \@tempcnta>\@ne
658       {\usekomafont{parnumber}{\parformat\nobreakspace}}%
659     \else
660       \def\reserved@a{\relax}%
661       \ifx\@tempa\reserved@a
662       \else
663         {\usekomafont{parnumber}{\parformat\nobreakspace}}%
664       \fi
665     \fi
666   \endgroup
667 \else
668   \setcounter{sentence}{-1}\refstepcounter{sentence}%
669 \fi
670 }

```

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

```

671 \newcommand*{\thisparnumber}[1]{%
672   \def\scrjura@special@par{#1}%
673 }
674 \newcommand*{\scrjura@special@par}{}
675 \let\scrjura@special@par\relax
676 \newcommand*{\scrjura@special@reset@par}{}
677 \let\scrjura@special@reset@par\relax

```

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

```

\ref@L
678 \newcommand*{\refL}{\@ifstar {\ref@L*}{\ref@L{}}}
679 \newcommand*{\ref@L}[2]{%
680   \begingroup
681     \def\parcite@format{0}%
682     \let\sentencecite@format\parcite@format
683     \ref#1{#2}%
684   \endgroup
685 }

```

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

```

\ref@S
686 \newcommand*{\refS}{\@ifstar {\ref@S*}{\ref@S{}}}
687 \newcommand*{\ref@S}[2]{%
688   \begingroup
689     \def\parcite@format{1}%
690     \let\sentencecite@format\parcite@format
691     \ref#1{#2}%
692   \endgroup
693 }

```

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

```

\ref@N
694 \newcommand*{\refN}{\@ifstar {\ref@N*}{\ref@N{}}}
695 \newcommand*{\ref@N}[2]{%
696   \begingroup
697     \def\parcite@format{2}%
698     \let\sentencecite@format\parcite@format
699     \ref#1{#2}%
700   \endgroup
701 }

```

`\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.

```

702 \newcommand*{\refClause}{%
703   \@ifstar {\ref@Clause*}{\ref@Clause{}}
704 }

```

```

705 \newcommand*{\ref@Clause}[2]{%
706   \expandafter\ifx\csname r@#2\endcsname\relax
707     \ref#1{#2}%
708   \else
709     \begingroup

```

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

```

710     \expandafter\expandafter\expandafter\expandafter
711     \expandafter\expandafter\expandafter\def
712     \expandafter\expandafter\expandafter\expandafter
713     \expandafter\expandafter\expandafter\@tempb
714     \expandafter\expandafter\expandafter\expandafter
715     \expandafter\expandafter\expandafter{%
716     \expandafter\expandafter\expandafter\@gobble\csname r@#2\endcsname}%

```

Den ersten Teil der Referenz in \@tempa ablegen.

```

717     \def\@tempc##1##2\@nil{##1}%
718     \let\scrjura@separator\@gobble
719     \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
720     \csname r@#2\endcsname\noexpand\@nil}%

```

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

```

721     \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
722     \@tempa\@nil}%
723     \let\@protect\protect
724     \let\protect\noexpand
725     \expandafter\edef\csname r@#2\endcsname{{\@tempa}\@tempb}%
726     \let\protect\@protect
727     \ref#1{#2}%
728   \endgroup
729 \fi
730 }

```

\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.

```

731 \newcommand*{\refClauseN}{%
732   \@ifstar {\ref@ClauseN*}{\ref@ClauseN{}}
733 }
734 \newcommand*{\ref@ClauseN}[2]{%
735   \begingroup
736     \let\Clauseformat\relax
737     \ref@Clause{#1}{#2}%
738   \endgroup
739 }

```



`\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.

