LATEX Package contract*

Markus Kohm

Version v0.91 of 2024-02-07

The contract package is intended for cautelar jurisprudence. It is intended to provide flexible help for lawyers and notaries in drafting contracts, statutes and legal commentaries. It has been developed in cooperation with Dr Alexander Willand, and is still in the process of development.

Package contract is a KOMA-Script spin-off. It has been released from 2011 till 2023 as scrjura. With KOMA-Script 3.42 scrjura is removed from KOMA-Script and released as new package contract.

Contents

Us	ser Manual	2
1	Package Loading and Option Setting	3
2	Changing the Fonts of Elements	3
3	Clauses in the Table of Contents	4
4	Environment for Contracts 4.1 Clauses	5 7 9
5	Cross References	9
6	Additional Contract Environments	12
7	Support for Different Languages	12
8	A Detailed Example	14
9	From scrjura to contract	18
10	State of Development	20

^{*}The repository of this package can be found at https://github.com/komascript/latex-contract where you also should report issues.

mplementation	20
11 Cooperation with hyperref	20
12 Prerequisites	20
13 Options	21
14 Deprecated Paragraph Commands	26
15 Contracts, Clauses, Paragraphs and Sentences	27
16 Entry to Table of Contents	37
17 Numbering of Paragraphs and Sentences	38
18 Referencing	40
19 Language Dependent Names	46
20 Using Values from last LATEX Run	47
References	48
Change History	48
ndex	52
List of Tables	
Available properties for the optional argument of \Clause and \SubClause . Available values for the clausemark option to activate running heads	7
List of Figures	
1 The four pages of the CfCH example of section 8	19

User Manual

If you want to write a contract, the articles of association for a company or an association, a law, or a legal commentary, the package contract will provide typographical support.

Although contract is intended to provide general help for legal documents, the contract is the central element of the package. Particular attention is paid to clauses, titles, and numbered provisions—if there are several of them in a clause—, numbered sentences, entries in the table of contents, and cross references according to German standards.

The package has been developed in cooperation with Dr Alexander Willand of Karlsruhe. Many of its features go back to constructive inquiries from Prof Heiner Richter of the Hochschule Stralsund University of Applied Sciences.

Some of you may search for the German user manual formally available in [Koh23a]. The author's acute overload has resulted in this no longer being freely available. However, the corresponding chapter in [Koh20a] and [Koh20b] is still accessible and applicable with appropriate adjustments to the package name.

1 Package Loading and Option Setting

You can load the package as common using:

```
\usepackage[\langle options \rangle] \{contract\}
```

In case of package contract the $\langle options \rangle$ are $\langle key \rangle = \langle value \rangle$ options.

Note: In case of using package contract with package hyperref you should load hyperref always after contract.

\contractSetup

To change options after loading the package, you should use:

```
\contractSetup{\langle options \rangle}
```

with $\langle options \rangle$ is a comma separated list of $\langle key \rangle = \langle value \rangle$ options as explained before.

Note: Currently the options are still implemented using the internal KOMA-Script package scrkbase. Therefore you could also setup \(options \)\) using the KOMA-Script commands \(\cong \cong

2 Changing the Fonts of Elements

Currently package contract still uses the font selection features of the internal KOMA-Script package scrkbase. So the commands \setkomafont, \addtokomafont, and \usekomafont, which are described in [Koh23b], can be used to change the fonts of the following elements:

Clause:

Alias for \(\lambda environment \) Clause within any contract environment, e.g., contract. Clause within environment contract. If no corresponding element is defined, contract. Clause is used.

contract.Clause:

(Default: \sffamily\bfseries\large)

The heading of a **\Clause** within the environment contract.

 $\langle environment \rangle$.Clause:

(Default: none)

The heading of a \Clause within environment \(\lambda environment \rangle , \) which has been defined using \\ \DeclareNewJuraEnvironment, if the font has been setup using property \(ClauseFont \) or the element has been defined explicitly.

parnumber: (Default: empty)

The paragraph number within a contract environment.

sentencenumber: (Default: empty)

The sentence number printed by \Sentence.

