

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

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

2015/05/13

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 Kolummentitel . . . . .	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 v0.9d 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 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 Kolummentitel

`paragraphmark` Bei langen Verträgen können automatische Kolummentitel hilfreich sein. Allerdings hängt die Ausgabe hier davon ab, wie die diese erzeugt werden. `scrjura` 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  
`\parellipsis`    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

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. »L<sup>A</sup>T<sub>E</sub>X-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 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{tobasic}
```

`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     \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@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
34 \newcommand*{\cpar@numberwidth}{2em}
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}{%
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 \KOMAv@kav@xreplacevalue{.scrjura.sty}{paragraphmark}{#1}%
119 }
120 \KOMAv@kav@add{.scrjura.sty}{paragraphmark}{false}
121 \KOMAv@DeclareDeprecatedOption[scrjura]{markright}{paragraphmark=forceright}
122 \KOMAv@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

```

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

```

Per Option kann die Voreinstellung verändert werden.

```

125 \KOMAv@key{ref}{%
126   \begingroup
127     \KOMAv@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},{numericssentence}{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     \KOMAC@kav@removekey{.scrjura.sty}{ref}%
184     \ifcase\parcite@format
185         \KOMAC@kav@add{.scrjura.sty}{ref}{parlong}%
186     \or
187         \KOMAC@kav@add{.scrjura.sty}{ref}{parshort}%
188     \or
189         \KOMAC@kav@add{.scrjura.sty}{ref}{parnumeric}%
190     \or
191         \KOMAC@kav@add{.scrjura.sty}{ref}{paroff}%
192     \fi
193     \ifcase\sentencecite@format
194         \KOMAC@kav@add{.scrjura.sty}{ref}{sentencelong}%
195     \or
196         \KOMAC@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 209 \providecommand*\DeprecatedParagraph{%
\refParagraphN 210 \PackageError{scrjura}{modification of old document needed}{%
\DeprecatedParagraph 211 It seem that this document was made for scrjura up to version
\ParagraphCompatibilityHacks 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 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`

```
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@skiphyperref\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{KOMAarg}
324 \DefineFamilyMember{KOMAarg}
325 \DefineFamilyKey{KOMAarg}{Clause}{%
326   \expandafter\gdef\csname \scrjura@env@type @Clause\endcsname{#1}%
327 }
328 \DefineFamilyKey{KOMAarg}{SubClause}{%
329   \expandafter\gdef\csname \scrjura@env@type @SubClause\endcsname{#1}%
330 }
331 \DefineFamilyKey{KOMAarg}{Sentence}{%
332   \expandafter\gdef\csname \scrjura@env@type @Sentence\endcsname{#1}%
333 }
334 \DefineFamilyKey{KOMAarg}{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]{KOMAarg}{#2}%
350 \endgroup
351 \@ifundefined{#1@Clauseformat}{%
352   \expandafter\DeclareRobustCommand\expandafter*%
353   \csname #1@Clauseformat\endcsname[1]{\Clauseformat{##1}}%
354 }{}%

355 \expandafter\renewcommand\expandafter*\csname the#1Clause\endcsname{%
356   \csname #1@Clauseformat\endcsname{\arabic{#1Clause}}}%

```

Umgebung:

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

```

405     \fi
406   }{%
407     #4%
408     \par
409   }%
410 }

```

`\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 411 \define@key{contract}{title}{%
notocentry 412   \def\contract@title{#1}%
413   \ifx\contract@entry\relax\def\contract@entry{\contract@title}\fi
414   \ifx\contract@head\relax\def\contract@head{\contract@title}\fi
415 }
416 \define@key{contract}{entry}{%
417   \PackageWarning{scrjura}{deprecated option 'entry'.\MessageBreak
418     You should use option 'tocentry' instead of\MessageBreak
419     option 'entry'%
420   }%
421   \def\contract@entry{#1}}
422 \define@key{contract}{tocentry}{\def\contract@entry{#1}}
423 \define@key{contract}{noentry}[]{%
424   \PackageWarning{scrjura}{deprecated option 'noentry'.\MessageBreak
425     You should use option 'notocentry' instead of\MessageBreak
426     option 'noentry'%
427   }%
428   \let\contract@entry\relax}
429 \define@key{contract}{notocentry}[]{\let\contract@entry\relax}
430 \define@key{contract}{head}{\def\contract@head{#1}}
431 \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.

```

432 \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.

```
433 \newcommand*{\contract@preskip}{2\baselineskip}
434 \newcommand*{\contract@postskip}{\baselineskip}
435 \define@key{contract}{preskip}{\def\contract@preskip{#1}}
436 \define@key{contract}{postskip}{\def\contract@postskip{#1}}
```

`dummy` Über diese 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.

```
437 \newif\ifcontract@dummy
438 \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!

```
439 \newkomafont{contract.Clause}{\sffamily\bfseries\large}
440 \newcommand*{\contract@paragraph@font}{\usekomafont{contract.Clause}%
441 \@hangfrom}
```

```
@AbsClause
\theH@AbsClause
\theHClause
\theHSubClause
442 % \textsf{hyperref}-Code, der noch nicht getestet ist!
443 \newcounter{@AbsClause}
444 \newcommand*{\theH@AbsClause}{P-\arabic{@AbsClause}}
445 \newcommand*{\theHClause}{\theH@AbsClause}
446 \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.

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



```

453 \let\contract@number\relax
454 \contract@dummysfalse
455 \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.

```

456 \ifcontract@dummys\else
457   \par
458   \@afterindentfalse
459   \addvspace{\contract@preskip}%
460 \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.