```

740 \newcommand*{\refPar}{%
741   \@ifstar {\ref@Par*}{\ref@Par{}}
742 }
743 \newcommand*{\ref@Par}[2]{%
744   \expandafter\ifx\csname r@#2\endcsname\relax
745     \ref#1{#2}%
746   \else
747     \begingroup

```

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

```

748     \expandafter\expandafter\expandafter\expandafter
749     \expandafter\expandafter\expandafter\def
750     \expandafter\expandafter\expandafter\expandafter
751     \expandafter\expandafter\expandafter\@tempb
752     \expandafter\expandafter\expandafter\expandafter
753     \expandafter\expandafter\expandafter{%
754     \expandafter\expandafter\expandafter\@gobble\csname r@#2\endcsname}%

```

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

```

755     \def\@tempc##1##2\@nil{##1}%
756     \let\scrjura@separator\@gobble
757     \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
758     \csname r@#2\endcsname\noexpand\@nil}%

```

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

```

759     \def\@tempc##1##2##3\@nil{##2}%
760     \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
761     \@tempa{%
762       \protect\G@refundefinedtrue
763       \nfss@text{\reset@font\bfseries ??}%
764       \@latex@warning{Reference ‘#2’ on page \thepage \space
765       with undefined par number}%
766     }\noexpand\@nil}%
767     \let\@@protect\protect
768     \let\protect\noexpand
769     \expandafter\edef\csname r@#2\endcsname{{\@tempa}\@tempb}%
770     \let\protect\@@protect
771     \ref#1{#2}%
772   \endgroup
773 \fi
774 }

```

```

\refParL
\ref@ParX
775 % Das Gleiche zwingend lang.
776 \newcommand*{\refParL}{%
777   \@ifstar {\ref@ParX0*}{\ref@ParX0{}}
778 }
779 \newcommand*{\ref@ParX}[3]{%
780   \begingroup
781     \def\parcite@format{#1}%
782     \let\sentencecite@format\parcite@format
783     \ref@Par{#2}{#3}%
784   \endgroup
785 }

```

```

\refParS  Das Gleiche zwingend kurz.

786 \newcommand*{\refParS}{%
787   \@ifstar {\ref@ParX1*}{\ref@ParX1{}}
788 }

```

```

\refParN  Das Gleiche zwingend numerisch.

789 \newcommand*{\refParN}{%
790   \@ifstar {\ref@ParN2*}{\ref@ParN2{}}
791 }
792 \newcommand*{\ref@ParN}[2]{%
793   \@ifnextchar [%]
794     {\ref@ParN{#1}{#2}}%
795     {\ref@ParX{#1}{#2}}%
796 }
797 \newcommand*{\ref@ParN}{%
798   \def\ref@ParN#1#2[#3]#4{%
799     \begingroup
800       \renewcommand*{\parnumericformat}[1]{%
801         \csname @#3\endcsname{\number ##1\relax}%
802       }%
803       \ref@ParX{#1}{#2}{#4}%
804     \endgroup
805 }

```

`\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.

```

806 \newcommand*{\refSentence}{%
807   \@ifstar {\ref@Sentence*}{\ref@Sentence{}}
808 }
809 \newcommand*{\ref@Sentence}[2]{%

```

```

810 \expandafter\ifx\csname r@#2\endcsname\relax
811   \ref#1{#2}%
812 \else
813   \begingroup

```

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

```

814   \expandafter\expandafter\expandafter\expandafter
815   \expandafter\expandafter\expandafter\def
816   \expandafter\expandafter\expandafter\expandafter
817   \expandafter\expandafter\expandafter\@tempb
818   \expandafter\expandafter\expandafter\expandafter
819   \expandafter\expandafter\expandafter{%
820   \expandafter\expandafter\expandafter\@gobble\csname r@#2\endcsname}%

```

Den ersten Teil der Referenz in \@tempa ablegen.

```

821   \def\@tempc##1##2\@nil{##1}%
822   \let\scrjura@separator\@gobble
823   \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
824   \csname r@#2\endcsname\noexpand\@nil}%

```

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

```

825   \def\@tempc##1##2##3##4\@nil{##3}%
826   \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
827   \@tempa}{%
828   \protect\G@refundefinedtrue
829   \nfss@text{\reset@font\bfseries ??}%
830   \@latex@warning{Reference ‘#2’ on page \thepage \space
831   with undefined sentence number}%
832   }\noexpand\@nil}%
833   \let\@@protect\protect
834   \let\protect\noexpand
835   \expandafter\edef\csname r@#2\endcsname{{\@tempa}\@tempb}%
836   \let\protect\@@protect
837   \ref#1{#2}%
838 \endgroup
839 \fi
840 }

```

\refSentenceL  
\ref@SentenceX

```

841 % Das Gleiche zwingend lang.
842 \newcommand*{\refSentenceL}{%
843   \@ifstar {\ref@SentenceX0*}{\ref@SentenceX0{}}
844 }
845 \newcommand*{\ref@SentenceX}[3]{%
846   \begingroup
847   \def\parcite@format{#1}%

```

```

848 \let\sentencecite@format\parcite@format
849 \ref@Sentence{#2}{#3}%
850 \endgroup
851 }

```

`\refSentenceS` Das Gleiche zwingend kurz.