Note: There are plans to no longer use package scrkbase, but maybe package scrextend instead, if scrjura is used with a non-KOMA-Script class. So the commands \setkomafont, \addtokomafont and \usekomafont would still be available in future.

3 Clauses in the Table of Contents

The headings of clauses can also be added automatically to the table of contents, if desired. For this the package uses command \DeclareTOCStyleEntry of KOMA-Script package tocbasic to define an entry level named cpar. Usage of package tocbasic also means, that you should not use another package to configure the Table of Contents, e.g., tocloft, tocstyle etc.

juratotoc (opt.)

Clauses are shown in the table of contents only if their level number is less than or equal to the tocdepth counter. By default, the level number is \maxdimen, which is also used if the option is switched off using juratotoc=false. Because the tocdepth counter usually has a one-digit value, clause entries are therefore not normally displayed in the table of contents.

If you switch on the option using the <u>juratotoc=true</u>, the level number 2 is used so that clauses are shown in the table of contents on the same level as subsections. For the default setting of tocdepth, clauses are then shown in all KOMA-Script classes or standard classes.

You can also use juratotoc= $\langle integer \rangle$ to use $\langle integer \rangle$ instead of \maxdimen or 2 as level number.

Internally usage of this option results in a call of

```
\label{lem:lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma:level=lemma
```

respectively

\DeclareTOCStyleEntry[level=2]{default}{cpar}

respectively

```
\DeclareTOCStyleEntry[level=\maxdimen]{default}{cpar}
```

 $\label{eq:continuous} \mbox{juratocindent } (\mbox{\it opt.}) \\ \mbox{juratocnumberwidth } (\mbox{\it opt.}) \\$

These options determine the indentation and spacing for clause entries in the table of contents. Any valid $\langle length \rangle$ can be assigned. The defaults are the same as for subsection entries in scrartcl.

Internally, usage of these options results in calls to:

```
\label{lem:default} $$ \DeclareTOCStyleEntry[indent=\langle length\rangle] $$ {\bf default} {\bf cpar}$
```

respective

 $\DeclareTOCStyleEntry[numwidth=\langle length\rangle] \{default\}\{cpar\}$.

4 Environment for Contracts

The essential mechanisms of package contract are available only inside contract environments, either the predefined contract or any other environment defined with command \DeclareNewJuraEnvironment.

contract

Currently, this is the one and only predefined environment for contract. Using it activates automatic numbering of paragraphs and the \Clause and \SubClause commands, which will be documented below, are given concrete form.

The contract environment must not be nested within itself. Within a document, however, you can use the environment several times. The clauses within these environments are treated as if they were within a single environment. As a result, ending the environment really only temporarily interrupts it, and the old environment is continued by the beginning of a new environment. However, you cannot end the environment within a clause. If you want instead print several contracts, you would need to define several contract environments using command \DeclareNewJuraEnvironment.

contract (opt.)

The whole document becomes a contract if you use this option while loading the package with:

\usepackage[contract]{contract}

or as a global option with \documentclass. The document then behaves exactly as if it would contain one contract environment.

4.1 Clauses

Clauses in a legal sense are defined in package contract only within contracts, that is inside the contract environment or other environments declared with \DeclareNewJuraEnvironment.

optional argument $\langle property | list \rangle$ is a comma separated list of $\langle key \rangle = \langle value \rangle$.

Each clause starts either with $\clause[(property\ list)]$ or $\clause[(property\ list)]$. The \Clause \SubClause

These are the most important commands inside of a contract. Without using any additional $\langle key \rangle$, Clause creates the heading of a clause, which consists of the sign "\{\xi\}", followed by its number. In contrast, \SubClause creates the heading of a clause with the last number used by \Clause and adds a lower-case letter. \SubClause is mainly intended for cases where an act or a contract is amended and not only are clauses changed or deleted but new clauses are inserted between existing ones without completely changing the numbering.

Both commands accept a comma-separated list of $\langle key \rangle = \langle value \rangle$ properties and also some $\langle key \rangle$ s without value. An overview of the available properties is shown in Table 1. The most important of them will be discussed in more detail.