```

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

```

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

```

487 \let\theSubClause\theClause

```

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

```

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

```

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

```

504   \scrjura@afterheading
505   \fi
506 }

```

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

```

507 \newcommand*{\contract@subparagraph}[1]{%
508   \stepcounter{\scrjura@env@type AbsoluteClause}%
509   \ManualPar\parnumbertrue
510   \let\contract@title\relax
511   \let\contract@entry\relax
512   \let\contract@head\relax
513   \let\contract@number\relax
514   \contract@dummysfalse
515   \ifx\relax#1\relax\else\setkeys{contract}{#1}\fi
516   \ifcontract@dummysfalse
517     \par
518     \@afterindentfalse
519     \vskip\contract@preskip
520   \fi
521   \ifx\contract@number\relax
522     \let\p@SubClause\@empty
523     \let\theSubClause\thecontractSubClause
524     \refstepcounter{SubClause}%
525   \else

```

```

526     \begingroup
527         \let\@elt\@stpelt
528         \cl@SubClause
529     \endgroup
530     \protected@edef\theSubClause{\theClause\contract@number}%
531     \protected@edef\@currentlabel{\theSubClause}%
532 \fi
533 \stepcounter{@AbsClause}%
534 \begingroup\expandafter\expandafter\expandafter\endgroup
535 \expandafter\ifx\csname if@skiphyperref\endcsname\relax
536 \else
537     \expandafter\let\csname if@scrjura@skiphyperref\endcsname
538     \csname if@skiphyperref\endcsname
539 \fi
540 \if@scrjura@skiphyperref\else
541     \hyper@refstepcounter{@AbsClause}%
542     \typeout{absolute Nummer: \the@AbsClause^^JLabel: '\@currentHref'}%
543 \fi
544 \ifcontract@dummys\else
545     \begingroup
546         \if@juratitlepagebreak\else\interlinepenalty\@M\fi
547         \contract@paragraph@font{\theSubClause
548             \ifx\contract@title\relax\else\enskip\fi}%
549         \contract@title
550         \ifx\contract@entry\relax\else
551             \addxcontentsline{toc}{cpar}{\theSubClause}\contract@entry
552             \addxcontentsline{cpa}{cpar}{\theSubClause}\contract@entry
553         \fi
554         \ifx\contract@head\relax\else
555             \expandafter\Clausemark\expandafter{%
556                 \expandafter\theSubClause\expandafter\enskip\contract@head}%
557         \fi
558     \par
559 \endgroup
560 \nobreak\vskip\contract@postskip
561 \scrjura@afterheading
562 \fi
563 }

```

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

```

564 \newcommand*{\AutoPar}{%
565     \expandafter\let\expandafter\scrjura@everypar
566     \csname \scrjura@env@type @everypar\endcsname
567 }
568 \newcommand*{\ManualPar}{%
569     \let\scrjura@everypar\relax
570 }

```

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

```
571 \newcommand*{\scrjura@afterheading}{%
572   \@nobreaktrue
573   \everypar{%
574     \if@nobreak
575       \@nobreakfalse
576       \clubpenalty \@M
577       \if@afterindent \else
578         {\setbox\z@\lastbox}%
579       \fi
580     \else
581       \clubpenalty \@clubpenalty
582       \everypar{%
583         \scrjura@everypar
584       }%
585     \fi
586   \scrjura@everypar
587 }%
588 \AutoPar
589 }
```

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

```
590 \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`

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

```

604     \csname scrjura@everypar\endcsname
605   }%
606   \@endpefalse
607 }%
608 }

```

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

```

609 \newcommand*\l@cpar}[2]{%
610   \ifnum\value{juratoclevel}>\value{tocdepth}\else
611     \scr@ifundefinedorrelax{bprot@dottedtocline}{%
612       \dottedtocline
613     }{%
614       \bprot@dottedtocline
615     }{\value{juratoclevel}}{\cpar@indent}{\cpar@numberwidth}{#1}{#2}%
616   \fi
617 }

```

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

```

618 \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.

```

619 \DeclareRobustCommand*\scrjura@usetype}[1]{\scrjura@@usetype{#1}}
620 \newcommand*\scrjura@@usetype}[1]{}

```

`\contract@everypar` `\scrjura@everpar` für Verträge.

`\ifparnumber` Der Schalter gibt an, ob überhaupt mit Absatznummern gearbeitet werden soll. Sind Absatznummern deaktiviert, werden auch keine manuellen  
`par`  
`\thepar` Absatznummern gesetzt. Die Absätze werden dann auch nicht gezählt. Ansonsten 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.