```

852 \newcommand*{\refSentenceS}{%
853 \@ifstar {\ref@SentenceX1*}{\ref@SentenceX1{}}
854 }

```

`\refSentenceN` Das Gleiche zwingend numerisch.

```

855 \newcommand*{\refSentenceN}{%
856 \@ifstar {\ref@SentenceX2*}{\ref@SentenceX2{}}
857 }

```

`\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
858 \newcounter{sentence}[par]
859 \renewcommand*{\thesentence}{\arabic{sentence}}
860 \newcommand*{\theHsentence}{\theHpar-\arabic{sentence}}
861 \renewcommand*{\p@sentence}{\expandafter\p@@sentence}
862 \newcommand*{\p@@sentence}[1]{\p@par{\par@cite{\thepar}}%
863 \scrjura@separator{\nobreakspace}}{\sentence@cite{#1}}
864 \newcommand*{\contract@sentence}{%
865 \refstepcounter{sentence}%
866 \textsuperscript{\thesentence}\nobreak\hskip\z@
867 }

```

`\parciteformat` Zitierstil für Absätze.

```

868 \DeclareRobustCommand*{\par@cite}[1]{\parciteformat{#1}}
869 \newcommand*{\parciteformat}[1]{%
870 \ifcase \parcite@format
871 \expandafter\parlongformat
872 \or
873 \expandafter\parshortformat
874 \or
875 \expandafter\parnumericformat
876 \else
877 \unskip\expandafter\@gobble
878 \fi
879 {#1}%

```

880 }

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

```
881 \DeclareRobustCommand*\sentence@cite}[1]{\sentenceciteformat{#1}}
882 \newcommand*\sentenceciteformat[1]{%
883   \ifcase \sentencecite@format
884     \expandafter\sentencelongformat
885   \or
886     \expandafter\sentenceshortformat
887   \or
888     \expandafter\sentencenumericformat
889   \else
890     \unskip\expandafter\@gobble
891   \fi
892   {#1}%
893 }
```

`\parlongformat` Formatierung in den sechs Formen.

```
\parshortformat
\parnumericformat 894 \newcommand*\parlongformat[1]{\parname~#1}
895 \newcommand*\parshortformat[1]{\parshortname~#1}
\sentencelongformat 896 \newcommand*\parnumericformat[1]{\@Roman{\number #1\relax}}
\sentenceshortformat 897 \newcommand*\sentencelongformat[1]{\sentencename~#1}
\sentencenumericformat 898 \newcommand*\sentenceshortformat[1]{\sentenceshortname~#1}
899 \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 900 \newcommand*\parname{\Paragraph}
\scrjura@lang@error 901 \AtBeginDocument{%
902   \providecaptionname{german,ngerman,austrian,naustrian}\parname{Absatz}%
903   \providecaptionname{german,ngerman,austrian,naustrian}\parshortname{Abs.}%
904   \providecaptionname{german,ngerman,austrian,naustrian}\sentencename{Satz}%
905   \providecaptionname{german,ngerman,austrian,naustrian}\sentenceshortname{S.}%
906   \providecaptionname{english,american,british,canadian,USenglish,UKenglish}\parname{Paragraph}%
907   \providecaptionname{english,american,british,canadian,USenglish,UKenglish}\parshortname{Par.}%
908   \providecaptionname{english,american,british,canadian,USenglish,UKenglish}\sentencename{Sentence}%
909   \providecaptionname{english,american,british,canadian,USenglish,UKenglish}\sentenceshortname{Sent.}%
910 }
911 \providecommand*\parname{\scrjura@lang@error{\parname}}
912 \providecommand*\parshortname{\scrjura@lang@error{\parshortname}}
913 \providecommand*\sentencename{\scrjura@lang@error{\sentencename}}
914 \providecommand*\sentenceshortname{\scrjura@lang@error{\sentenceshortname}}
915 \newcommand*\scrjura@lang@error[1]{%
916   \PackageError{scrjura}{%
917     current language not supported%
```

```

918 }{%
919     Currently scrjura only supports languages ‘german’, ‘ngerman’,
920     ‘austrian’,\MessageBreak
921     ‘naustrian’, ‘english’, ‘american’, ‘british’, ‘canadian’, ‘USenglish’,
922     and\MessageBreak
923     ‘UKenglish’.\MessageBreak
924     It seems, that you are using another language (maybe ‘\language’) or
925     that\MessageBreak
926     your language selection isn’t compatible to package ‘babel’.\MessageBreak
927     Because of this you have to define ‘\string#1’ by yourself!\MessageBreak
928     It would be nice if you’ll send your definitions to the author.%
929 }%
930 \textbf{??}%
931 }