By default, a skip of two lines is inserted before the heading and a skip of one line afterwards. You can change the size of these skips with the preskip and postskip properties. The new values apply not only to the current clause but from the current clause until the end of the current contract environment. You can also make the appropriate settings in advance with

 $\strut {contract} {preskip=\langle skip \rangle, postskip=\langle skip \rangle}$

5

¹In English, a "clause" in a legal document is a section, paragraph, or phrase that relates to a particular point. Although it is common in English to also use the terms "article" or "section" for what we here call a "clause", we use the latter term throughout to avoid confusion with the article class and the \section and \paragraph sectioning divisions of most document classes.

Table 1: Available properties for the optional argument of \Clause and \SubClause

dummy	The heading will not be printed but is counted in the automatic numbering.
$\texttt{head=}\langle running\ head\rangle$	If running heads are active, this $\langle running\ head \rangle$ is used instead of the clause $\langle title \rangle$.
nohead	The running head stays unchanged.
notocentry	Does not make an entry into the table of contents.
${\tt number=} \langle number \rangle$	Uses $\langle number \rangle$ for the output of the clause number.
$\texttt{preskip=}\langle skip\rangle$	Changes the vertical $\langle skip \rangle$ before the clause heading.
${\tt postskip=}\langle skip\rangle$	Changes the vertical $\langle skip \rangle$ after the clause heading.
$\mathtt{title=}\langle\mathit{title}\rangle$	The clause $\langle title \rangle$ will be printed in addition to the clause number. This is also used as the default for the $\langle running\ head \rangle$ and the $\langle entry \rangle$ in the table of contents.
$\texttt{tocentry=}\langle \mathit{entry}\rangle$	Regardless of the clause $\langle title \rangle$, an $\langle entry \rangle$ into the table of contents will be made, if such entries are activated (see option <code>juratotoc</code>).

regardless of the specific clause and outside of a contract environment. You can also set these options inside the preamble after loading contract, but you cannot set them while loading the package or by using \contractSetup.

By default, clause headings use the font style \sffamily\bfseries\large. See section 2 for information about how to change the font for element contract.Clause.

With the title, head, and tocentry property, you can title a clause in addition to the number. You should enclose the $\langle value \rangle$ of each property inside curly brackets. Otherwise, for example, commas which are meant to be part of the $\langle value \rangle$ will be confused with the delimiters between different properties of the $\langle property | list \rangle$. Empty values for head and tocentry cause empty entries. If you want to avoid an entry, use the nohead resp. notocentry property.

Instead of consecutive numbers, you can also set a clause number manually with the number property. However, this does not affect the numbers of the subsequent clauses. Empty numbers are not possible. Fragile commands inside number have to be protected with \protect. You should use only numbers and letters as a number.

With the dummy property, you can suppress the output of the whole heading of a clause. The automatic numbering, however, will still count this clause. In this way, you can skip an automatically numbered clause with

\Clause[dummy]

in case the clause corresponding clause has been deleted in a later version of a contract.

Note that the dummy property only accepts the $\langle value \rangle$ s true and false. All other $\langle value \rangle$ s are usually ignored, but can lead to an error message in the worst case scenario.

As already mentioned, clauses and subclauses are normally numbered. The number is formatted with the help of the \Clauseformat command, which expects the $\langle number \rangle$ as the only argument. The default is the following:

\newcommand*{\Clauseformat}[1]{\S #1}

\Clauseformat

Table 2: Available values for the clausemark option to activate running heads

both	Clauses generate left and right marks for running heads, if the document provides automatic running heads.
false, off, no	Clauses do not generate marks for running heads and therefore do not change running heads.
forceboth	Clauses use \markboth to generate left and right marks for running heads even if the document does not provide automatic running heads for the current page style.
forceright	Clauses use \markright to generate right marks for running heads even if the document does not provide automatic running heads for the current page style.
right	Clauses generate right marks for running heads, if the document provides automatic running heads.

This produces the section mark, \S (§), followed by a non-breaking space and the number. If you redefine this command, be sure it remains expandable.

juratitlepagebreak (opt.)