```

621 \newcounter{par}
622 \renewcommand*\thepar{\arabic{par}}
623 \newcommand*\theHpar{\theH@AbsClause-\Roman{par}}

```

```

624 \newcommand*{\parformat}{(\thepar)}
625 \newkomafont{parnumber}{}
626 \renewcommand*\p@par{\scrjura@usetype{\scrjura@env@type}\theSubClause\scrjura@
627 \@addtoreset{par}{Clause}
628 \@addtoreset{par}{SubClause}

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

```

```
672 \fi
673 }
```

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

```
674 \newcommand*{\thisparnumber}[1]{%
675 \def\scrjura@special@par{#1}%
676 }
677 \newcommand*{\scrjura@special@par}{}
678 \let\scrjura@special@par\relax
679 \newcommand*{\scrjura@special@reset@par}{}
680 \let\scrjura@special@reset@par\relax
```

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

```
\ref@L
681 \newcommand*{\refL}{\@ifstar {\ref@L*}{\ref@L{}}}
682 \newcommand*{\ref@L}[2]{%
683 \begingroup
684 \def\parcite@format{0}%
685 \let\sentencecite@format\parcite@format
686 \ref#1{#2}%
687 \endgroup
688 }
```

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

```
\ref@S
689 \newcommand*{\refS}{\@ifstar {\ref@S*}{\ref@S{}}}
690 \newcommand*{\ref@S}[2]{%
691 \begingroup
692 \def\parcite@format{1}%
693 \let\sentencecite@format\parcite@format
694 \ref#1{#2}%
695 \endgroup
696 }
```

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

```
\ref@N
697 \newcommand*{\refN}{\@ifstar {\ref@N*}{\ref@N{}}}
698 \newcommand*{\ref@N}[2]{%
699 \begingroup
700 \def\parcite@format{2}%
701 \let\sentencecite@format\parcite@format
702 \ref#1{#2}%
703 \endgroup
704 }
```

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

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

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

```
713   \expandafter\expandafter\expandafter\expandafter
714   \expandafter\expandafter\expandafter\def
715   \expandafter\expandafter\expandafter\expandafter
716   \expandafter\expandafter\expandafter\@tempb
717   \expandafter\expandafter\expandafter\expandafter
718   \expandafter\expandafter\expandafter{%
719   \expandafter\expandafter\expandafter\@gobble\csname r@#2\endcsname}%
```

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

```
720   \def\@tempc##1##2\@nil{##1}%
721   \let\scrjura@separator\@gobble
722   \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
723   \csname r@#2\endcsname\noexpand\@nil}%
```

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

```
724   \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
725   \@tempa\@nil}%
726   \let\@@protect\protect
727   \let\protect\noexpand
728   \expandafter\edef\csname r@#2\endcsname{{\@tempa}\@tempb}%
729   \let\protect\@@protect
730   \ref#1{#2}%
731 \endgroup
732 \fi
733 }
```

`\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.

```
734 \newcommand*\refClauseN}{%
735   \@ifstar {\ref@ClauseN*}{\ref@ClauseN{}}
```



```

736 }
737 \newcommand*{\ref@ClauseN}[2]{%
738   \begingroup
739     \let\Clauseformat\relax
740     \ref@Clause{#1}{#2}%
741   \endgroup
742 }

```

`\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.

```

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

```

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

```

751     \expandafter\expandafter\expandafter\expandafter
752     \expandafter\expandafter\expandafter\def
753     \expandafter\expandafter\expandafter\expandafter
754     \expandafter\expandafter\expandafter\@tempb
755     \expandafter\expandafter\expandafter\expandafter
756     \expandafter\expandafter\expandafter{%
757       \expandafter\expandafter\expandafter\@gobble\csname r@#2\endcsname}%

```

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

```

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

```

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

```

762     \def\@tempc##1##2##3\@nil{##2}%
763     \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
764       \@tempa{%
765         \protect\G@refundefinedtrue
766         \nfss@text{\reset@font\bfseries ??}%
767         \@latex@warning{Reference ‘#2’ on page \thepage \space
768           with undefined par number}%
769       }\noexpand\@nil}%
770     \let\@@protect\protect

```

```

771     \let\protect\noexpand
772     \expandafter\edef\csname r@#2\endcsname{\@tempa}\@tempb}%
773     \let\protect\@@protect
774     \ref#1{#2}%
775 \endgroup
776 \fi
777 }

\refParL
\ref@ParX
778 % Das Gleiche zwingend lang.
779 \newcommand*\refParL}{%
780   \@ifstar {\ref@ParX0*}{\ref@ParX0{}}
781 }
782 \newcommand*\ref@ParX}[3]{%
783   \begingroup
784     \def\parcite@format{#1}%
785     \let\sentencecite@format\parcite@format
786     \ref@Par{#2}{#3}%
787   \endgroup
788 }

\refParS Das Gleiche zwingend kurz.

789 \newcommand*\refParS}{%
790   \@ifstar {\ref@ParX1*}{\ref@ParX1{}}
791 }

\refParN Das Gleiche zwingend numerisch.

792 \newcommand*\refParN}{%
793   \@ifstar {\ref@ParN2*}{\ref@ParN2{}}
794 }
795 \newcommand*\ref@ParN}[2]{%
796   \@ifnextchar [%]
797     {\ref@@ParN{#1}{#2}}%
798     {\ref@ParX{#1}{#2}}%
799 }
800 \newcommand*\ref@@ParN}{%
801   \def\ref@@ParN#1#2[#3]#4{%
802     \begingroup
803       \renewcommand*\parnumericformat}[1]{%
804         \csname @#3\endcsname{\number ##1\relax}%
805       }%
806       \ref@ParX{#1}{#2}{#4}%
807     \endgroup
808 }

```

`\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.