```

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

```

932 \newcommand*{\newmaxpar}[3]{%
933     \begingroup
934     \expandafter\let\csname #1@Clauseformat\endcsname\@firstofone
935     \protected@edef\@tempa{#2}\@onelevel@sanitize\@tempa
936     \expandafter\xdef\csname max@#1@\@tempa\endcsname{#3}%
937 \endgroup
938 }
939 \newcommand*{\getmaxpar}[3]{%
940     \begingroup
941     \expandafter\let\csname #2@Clauseformat\endcsname\@firstofone
942     \protected@edef\@tempa{#3}%
943     \@onelevel@sanitize\@tempa
944     \expandafter\ifx \csname max@#2@\@tempa\endcsname\relax
945         \edef\@tempa{\endgroup\edef\noexpand#1{\expandafter\the\value{par}}}%
946     \else
947         \edef\@tempa{\endgroup
948             \edef\noexpand#1{\csname max@#2@\@tempa\endcsname}}%
949     \fi
950     \@tempa
951 }

```

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.

```

952 \AtBeginDocument{%
953     \immediate\write\@mainaux{%
954         \string\providecommand*{\string\newmaxpar}[3]{%
955             }%
956     }

```

# Index

Numbers written in *italic* 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) .....	<u>440</u>	\if@juratitlepagebreak .....	<u>50</u>
\@defjuraenvironment .....	<u>323</u>	\if@juratotoc .....	<u>15</u>
\@doendpe .....	<u>588</u>	\ifcontract@dummy .....	<u>435</u>
		\ifparnumber .....	<i>12</i> , <u>618</u>
A		J	
\AutoPar .....	<i>12</i> , <u>561</u>	juratitlepagebreak (Option) .....	<u>0</u> , <u>50</u>
C		juratocindent (Option) .....	<u>34</u>
\Clause .....	<u>5</u>	\juratoclevel .....	<u>4</u>
\Clauseformat .....	<u>273</u>	juratoclevel (Zähler) .....	<u>15</u>
\Clausemark .....	<u>74</u>	juratocnumberwidth (Option) ..	<u>34</u>
\contract .....	<u>273</u>	juratotoc (Option) .....	<u>0</u> , <u>15</u>
contract (Umgebung) .....	<u>273</u>	L	
contract (Option) .....	<u>0</u> , <u>12</u>	\l@cpar .....	<u>606</u>
\contract@Clauseformat .....	<u>273</u>	M	
\contract@everypar .....	<u>618</u>	\ManualPar .....	<i>12</i> , <u>561</u>
\contract@paragraph .....	<u>409</u>	markboth (Option) .....	<u>74</u>
\contract@paragraph@font ...	<u>437</u>	markright (Option) .....	<u>74</u>
\contract@postskip .....	<u>431</u>	N	
\contract@preskip .....	<u>431</u>	\newmaxpar .....	<i>12</i> , <u>932</u>
\contract@sentence .....	<u>858</u>	noentry (Option) .....	<u>409</u>
\contract@subparagraph .....	<u>504</u>	nohead (Option) .....	<u>409</u>
contractAbsoluteClause (Zähler)	<u>273</u>	notocentry (Option) .....	<u>409</u>
contractClause (Zähler) .....	<u>273</u>	number (Option) .....	<u>430</u>
contractSubClause (Zähler) ...	<u>273</u>	O	
\cpar@indent .....	<u>34</u>	Optionen:	
\cpar@numberwidth .....	<u>34</u>	contract .....	<u>0</u> , <u>12</u>
D		dummy .....	<u>435</u>
\DeclareNewJuraEnvironment	<i>10</i> , <u>305</u>	entry .....	<u>409</u>
\DeprecatedParagraph .....	<u>209</u>	head .....	<u>409</u>
dummy (Option) .....	<u>435</u>	juratitlepagebreak .....	<u>0</u> , <u>50</u>
E		juratocindent .....	<u>34</u>
\ellipsispar .....	<i>6</i> , <u>257</u>	juratocnumberwidth .....	<u>34</u>
entry (Option) .....	<u>409</u>	juratotoc .....	<u>0</u> , <u>15</u>
G		markboth .....	<u>74</u>
\getmaxpar .....	<i>12</i> , <u>932</u>	markright .....	<u>74</u>
H		noentry .....	<u>409</u>
head (Option) .....	<u>409</u>	nohead .....	<u>409</u>