Usually, page breaks are prohibited within heading of all kinds. However, some lawyers require page breaks within clause headings. You can allow such a break by using option juratitlepagebreak or juratitlepagebreak=(boolean). Boolean values of true or false can be used to toggle the option on or off. Using the option without a value is the same as using true. The option can be used as an optional argument of \documentclass, \usepackage when loading package contract, or as an argument of \contractSetup.

clausemark (opt.)

Since clauses are a subordinate structure with independent numbering, they do not produce running heads by default. You can, however, create running heads with various settings using:

```
clausemark = \langle value \rangle
```

as an optional argument of \documentclass, \usepackage when loading package contract, or as an argument of \contractSetup. You can find the available $\langle value \rangle$ s and their meanings in Table 2.

4.2 Paragraphs

Within clauses, contract usually numbers paragraphs automatically. With this, the paragraphs provide a powerful structuring element, similar to \paragraph or \subparagraph in normal documents. For this reason, contracts usually use a vertical skip between paragraphs. The contract package does not provide its own mechanism for this. Instead, you should uses the parskip option of the KOMA-Script classes. If you do not use a KOMA-Script class, see the documentation of the used class or package parskip [Mit21].

parnumber (opt.)

The option can be used as an optional argument of \documentclass, \usepackage when loading package contract, or as an argument of \contractSetup.

The default numbering of paragraphs is parnumber=auto and parnumber=true. Sometimes you may need to disable the automatic numbering. You can do this with parnumber=false. In this case, only the sentence numbering is reset.

To implement this option, it has been necessary to hook into the paragraph-building mechanism of LATEX. In some rare cases, this can have a negative effect. If so, you can undo the change with parnumber=manual. On the other hand, LATEX itself sometimes undoes the change. In those cases you can activate it again with parnumber=auto.

Clauses that consist of a single paragraph do not automatically receive a paragraph number. For this to work, there must not be two clauses with an identical number in a document. However should you ever need such numbering, you should switch to another contract environment (see \DeclareNewJuraEnvironment). Note that the number of paragraphs in a clause is not available before the end of the clause. Therefore you need a least two LATEX runs before the automatic paragraph numbering is correct.

 For numbering the paragraphs inside a clause we use the par counter. The output of \thepar will display an Arabic number, because the default is \arabic{par}. \parformat provides the format, which is \thepar in rounded brackets. When numbering a paragraph manually, you should also use \parformat. It makes sense to call \parformat with a subsequent \parformatseparation, or at least a \nobreakspace or tilde.

With automatic numbering, \parformat is followed by \parformatseparation, which currently consists of \nonbreakspace, the non-breakable space.

The paragraph number is usually printed using the currently active font. See section 2 for information how to change the font of the parnumber element.

Note: contract currently assumes internally that **\thepar** is an Arabic number. Therefore you should definitely not redefine it!

\withoutparnumber

If the paragraph number is not printed, contract executes the \withoutparnumber command at the beginning of the new paragraph. The initial definition of this command is empty. This means it is a kind of dummy command that does nothing. It has been implemented because of a user request. Most users can ignore this command.

\ellipsispar \parellipsis

Sometimes—particularly in comparative commentaries—it is desirable to omit paragraphs but to mark the omission. Those omitted paragraphs should be taken into account by the paragraph counter. The package contract provides the command \ellipsispar to do this.

By default, \ellipsispar omits precisely one paragraph. Using the optional argument of

```
\left( \left( number \right) \right)
```

you can omit multiple paragraphs. In any case, the output shows just one unnumbered paragraph, which consists only of the ellipsis defined by **\parellipsis**. The automatic numbering of paragraphs takes the $\langle number \rangle$ of omitted paragraphs into account.

Example: Suppose you are writing a comment on the German² penal code, but only on paragraph 3 of § 2. Nevertheless, you'd like to indicate the omission indirectly. You can do this with:

```
\documentclass[parskip=half]{scrartcl}
\usepackage{contract}
```

²Please remember, this translation does not refer to an existing law but is only an example of how you might realise such a commentary with contract.

```
\ellipsispar[2]
```

If the law that applies at the time the criminal act is committed is changed before the verdict, then the most lenient law shall be applicable.

```
\ellipsispar[3]
\end{contract}
\end{document}
```

To see the result, just give it a try.