```
809 \newcommand*{\refSentence}{%
810   \@ifstar {\ref@Sentence*}{\ref@Sentence{}}
811 }
812 \newcommand*{\ref@Sentence}[2]{%
813   \expandafter\ifx\csname r@#2\endcsname\relax
814     \ref#1{#2}%
815   \else
816     \begingroup
```

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

```
817   \expandafter\expandafter\expandafter\expandafter
818   \expandafter\expandafter\expandafter\def
819   \expandafter\expandafter\expandafter\expandafter
820   \expandafter\expandafter\expandafter\@tempb
821   \expandafter\expandafter\expandafter\expandafter
822   \expandafter\expandafter\expandafter{%
823     \expandafter\expandafter\expandafter\@gobble\csname r@#2\endcsname}%
```

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

```
824   \def\@tempc##1##2\@nil{##1}%
825   \let\scrjura@separator\@gobble
826   \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
827     \csname r@#2\endcsname\noexpand\@nil}%
```

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

```
828   \def\@tempc##1##2##3##4\@nil{##3}%
829   \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
830     \@tempa}{%
831     \protect\G@refundefinedtrue
832     \nfss@text{\reset@font\bfseries ??}%
833     \latex@warning{Reference ‘#2’ on page \thepage \space
834       with undefined sentence number}%
835     }\noexpand\@nil}%
836   \let\@@protect\protect
837   \let\protect\noexpand
838   \expandafter\edef\csname r@#2\endcsname{\@tempa\@tempb}%
839   \let\protect\@@protect
840   \ref#1{#2}%
841   \endgroup
842 \fi
843 }
```

```

\refSentenceL
\ref@SentenceX 844 % Das Gleiche zwingend lang.
845 \newcommand*{\refSentenceL}{%
846   \@ifstar {\ref@SentenceX0*}{\ref@SentenceX0{}}
847 }
848 \newcommand*{\ref@SentenceX}[3]{%
849   \begingroup
850     \def\parcite@format{#1}%
851     \let\sentencecite@format\parcite@format
852     \ref@Sentence{#2}{#3}%
853   \endgroup
854 }

\refSentenceS  Das Gleiche zwingend kurz.

855 \newcommand*{\refSentenceS}{%
856   \@ifstar {\ref@SentenceX1*}{\ref@SentenceX1{}}
857 }

\refSentenceN  Das Gleiche zwingend numerisch.

858 \newcommand*{\refSentenceN}{%
859   \@ifstar {\ref@SentenceX2*}{\ref@SentenceX2{}}
860 }

\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
861 \newcounter{sentence}[par]
862 \renewcommand*{\thesentence}{\arabic{sentence}}
863 \newcommand*{\theHsentence}{\theHpar-\arabic{sentence}}
864 \renewcommand*{\p@sentence}{\expandafter\p@@sentence}
865 \newcommand*{\p@@sentence}[1]{\p@par{\par@cite{\theHpar}}%
866   \scrjura@separator{\nobreakspace}{\sentence@cite{#1}}}
867 \newcommand*{\contract@sentence}{%
868   \refstepcounter{sentence}%
869   \textsuperscript{\thesentence}\nobreak\hskip\z@
870 }

\parciteformat  Zitierstil für Absätze.

871 \DeclareRobustCommand*{\par@cite}[1]{\parciteformat{#1}}
872 \newcommand*{\parciteformat}[1]{%
873   \ifcase \parcite@format

```

```

874 \expandafter\parlongformat
875 \or
876 \expandafter\parshortformat
877 \or
878 \expandafter\parnumericformat
879 \else
880 \unskip\expandafter\@gobble
881 \fi
882 {#1}%
883 }

```

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

```

884 \DeclareRobustCommand*\sentence@cite}[1]{\sentenceciteformat{#1}}
885 \newcommand*\sentenceciteformat}[1]{%
886 \ifcase \sentencecite@format
887 \expandafter\sentencelongformat
888 \or
889 \expandafter\sentenceshortformat
890 \or
891 \expandafter\sentencenumericformat
892 \else
893 \unskip\expandafter\@gobble
894 \fi
895 {#1}%
896 }

```

```

\parlongformat Formatierung in den sechs Formen.
\parshortformat
\parnumericformat 897 \newcommand*\parlongformat}[1]{\parname~#1}
\sentencelongformat 898 \newcommand*\parshortformat}[1]{\parshortname~#1}
\sentenceshortformat 899 \newcommand*\parnumericformat}[1]{\@Roman{\number #1\relax}}
\sentencenumericformat 900 \newcommand*\sentencelongformat}[1]{\sentencename~#1}
901 \newcommand*\sentenceshortformat}[1]{\sentenceshortname~#1}
902 \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 903 \newcommand*\parname}{Paragraph}
\sentenceshortname 904 \AtBeginDocument{%
\scrjura@lang@error 905 \providecaptionname{german,ngerman,austrian,naustrian}\parname{Absatz}%
906 \providecaptionname{german,ngerman,austrian,naustrian}\parshortname{Abs.}%
907 \providecaptionname{german,ngerman,austrian,naustrian}\sentencename{Satz}%
908 \providecaptionname{german,ngerman,austrian,naustrian}\sentenceshortname{S.}%
909 \providecaptionname{english,american,british,canadian,USenglish,UKenglish}\parname{Paragraph}%
910 \providecaptionname{english,american,british,canadian,USenglish,UKenglish}\parshortname{Abs.}%
911 \providecaptionname{english,american,british,canadian,USenglish,UKenglish}\sentencename{Satz}%

```