notocentry	409	\refL	678
number	430	\refN	694
paragraphmark	0, 74	\refPar	12, 740
parcitename	123	\refParagraph	209
parnumber	51	\refParagraphN	209
postskip	0, 431	\refParL	775
preskip	0, 431	\refParN	789
ref	123	\refParS	786
sentencecitename	123	\refS	686
title	409	\refSentence	12, 806
tocentry	409	\refSentenceL	841
		\refSentenceN	855
		\refSentenceS	852
<b>P</b>			
\p@par	618	<b>S</b>	
\p@sentence	858	\scrjura@@usetype	616
\Par	12	\scrjura@afterheading	568
\par	12	\scrjura@doendpe	588
par (Zähler)	618	\scrjura@env@type	256
\Paragraph	209	\scrjura@everypar	587
\ParagraphCompatibilityHacks	209	\scrjura@lang@error	900
paragraphmark (Option)	0, 74	\scrjura@separator	615
\parcite@fromat	123	\scrjura@special@par	671
\parciteformat	12, 868	\scrjura@special@reset@par	671
parcitename (Option)	123	\scrjura@usetype	616
\parellipsis	6, 257	\Sentence	9
\parformat	12, 618	\sentence	12
\parlongformat	894	sentence (Zähler)	858
\parname	12, 900	\sentencecite@fromat	123
parnumber (Option)	51	\sentenceciteformat	12, 881
\parnumberfalse	9, 12	sentencecitename (Option)	123
\parnumbertrue	9, 12	\sentencelongformat	894
\parnumericformat	894	\sentencename	12, 900
\parshortformat	894	\sentencenumericformat	894
\parshortname	900	\sentenceshortformat	894
postskip (Option)	0, 431	\sentenceshortname	900
preskip (Option)	0, 431	\SubClause	6
		\SubParagraph	209
<b>R</b>			
ref (Option)	123	<b>T</b>	
\ref@Clause	702	\thecontractClause	12, 273
\ref@ClauseN	731	\thecontractSubClause	12, 273
\ref@L	678	\theH@AbsClause	440
\ref@N	694	\theHClause	440
\ref@Par	740	\theHpar	618
\ref@ParX	775	\theHsentence	858
\ref@S	686	\theHSubClause	440
\ref@Sentence	806	\thepar	12, 618
\ref@SentenceX	841	\thesentence	12, 858
\refClause	12, 702	\thisparnumber	671
\refClauseN	731		



title (Option) . . . . .	409	<b>Z</b>
tocentry (Option) . . . . .	409	Zähler:
\toclevel@cpar . . . . .	15	@AbsClause . . . . .
		contractAbsoluteClause . .
		contractClause . . . . .
		contractSubClause . . . . .
		juratoclevel . . . . .
		par . . . . .
		sentence . . . . .
<b>U</b>		
Umgebungen:		
contract . . . . .	273	