The ellipsis is by default \textellipsis, if such a command is defined. If not, \dots is used. You can redefine \parellipsis at any time with \renewcommand.

4.3 Sentences

Paragraphs in contracts consist of one or more sentences, some of which may also be numbered. However, as automatic numbering is cumbersome and error-prone, it has not yet been implemented in contract. Semi-automatic numbering, however, is supported.

Manual numbering of sentences is done with the \Sentence command. It adds one to the sentence counter. By default, \sentencenumberformat prints \thesentence as an Arabic number in superscript.

The sentence number is usually printed using the currently active font. See section 2 for information how to change the font of the sentencenumber element.

Using babel offers an easy way to define a shorthand for \Sentence:

```
\useshorthands{'}
\defineshorthand{'S}{\Sentence\ignorespaces}
```

With this definition, any space after 'S will be ignored. You can even use the dot as an abbreviation for a dot and a new sentence number:

```
\defineshorthand{'.}{. \Sentence\ignorespaces}
```

For details regarding \useshorthands and \defineshorthand, please consult the manual of the babel package (see [BB24]). You can find an example of their application in section 8, page 14.

5 Cross References

The conventional mechanism to set cross references using \label, \ref, and \pageref does not suffice for clauses, paragraphs, and sentences. Therefore contract provides additional commands.

give a full reference to clause, paragraph and sentence. \refL is a long text, \refS a short text, and \refN an abbreviated, numeric form. \ref defaults to \refL.

\refClause \refClauseN

The commands

 $\begin{tabular}{ll} & sentence \ (cnt.) \\ & \begin{tabular}{ll} & the sentence \\ sentence number format \\ & \begin{tabular}{ll} & sentence \\ & \begin{tabu$

```
\refClause{\langle label \rangle} \refClauseN{\langle label \rangle}
```

reference a clause without displaying the paragraph or sentences. \refClause puts a section mark (§) in front of the reference, while \refClauseN does not.

\refPar \refParL \refParS \refParN

```
The commands
```

```
\label{label} $$ \operatorname{ParL}(\langle label \rangle) $$ \operatorname{ParL}(\langle label \rangle) $$ \operatorname{ParS}(\langle label \rangle) $$ \operatorname{ParN}[\langle number\ format \rangle] {\langle label \rangle} $$
```

reference a paragraph of a clause. The differences between the forms correspond to the differences between \refl , \refl and \refl . A feature worth noting is the optional argument of \refl Usually the numeric reference to a paragraph uses a Roman number. You can, however, specify a different $\langle number\ format \rangle$ in the optional argument. This option primarily makes sense to use Arabic numbers. By default, \refl is \refl as \refl and \refl is \refl and \refl are \refl and \refl are \refl and \refl are \refl are \refl and \refl are \refl and \refl are \refl and \refl are \refl are \refl and \refl are \refl and \refl are \refl and \refl are \refl are \refl and \refl are \refl and \refl are \refl and \refl are \refl are \refl are \refl and \refl are \refl and \refl are \refl and \refl are \refl are \refl and \refl are \refl are \refl and \refl are \refl and \refl are \refl are \refl are \refl are \refl are \refl and \refl are \refl are

\refSentence \refSentenceS \refSentenceN

```
The commands
```

```
\label{local_section} $$\operatorname{Sentence}_{\langle label\rangle}$$ \operatorname{Sentence}_{\langle label\rangle}$$ \operatorname{Sentence}_{\langle label\rangle}$$ \operatorname{Sentence}_{\langle label\rangle}$$
```

reference a sentence of a paragraph. Again, there is a long text form, a short text form, and a numerical form. By default, \refSentence is \refSentenceL.

ref (opt.)

```
The results of \ref, \refPar, and \refSentence depend on the option
```

```
ref=(value)
```

that can be used as an optional argument of \documentclass, \usepackage when loading package contract, or as an argument of \contractSetup. The default is ref=long and therefore \refL, \refParL and \refSentenceL. You can find the available \(\lambda value \rangle \)s for his option and their meaning in Table 3.