```

912 \providecaptionname{english,american,british,canadian,USenglish,UKenglish}\sent
913 }
914 \providecommand*\parname{\scrjura@lang@error{\parname}}
915 \providecommand*\parshortname{\scrjura@lang@error{\parshortname}}
916 \providecommand*\sentencename{\scrjura@lang@error{\sentencename}}
917 \providecommand*\sentenceshortname{\scrjura@lang@error{\sentenceshortname}}
918 \newcommand*\scrjura@lang@error[1]{%
919 \PackageError{scrjura}{%
920 current language not supported%
921 }{%
922 Currently scrjura only supports languages 'german', 'ngerman',
923 'austrian',\MessageBreak
924 'naustrian', 'english', 'american', 'british', 'canadian', 'USenglish',
925 and\MessageBreak
926 'UKenglish'.\MessageBreak
927 It seems, that you are using another language (maybe '\language') or
928 that\MessageBreak
929 your language selection isn't compatible to package 'babel'.\MessageBreak
930 Because of this you have to define '\string#1' by yourself!\MessageBreak
931 It would be nice if you'll send your definitions to the author.%
932 }%
933 \textbf{??}%
934 }

```

`\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.

```

935 \newcommand*\newmaxpar[3]{%
936 \begingroup
937 \expandafter\let\csname #1@clauseformat\endcsname\@firstofone
938 \protected@edef\@tempa{#2}\@onelevel@sanitize\@tempa
939 \expandafter\xdef\csname max@#1@\@tempa\endcsname{#3}%
940 \endgroup
941 }
942 \newcommand*\getmaxpar[3]{%
943 \begingroup
944 \expandafter\let\csname #2@clauseformat\endcsname\@firstofone
945 \protected@edef\@tempa{#3}%
946 \@onelevel@sanitize\@tempa
947 \expandafter\ifx \csname max@#2@\@tempa\endcsname\relax
948 \edef\@tempa{\endgroup\edef\noexpand#1{\expandafter\the\value{par}}}%
949 \else
950 \edef\@tempa{\endgroup
951 \edef\noexpand#1{\csname max@#2@\@tempa\endcsname}}%
952 \fi
953 \@tempa
954 }

```

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.