## Change History

1.0	\protected@edef durch	
\DeprecatedParagraph: wird	\edef ersetzt . . . . .	38
noch ein paar Versionen einen		
Fehler melden . . . . .	v0.5d	
18	\contract: \jura@env@type	
\Paragraph: wird noch ein paar	wird gesetzt . . . . .	19
Versionen einen Fehler mel-	fehlende Fehlermeldung er-	
den . . . . .	gänzt . . . . .	19
18	Umgebung definiert sich nicht	
\ParagraphCompatibilityHacks:	mehr selbst um . . . . .	19
wird noch ein paar Versionen	ref: neue Einstellungen	
verbleiben . . . . .	nopar, nosentence,	
18	OnlyParagraph . . . . .	16
\refParagraph: wird noch ein	\p@par: \jura@usetype mit Ar-	
paar Versionen einen Fehler	gument \jura@env@type ein-	
melden . . . . .	gefügt . . . . .	29
18	\parciteformat: Argument ver-	
\refParagraphN: wird noch ein	schoben . . . . .	36
paar Versionen einen Fehler	\scrjura@usetype: neu (in-	
melden . . . . .	tern) . . . . .	29
18	\scrjura@env@type: neu (in-	
\SubParagraph: wird noch ein	tern) . . . . .	19
paar Versionen einen Fehler	\scrjura@usetype: neu (in-	
melden . . . . .	tern) . . . . .	29
18	\sentenceciteformat: Argu-	
v0.5b	ment verschoben . . . . .	37
juratitlepagebreak: neue Op-	v0.5e	
tion . . . . .	\Clausemark: Beachtung von	
14	\MakeMarkcase . . . . .	15
v0.5c		
\contract@everypar: \contract@Clauseformat	v0.6	
expandiert beim Schreiben zu	parnumber: neue Option . . . .	14
seinem Argument . . . . .	\cpar@indent: neu (intern) . .	14
30	\cpar@numberwidth: neu (in-	
erstes Argument von	tern) . . . . .	14
\newmaxpar und zweites Ar-	\getmaxpar: Argument 3 muss	
gument von \getmaxpar		
ist contract nicht		
contractpars . . . . .		
30		
\getmaxpar: \#2@Clauseformat		
expandiert zu seinem Argu-		
ment . . . . .		
38		

mit <code>\protected@edef</code> expan-	Satznummer bei manuell num-
diert werden . . . . .	merierten Absätzen auf 0
38	statt 1 initialisiert, weil
<code>\if@juratoc</code> : Schalter durch	<code>\Sentence</code> die Nummer als
Zähler überflüssig . . . . .	erstes hochzählt. . . . .
13	30
<code>\ifparnumber</code> : Definition des	Zähler <code>contractAbsoluteClause</code>
Schalters in die Definition der	wird verwendet . . . . .
Option verschoben . . . . .	30
29	<code>\contract@paragraph</code> : Abstand
<code>entry</code> : Option veraltet . . . . .	nach der Nummer als Be-
23	standteil der Nummer setzen
<code>tocentry</code> : neue Option . . . . .	25
23	Zähler <code>contractAbsoluteClause</code>
<code>noentry</code> : Option veraltet . . . . .	wird erhöht . . . . .
23	24
<code>notocentry</code> : neue Option . . . . .	<code>\contract@ subparagraph</code> : Ab-
23	stand nach der Nummer als
<code>juratocindent</code> : neu . . . . .	Bestandteil der Nummer set-
14	zen . . . . .
<code>juratoclevel</code> : Zählerdefinition	26
verschoben . . . . .	Zähler <code>contractAbsoluteClause</code>
13	wird erhöht . . . . .
<code>juratocnumberwidth</code> : neu . . .	26
14	<code>ref</code> : Verwendung der reno-
v0.6a	vierten Schnittstelle mit
<code>parnumber</code> : Meldung bei falschen	<code>\FamilyKeyStateProcessed</code>
Werten korrigiert . . . . .	. . . . .
14	16
<code>\contract@paragraph</code> : fehlen-	<code>\ellipsispar</code> : neu . . . . .
des <code>\par</code> ergänzt . . . . .	19
25	<code>contractAbsoluteClause</code> : neu-
<code>\contract@ subparagraph</code> : feh-	er (interner) Zähler über al-
lendes <code>\par</code> ergänzt . . . . .	le Paragraphen, um ggf. die
26	Zählung neu beginnen zu
v0.6b	können . . . . .
<code>\Clauseformat</code> : <code>\paragraphformat</code>	19
umdefiniert . . . . .	<code>\parellipsis</code> : neu . . . . .
19	19
<code>\contract@paragraph</code> : Verwen-	<code>\parformat</code> : neues Font-
dung von <code>\contract@Clauseformat</code>	Element <code>parnumber</code> . . . . .
auch bei manueller Numme-	29
rierung der Paragraphen. . .	<code>\scrjura@lang@error</code> : Verwen-
25	dung von <code>\PackageError</code> an
<code>\newmaxpar</code> : <code>\#1@Clauseformat</code>	Stelle von <code>\PackageErrorNoLine</code>
expandiert zu seinem Argu-	. . . . .
ment . . . . .	37
38	<code>\scrjura@special@par</code> : neue
v0.7	Anweisung (intern) . . . . .
<code>paragraphmark</code> : Verwen-	30
dung der renovier-	<code>\scrjura@special@reset@par</code> :
ten Schnittstelle mit	neue Anweisung (intern) . .
<code>\FamilyKeyStateProcessed</code>	30
. . . . .	<code>\thisparnumber</code> : neue Anwei-
15	sung . . . . .
<code>parnumber</code> : Verwendung der re-	30
novierten Schnittstelle mit	<code>\toclevel@ cpar</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>
30	. . . . .
Im Fall von nicht rein numme-	14
rischen Absatznummern die	<code>juratoclevel</code> : Verwendung der
Absätze auf jeden Fall num-	
merieren . . . . .	
30	