Example: Suppose you always want to reference paragraphs in the form "paragraph 1 in clause 1". As there is no predefined command for this, you have to create your own definition from the available options. You can achieve this easily with:

```
\newcommand*{\refParM}[1]{%
  paragraph~\refParN[arabic]{#1}
  in clause~\refClauseN{#1}%
```

This new command can be used in the same way as \refParL.

You can find examples of results of the basic commands—this means the commands, that are independent from option **ref**—in Table 4.

Table 3: Available values for the ref option to configure the cross reference format of \ref, \refPar, and \refSentence

long	A combination of parlong and sentencelong.	
numeric	A combination of parnumeric and sentencenumeric.	
<pre>clauseonly, onlyClause ClauseOnly, OnlyClause</pre>	A combination of paroff and sentenceoff. Note that \refPar and \refSentence produce empty results!	
<pre>parlong, longpar, ParL</pre>	Paragraphs are referenced in long textual form.	
parnumeric, numericpar, ParN	Paragraphs are referenced in simple numerical form.	
paroff, nopar	Paragraphs have no reference. Note that \refPar produces an empty result!	
parshort, shortpar, ParS	Paragraphs are referenced in short textual form.	
sentencelong, longsentence, SentenceL	longsentence, Sentences are referenced in long textual form.	
$\begin{tabular}{ll} \tt sentence numeric, numeric sentence, \\ \tt Sentence N \end{tabular}$	Sentences are referenced in simple numeric form.	
sentenceoff, nosentence	Sentences have no reference. Note that \refSentence produces an empty result!	
sentenceshort, shortsentence, SentenceS	Sentences are referenced in short textual form.	
short	A combination of parshort and sentenceshort.	

Table 4: Example outputs of the ref-independent cross reference commands

Command	Example output
$\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $	§ 1 paragraph 1 sentence 1
$\mathbf{refS}\{\langle label\rangle\}$	§ 1 par. 1 sent. 1
$\mathbf{refN}\{\langle label \rangle\}$	§ 1 I 1.
$\rdet{refClause} \{\langle label \rangle\}$	§ 1
$\rdet{refClauseN}\{\langle label \rangle\}$	1
$\mathbf{refParL}\{\langle label \rangle\}$	paragraph 1
$\mathbf{refParS}\{\langle label \rangle\}$	par. 1
\r efParN $\{\langle label angle\}$	I
\r [arabic] { $\langle label \rangle$ }	1
$\rdet{refParN}$ [roman] $\{\langle label \rangle\}$	i
\r	sentence 1
$\rdet{refSentenceS}(\langle label \rangle)$	sent. 1
$\rack \rack \rac$	1.

6 Additional Contract Environments

Some users do not use contract to draft contracts or commentaries on individual laws but to examine different types of laws, which may not necessarily use the section prefix (§) before the title of each clause but perhaps something like "Art." or "IAS", and so forth. An independent counter is also required for each of these different clause types.

\DeclareNewJuraEnvironment

You can use:

to define new and independent environments for contracts or other legal texts. The argument $\langle environment \rangle$ is the name of the new environment, of course. The $\langle start\ commands \rangle$ are commands which will be executed at the beginning of the environment, as if they were added directly after $\langle begin\{\langle environment \rangle\}$. Correspondingly $\langle end\ commands \rangle$ will be executed at the end of the environment, as if added directly before $\langle end\{\langle environment \rangle\}$. Without a $\langle property\ list \rangle$ the new environment behaves like the contract environment, but with its own counters. You can use several $\langle key \rangle = \langle value \rangle$ properties as a comma-separated $\langle property\ list \rangle$. See Table 5 for the currently supported $\langle options \rangle$.

Example: To define the environment for articles we mentioned in the preface of this section, it is sufficient to write:

```
\DeclareNewJuraEnvironment{Article}[ClauseNumberFormat=Art.~]{}{}
```

If we are using a KOMA-Script class and want to separate the paragraphs in this environment with space instead of using paragraph indentation, we can use:

In cross references, "Art." will of course be used instead of "§".