```

955 \AtBeginDocument{%
956   \immediate\write\@mainaux{%
957     \string\providecommand*{\string\newmaxpar}[3]{ }
958   }%
959 }

```

## 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.

	<b>Symbols</b>	<code>dummy</code> (Option) . . . . .	<u>437</u>
<code>@AbsClause</code> (Zähler) . . . . .	<u>442</u>		
<code>\@defjuraenvironment</code> . . . . .	<u>323</u>	<b>E</b>	
<code>\@doendpe</code> . . . . .	<u>591</u>	<code>\ellipsispar</code> . . . . .	<i>6</i> , <u>257</u>
		entry (Option) . . . . .	<u>411</u>
	<b>A</b>		
<code>\AutoPar</code> . . . . .	<i>12</i> , <u>564</u>	<b>G</b>	
	<b>C</b>	<code>\getmaxpar</code> . . . . .	<i>12</i> , <u>935</u>
<code>\Clause</code> . . . . .	<i>5</i>	<b>H</b>	
<code>\Clauseformat</code> . . . . .	<u>273</u>	head (Option) . . . . .	<u>411</u>
<code>\Clausemark</code> . . . . .	<u>74</u>	<b>I</b>	
<code>\contract</code> . . . . .	<u>273</u>	<code>\if@juratitlepagebreak</code> . . . . .	<i>50</i>
contract (Umgebung) . . . . .	<u>273</u>	<code>\if@juratotoc</code> . . . . .	<u>15</u>
contract (Option) . . . . .	<i>0</i> , <u>12</u>	<code>\ifcontract@dummy</code> . . . . .	<u>437</u>
<code>\contract@Clauseformat</code> . . . . .	<u>273</u>	<code>\ifparnumber</code> . . . . .	<i>12</i> , <u>621</u>
<code>\contract@everypar</code> . . . . .	<u>621</u>	<b>J</b>	
<code>\contract@paragraph</code> . . . . .	<u>411</u>	juratitlepagebreak (Option) <i>0</i> , <u>50</u>	
<code>\contract@paragraph@font</code> . . . . .	<u>439</u>	juratocindent (Option) . . . . .	<u>34</u>
<code>\contract@postskip</code> . . . . .	<u>433</u>	<code>\juratoclevel</code> . . . . .	<i>4</i>
<code>\contract@preskip</code> . . . . .	<u>433</u>	juratoclevel (Zähler) . . . . .	<u>15</u>
<code>\contract@sentence</code> . . . . .	<u>861</u>	juratocnumberwidth (Option) . . . . .	<u>34</u>
<code>\contract@subparagraph</code> . . . . .	<u>507</u>	juratotoc (Option) . . . . .	<i>0</i> , <u>15</u>
contractAbsoluteClause (Zähler)	<u>273</u>	<b>L</b>	
contractClause (Zähler) . . . . .	<u>273</u>	<code>\l@cpar</code> . . . . .	<u>609</u>
contractSubClause (Zähler) . . . . .	<u>273</u>	<b>M</b>	
<code>\cpar@indent</code> . . . . .	<u>34</u>	<code>\ManualPar</code> . . . . .	<i>12</i> , <u>564</u>
<code>\cpar@numberwidth</code> . . . . .	<u>34</u>	markboth (Option) . . . . .	<u>74</u>
<b>D</b>		markright (Option) . . . . .	<u>74</u>
<code>\DeclareNewJuraEnvironment</code> <i>10</i> , <u>305</u>			
<code>\DeprecatedParagraph</code> . . . . .	<u>209</u>		

<b>N</b>		<code>\parnumberfalse</code> . . . . . 9, 12
<code>\newmaxpar</code> . . . . . 12, 935		<code>\parnumbertrue</code> . . . . . 9, 12
<code>noentry</code> (Option) . . . . . 411		<code>\parnumericformat</code> . . . . . 897
<code>nohead</code> (Option) . . . . . 411		<code>\parshortformat</code> . . . . . 897
<code>notocentry</code> (Option) . . . . . 411		<code>\parshortname</code> . . . . . 903
<code>number</code> (Option) . . . . . 432		<code>postskip</code> (Option) . . . . . 0, 433
		<code>preskip</code> (Option) . . . . . 0, 433
<b>O</b>		
Optionen:		<b>R</b>
<code>contract</code> . . . . . 0, 12		<code>ref</code> (Option) . . . . . 123
<code>dummy</code> . . . . . 437		<code>\ref@Clause</code> . . . . . 705
<code>entry</code> . . . . . 411		<code>\ref@ClauseN</code> . . . . . 734
<code>head</code> . . . . . 411		<code>\ref@L</code> . . . . . 681
<code>juratitlepagebreak</code> . . . . . 0, 50		<code>\ref@N</code> . . . . . 697
<code>juratocindent</code> . . . . . 34		<code>\ref@Par</code> . . . . . 743
<code>juratocnumberwidth</code> . . . . . 34		<code>\ref@ParX</code> . . . . . 778
<code>juratotoc</code> . . . . . 0, 15		<code>\ref@S</code> . . . . . 689
<code>markboth</code> . . . . . 74		<code>\ref@Sentence</code> . . . . . 809
<code>markright</code> . . . . . 74		<code>\ref@SentenceX</code> . . . . . 844
<code>noentry</code> . . . . . 411		<code>\refClause</code> . . . . . 12, 705
<code>nohead</code> . . . . . 411		<code>\refClauseN</code> . . . . . 734
<code>notocentry</code> . . . . . 411		<code>\refL</code> . . . . . 681
<code>number</code> . . . . . 432		<code>\refN</code> . . . . . 697
<code>paragraphmark</code> . . . . . 0, 74		<code>\refPar</code> . . . . . 12, 743
<code>parcitename</code> . . . . . 123		<code>\refParagraph</code> . . . . . 209
<code>parnumber</code> . . . . . 51		<code>\refParagraphN</code> . . . . . 209
<code>postskip</code> . . . . . 0, 433		<code>\refParL</code> . . . . . 778
<code>preskip</code> . . . . . 0, 433		<code>\refParN</code> . . . . . 792
<code>ref</code> . . . . . 123		<code>\refParS</code> . . . . . 789
<code>sentencecitename</code> . . . . . 123		<code>\refS</code> . . . . . 689
<code>title</code> . . . . . 411		<code>\refSentence</code> . . . . . 12, 809
<code>tocentry</code> . . . . . 411		<code>\refSentenceL</code> . . . . . 844
		<code>\refSentenceN</code> . . . . . 858
		<code>\refSentenceS</code> . . . . . 855
<b>P</b>		
<code>\p@par</code> . . . . . 621		
<code>\p@sentence</code> . . . . . 861		
<code>\Par</code> . . . . . 12		<b>S</b>
<code>\par</code> . . . . . 12		<code>\scrjura@usetype</code> . . . . . 619
<code>par</code> (Zähler) . . . . . 621		<code>\scrjura@afterheading</code> . . . . . 571