renovierten Schnittstelle mit <code>\FamilyKeyState</code> . . . . .	13	umbenannt . . . . .	19
<code>juratocnumberwidth:</code> Ver- wendung der renovier- ten Schnittstelle mit <code>\FamilyKeyStateProcessed</code> . . . . .	14	<code>\scrjura@everypar:</code> <code>\jura@everypar</code> umbenannt . . . . .	28
		<code>\scrjura@separator:</code> <code>\jura@separator</code> umbenannt . . . . .	29
		<code>\scrjura@usetype:</code> <code>\jura@usetype</code> umbenannt . . . . .	29
v0.7a		v0.9	
<code>\contract@sentence:</code> <code>\nobreak\hskip\z@</code> eingefügt, um die Trennung des ersten Worts nach der Satzmarkierung zu ermögli- chen . . . . .	36	<code>\z@defjuraenvironment:</code> Neue (interne) Anweisung . . . . .	21
auf <code>\textsuperscript</code> umge- stellt . . . . .	36	<code>\DeclareNewJuraEnvironment:</code> Neue Anweisung . . . . .	20
v0.7b		v0.9a	
General: Generelle Umbenen- nung von „Paragraph“ in „Clause“ für alle Befehle, Zähler und Optionen . . . . .	12	<code>paragraphmark:</code> Wert wird in der internen Liste erfasst . .	15
<code>\contract@everypar:</code> <code>contractAbsoluteClause</code> flexibilisiert . . . . .	30	<code>parnumber:</code> Wert wird in der in- ternen Liste erfasst . . . . .	14
<code>\contract@paragraph:</code> <code>contractAbsoluteClause</code> flexibilisiert . . . . .	24	<code>ref:</code> Wert wird in der internen Liste erfasst . . . . .	16
<code>\ellipsispar:</code> <code>\thecontractAbsoluteClause</code> flexibilisiert . . . . .	19	<code>juratocindent:</code> Wert wird in der internen Liste erfasst . .	14
<code>\getmaxpar:</code> Notfallcode in die aux-Datei . . . . .	38	<code>juratoclevel:</code> Wert wird in der Liste erfasst . . . . .	13
<code>\scrjura@usetype:</code> <code>\jura@usetype</code> umbenannt . . . . .	29	<code>juratocnumberwidth:</code> Wert wird in der internen Liste er- fasst . . . . .	14
<code>\scrjura@afterheading:</code> <code>\jura@afterheading</code> umbe- nannt . . . . .	27	v0.9b	
benötigt kein Argument mehr	27	<code>contractSubClause:</code> muss unbe- dingt den Zähler <code>SubClause</code> statt <code>contractSubClause</code> verwenden . . . . .	19
<code>\scrjura@env@type:</code> <code>\jura@env@type</code>		<code>\thecontractClause:</code> muss un- bedingt den Zähler <code>Clause</code> statt <code>contractClause</code> ver- wenden . . . . .	19