The new environment is used like contract:

```
\begin{Article}
  \Clause{}
  Human dignity is inviolable. To respect and protect people is a
  duty of all state authority.
\end{Article}
```

7 Support for Different Languages

The contract package has been developed in cooperation with a German lawyer. Therefore it initially supported only the languages german, ngerman, austrian, and naustrian. Nevertheless, it has been designed to support common language packages like babel. Users can easily make changes by using \providecaptionname. If you have definitive information about the correct legal terms and conventions of a language, please contact the KOMA-Script author. Support for English has been added in this way, and so contract now also provides terms for the languages english, american, british, canadian, USenglish, and UKenglish.

\parname \partshortname \sentencename \sentenceshortname These are the language-dependent terms used by contract. The meaning of the terms and their English defaults are shown in Table 6. The package itself defines them by using

Table 5: Properties provided by \DeclareNewJuraEnvironment for new contract environments

Clause= $\langle command \rangle$	Defines the $\langle command \rangle$ to which the \Clause command
,	is mapped within the environment. This $\langle command \rangle$,
	like the one documented for contract, expects exactly
	one argument. To use it correctly requires advanced knowledge of the contract's internal functioning. Fur-
	thermore, the requirements for the $\langle command \rangle$ may
	change in future versions. Therefore it is recommended not to use this option!
$\texttt{ClauseFont=}\langle commands \rangle$	If this property is used, a new \(\langle environment \rangle \). Clause
oranger one (communical)	element is defined with the $\langle commands \rangle$ used as its default setting. If the element was previously defined
	as an alias, it will become an independent element instead. If it has already been defined as an independent
	element, the $\langle commands \rangle$ are used as new font settings. Please note the limitations for font settings in section 2.
${\tt SubClause=}\langle command \rangle$	Defines the $\langle command \rangle$ to which the \SubClause com-
	mand is mapped within the environment. This $\langle command \rangle$, like the one documented for contract, ex-
	pects exactly one argument. To use it correctly re-
	quires advanced knowledge of the contract's internal
	functioning. Furthermore, the requirements for the $\langle command \rangle$ may change in future versions. Therefore
	it is recommended not to use this property!
$\texttt{Sentence=}\langle command \rangle$	Defines the $\langle command \rangle$ to which the \Sentence is mapped within the environment. This $\langle command \rangle$
	must not have an argument. Typically it should add
	one to the sentence (using \refstepcounter) counter and display it appropriately. It is particularly impor-
	tant to avoid adding unwanted spaces.
${\tt ClauseNumberFormat=} \langle command \rangle$	Formats the numbers of clauses within the environment. The $\langle command \rangle$ should expect exactly one ar-
	gument: the number of the clause. If the $\langle command \rangle$
	implements a series of commands and the number is
	the last argument of a that series, you can directly use the series of commands as $\langle command \rangle$.

Table 6: Meanings and English defaults of languagedependent terms, if not already defined

Command	Meaning	Default
\parname \parshortname \sentencename \sentenceshortname	long form "paragraph" short form "paragraph" long form "sentence" short form "sentence"	paragraph par. sentence sent.

\providecaptionname inside \begin{document} only if other requirements have not already been met. If you use contract with an unsupported language, the package will throw an error.

8 A Detailed Example

You may remember the letter from the scrittr2 chapter of the KOMA-Script manual [Koh23b], in which a club member wanted to remind the board about an overdue meeting that was prescribed by the club's by-laws. Such club by-laws are a kind of contract, and you can create them using contract.

```
\documentclass[fontsize=12pt,parskip=half]
{scrartcl}
```

We use class scrartcl. Because paragraph distance instead of paragraph indentation is usual in club by-laws, we load the class with option parskip=half.

```
\usepackage[british]{babel}
```

The club rules are in British English. Therefore we load the babel package with the british option too.

```
\usepackage[T1]{fontenc}
\usepackage{lmodern}
```

We make some default font settings. Earlier versions of the example also loaded the <code>textcomp</code> package here for an improved section mark (§). Since LATEX 2020/02/01, however, the desired functionality is directly integrated in the LATEX kernel.

```
\usepackage{enumerate}
```

Later in the document, we want lists numbered not with Arabic numbers but with lower-case letters. We can do this easily with the enumerate package. Alternatively, we could have used the enumitem package.