<code>\Paragraph</code> . . . . . 209		<code>\scrjura@doendpe</code> . . . . . 591
<code>\ParagraphCompatibilityHacks</code> 209		<code>\scrjura@env@type</code> . . . . . 256
<code>paragraphmark</code> (Option) . . . . . 0, 74		<code>\scrjura@everypar</code> . . . . . 590
<code>\parcite@fromat</code> . . . . . 123		<code>\scrjura@lang@error</code> . . . . . 903
<code>\parciteformat</code> . . . . . 12, 871		<code>\scrjura@separator</code> . . . . . 618
<code>parcitename</code> (Option) . . . . . 123		<code>\scrjura@special@par</code> . . . . . 674
<code>\parellipsis</code> . . . . . 6, 257		<code>\scrjura@special@reset@par</code> . 674
<code>\parformat</code> . . . . . 12, 621		<code>\scrjura@usetype</code> . . . . . 619
<code>\parlongformat</code> . . . . . 897		<code>\Sentence</code> . . . . . 9
<code>\parname</code> . . . . . 12, 903		<code>\sentence</code> . . . . . 12
<code>parnumber</code> (Option) . . . . . 51		<code>sentence</code> (Zähler) . . . . . 861
		<code>\sentencecite@fromat</code> . . . . . 123



<code>\sentenceciteformat</code> . . . . .	12, 884	<code>\thesentence</code> . . . . .	12, 861
<code>sentencecite</code> name (Option) . . . . .	123	<code>\thisparnumber</code> . . . . .	674
<code>\sentencelongformat</code> . . . . .	897	<code>title</code> (Option) . . . . .	411
<code>\sentencename</code> . . . . .	12, 903	<code>tocentry</code> (Option) . . . . .	411
<code>\sentencenumericformat</code> . . . . .	897	<code>\toclevel@cp</code> ar . . . . .	15
<code>\sentenceshortformat</code> . . . . .	897	<b>U</b>	
<code>\sentenceshortname</code> . . . . .	903	Umgebungen:	
<code>\SubClause</code> . . . . .	6	<code>contract</code> . . . . .	273
<code>\SubParagraph</code> . . . . .	209	<b>Z</b>	
<b>T</b>		Zähler:	
<code>\thecontractClause</code> . . . . .	12, 273	<code>@AbsClause</code> . . . . .	442
<code>\thecontractSubClause</code> . . . . .	12, 273	<code>contractAbsoluteClause</code> . . . . .	273
<code>\theH@AbsClause</code> . . . . .	442	<code>contractClause</code> . . . . .	273
<code>\theHClause</code> . . . . .	442	<code>contractSubClause</code> . . . . .	273
<code>\theHpar</code> . . . . .	621	<code>juratoclevel</code> . . . . .	15
<code>\theHsentence</code> . . . . .	861	<code>par</code> . . . . .	621
<code>\theHSubClause</code> . . . . .	442	<code>sentence</code> . . . . .	861
<code>\theHpar</code> . . . . .	12, 621		

## Change History

1.0	<code>\DeprecatedParagraph</code> : wird noch ein paar Versionen einen Fehler melden . . . . .	18	erweitert beim Schreiben zu seinem Argument . . . . .	30
	<code>\Paragraph</code> : wird noch ein paar Versionen einen Fehler melden . . . . .	18	erstes Argument von <code>\newmaxpar</code> und zweites Argument von <code>\getmaxpar</code> ist <code>contract</code> nicht <code>contractpars</code> . . . . .	30
	<code>\ParagraphCompatibilityHacks</code> : wird noch ein paar Versionen verbleiben . . . . .	18	<code>\getmaxpar</code> : <code>\#2@Clauseformat</code> expandiert zu seinem Argument . . . . .	38
	<code>\refParagraph</code> : wird noch ein paar Versionen einen Fehler melden . . . . .	18	<code>\protected@edef</code> durch <code>\edef</code> ersetzt . . . . .	38
	<code>\refParagraphN</code> : wird noch ein paar Versionen einen Fehler melden . . . . .	18	v0.5d	
	<code>\SubParagraph</code> : wird noch ein paar Versionen einen Fehler melden . . . . .	18	<code>\contract</code> : <code>\jura@env@type</code> wird gesetzt . . . . .	19
v0.5b	<code>juratitlepagebreak</code> : neue Option . . . . .	14	fehlende Fehlermeldung ergänzt . . . . .	19
v0.5c	<code>\contract@everypar</code> : <code>\contract@Clauseformat</code> . . . . .	29	Umgebung definiert sich nicht mehr selbst um . . . . .	19
			<code>ref</code> : neue Einstellungen <code>nopar</code> , <code>nosentence</code> , <code>OnlyParagraph</code> . . . . .	16
			<code>\p@par</code> : <code>\jura@usetyp</code> e mit Argument <code>\jura@env@type</code> eingefügt . . . . .	29

<code>\parciteformat</code> : Argument verschoben . . . . .	36	<code>\newmaxpar</code> : <code>\#1@Clauseformat</code> expandiert zu seinem Argument . . . . .	38
<code>\scrjura@@usetype</code> : neu (intern) . . . . .	29	v0.7	
<code>\scrjura@env@type</code> : neu (intern) . . . . .	19	<code>paragraphmark</code> : Verwendung der renovierten Schnittstelle mit <code>\FamilyKeyStateProcessed</code> . . . . .	15
<code>\scrjura@usetype</code> : neu (intern) . . . . .	29	<code>parnumber</code> : Verwendung der renovierten Schnittstelle mit <code>\FamilyKeyStateProcessed</code> . . . . .	14
<code>\sentenceciteformat</code> : Argument verschoben . . . . .	37	<code>\contract@everypar</code> : Absatznummer von <code>\thisparnumber</code> beachtet . . . . .	30
v0.5e		Im Fall von nicht rein numerischen Absatznummern die Absätze auf jeden Fall nummerieren . . . . .	30
<code>\Clausemark</code> : Beachtung von <code>\MakeMarkcase</code> . . . . .	15	Satznummer bei manuell nummerierten Absätzen auf 0 statt 1 initialisiert, weil <code>\Sentence</code> die Nummer als erstes hochzählt. . . . .	30
v0.6		Zähler <code>contractAbsoluteClause</code> wird verwendet . . . . .	30
<code>parnumber</code> : neue Option . . . . .	14	<code>\contract@paragraph</code> : Abstand nach der Nummer als Bestandteil der Nummer setzen	26
<code>\cpar@indent</code> : neu (intern) . . . . .	14	Zähler <code>contractAbsoluteClause</code> wird erhöht . . . . .	24
<code>\cpar@numberwidth</code> : neu (intern) . . . . .	14	<code>\contract@subparagraph</code> : Abstand nach der Nummer als Bestandteil der Nummer setzen . . . . .	26
<code>\getmaxpar</code> : Argument 3 muss mit <code>\protected@edef</code> expandiert werden . . . . .	38	Zähler <code>contractAbsoluteClause</code> wird erhöht . . . . .	26
<code>\if@juratoc</code> : Schalter durch Zähler überflüssig . . . . .	13	<code>ref</code> : Verwendung der renovierten Schnittstelle mit <code>\FamilyKeyStateProcessed</code> . . . . .	16
<code>\ifparnumber</code> : Definition des Schalters in die Definition der Option verschoben . . . . .	29	<code>\ellipsispar</code> : neu . . . . .	19
<code>entry</code> : Option veraltet . . . . .	23	<code>contractAbsoluteClause</code> : neuer (interner) Zähler über alle Paragraphen, um ggf. die Zählung neu beginnen zu können . . . . .	19
<code>tocentry</code> : neue Option . . . . .	23	<code>\parellipsis</code> : neu . . . . .	19
<code>noentry</code> : Option veraltet . . . . .	23		
<code>notocentry</code> : neue Option . . . . .	23		
<code>juratocindent</code> : neu . . . . .	14		
<code>juratoclevel</code> : Zählerdefinition verschoben . . . . .	13		
<code>juratocnumberwidth</code> : neu . . . . .	14		
v0.6a			
<code>parnumber</code> : Meldung bei falschen Werten korrigiert . . . . .	14		
<code>\contract@paragraph</code> : fehlendes <code>\par</code> ergänzt . . . . .	25		
<code>\contract@subparagraph</code> : fehlendes <code>\par</code> ergänzt . . . . .	26		
v0.6b			
<code>\Clauseformat</code> : <code>\paragraphformat</code> undefiniert . . . . .	19		
<code>\contract@paragraph</code> : Verwendung von <code>\contract@Clauseformat</code> auch bei manueller Nummerierung der Paragraphen. . . . .	25		

<code>\parformat:</code> neues Font-Element <code>parnumber</code> . . . . .	29	<code>\scrjura@@usetype:</code> <code>\jura@@usetype</code> umbenannt . . . . .	29
<code>\scrjura@lang@error:</code> Verwendung von <code>\PackageError</code> an Stelle von <code>\PackageErrorNoLine</code> . . . . .	37	<code>\scrjura@afterheading:</code> <code>\jura@afterheading</code> umbenannt . . . . .	28
<code>\scrjura@special@par:</code> neue Anweisung (intern) . . . . .	31	benötigt kein Argument mehr	28
<code>\scrjura@special@reset@par:</code> neue Anweisung (intern) . .	31	<code>\scrjura@env@type:</code> <code>\jura@env@type</code> umbenannt . . . . .	19
<code>\thisparnumber:</code> neue Anweisung . . . . .	31	<code>\scrjura@everypar:</code> <code>\jura@everypar</code> umbenannt . . . . .	28
<code>\toclevel@cpair:</code> neue für hyperref . . . . .	13	<code>\scrjura@separator:</code> <code>\jura@separator</code> umbenannt . . . . .	29
<code>juratocindent:</code> Verwendung der renovierten Schnittstelle mit <code>\FamilyKeyStateProcessed</code> . . . . .	14	<code>\scrjura@usetype:</code> <code>\jura@usetype</code> umbenannt . . . . .	29
<code>juratoclevel:</code> Verwendung der renovierten Schnittstelle mit <code>\FamilyKeyState</code> . . . . .	13	v0.9	
<code>juratocnumberwidth:</code> Verwendung der renovierten Schnittstelle mit <code>\FamilyKeyStateProcessed</code> . . . . .	14	<code>\@defjuraenvironment:</code> Neue (interne) Anweisung . . . . .	21
v0.7a		<code>\DeclareNewJuraEnvironment:</code> Neue Anweisung . . . . .	20
<code>\contract@sentence:</code> <code>\nobreak\hskip\z@</code> eingefügt, um die Trennung des ersten Worts nach der Satzmarkierung zu ermöglichen . . . . .	36	v0.9a	
auf <code>\textsuperscript</code> umgestellt . . . . .	36	<code>paragraphmark:</code> Wert wird in der internen Liste erfasst . .	15
v0.7b		<code>parnumber:</code> Wert wird in der internen Liste erfasst . . . . .	14
General: Generelle Umbenennung von „Paragraph“ in „Clause“ für alle Befehle, Zähler und Optionen . . . . .	12	<code>ref:</code> Wert wird in der internen Liste erfasst . . . . .	16
<code>\contract@everypar:</code> <code>contractAbsoluteClause</code> flexibelisiert . . . . .	30	<code>juratocindent:</code> Wert wird in der internen Liste erfasst . .	14
<code>\contract@paragraph:</code> <code>contractAbsoluteClause</code> flexibelisiert . . . . .	24	<code>juratoclevel:</code> Wert wird in der internen Liste erfasst . . . . .	13
<code>\ellipsispar:</code> <code>\thecontractAbsoluteClause</code> flexibelisiert . . . . .	19	<code>juratocnumberwidth:</code> Wert wird in der internen Liste erfasst . . . . .	14
<code>\getmaxpar:</code> Notfallcode in die aux-Datei . . . . .	38	v0.9b	
		<code>contractSubClause:</code> muss unbedingt den Zähler <code>SubClause</code> statt <code>contractSubClause</code> verwenden . . . . .	19
		<code>\thecontractClause:</code> muss unbedingt den Zähler <code>Clause</code> verwenden . . . . .	19
		<code>\@defjuraenvironment:</code> <code>\the...Clause</code> definiert . .	21
		<code>\contract@paragraph:</code> <code>\thecontractClause</code> durch umgebungsabhängiges Macro ersetzt. . . . .	25