\pagestyle{myheadings}

Now it is time for contract. The clausemark=forceboth option forces clauses to create left and right marks for the running head. On the other hand, we do not want \section to

change the marks for the running head. Therefore we use the myheadings page style. This page style generally does not provide automatic running heads.

Later, we also want a table of contents with the clauses. This can be achieved with the <code>juratotoc</code> option. Doing so we will see that the default width for these numbers is insufficient for the clause numbers in the table of contents. With <code>juratocnumberwidth=2.5em</code>, we reserve more space.

The definition of shorthands has already been explained in section 4.3. In this example we do the same thing to simplify the input.

```
\begin{document}
```

It is time to begin the actual document.

```
\subject{By-Laws}
\title{CfCH}
\subtitle{Club for Club Hoppers}
\date{11.\,11.\,2011}
\maketitle
```

Like other documents, the by-laws have a title. We created it with the usual **KOMA-Script** commands.

```
\tableofcontents
```

As already mentioned, we want to create a table of contents.

```
\addsec{Preamble}
```

```
The club landscape in England is diverse. But we have unfortunately been forced to conclude that it often suffers seriously when dealing with seriousness.
```

Preambles are not unusual in club by-laws. Here we use \addsec to create one because we want it to have an entry in the table of contents.

```
\appendix
```

Here we use a small trick. The articles of the club by-laws should be numbered with uppercase letters instead of Arabic numbers, just as the appendix sections of an article using scrartcl are.

```
\section{Overview}
\begin{contract}
```

We begin the contract with the first article.

```
\Clause[title={Name, Legal Form, Headquarters}]
```

```
The name of this club shall be the ''Club for Club Hoppers'' and is not registered in any club register.
```

'S The club is a non-economic, useless club'. It has no headquarters because its members heads are in their hindquarters.

```
The fiscal year is from March 31st through April 1st.
```

The first clause has a number and a title. We will do the same with all following clauses.

The first paragraph of the clause contains nothing unusual. Because it is not the only paragraph, every paragraph will be automatically preceded by a paragraph number. Note that the numbering the first paragraph requires at least two LATEX runs. Since this is the case for the table of contents as well, this does not create any additional problems.

In the second paragraph we have two sentences. Here we can see the shorthands 'S and '. in action. The first one only generates the sentence number. The second one generates not only the full stop but also the sentence number. With this, both sentences are numbered.

```
\Clause[title={Purpose of the Club}]

'S The club is pointless but not useless'. Rather,
it should put the serious handling of seriousness on a
sound footing.

For this purpose, the club members can
\begin{enumerate}[\qquad a)]
\item pick their noses,
\item crack nuts,
\item such their thumbs.
\end{enumerate}

The club is selfish and stands by it.

The club has no financial means.\label{a:mittel}
```

The second clause: again this contains several paragraphs, some of which include several sentences. The second paragraph also has a numbered list. In the last paragraph, we set a label, because we want to reference it later.

```
\Clause[title={Club Officers}]
The club officers hold honorary positions.
'S If the club had resources (see \ref{a:mittel}), it could afford a full-time manager'. Without the necessary funds, this is not possible.
```

The third clause contains something special: a cross reference. Here we use the long form with clause, paragraph, and sentence. If we decided later that sentences should not be included in the reference, we could use the ref=nosentence option to set this globally.

```
\Clause[title={Club Hopper},dummy]
\label{p.maier}
```

Here we have a special kind of clause. In earlier versions of the club by-laws, this was a real clause, but it was later removed. However, the numbering of the following clauses should not be changed by removing this one. Therefore the **\Clause** statement has not been removed but supplemented by **dummy** property. With this, we also can set a label even though the clause will not be printed.

```
\end{contract}
\section{Membership}
\begin{contract}
```

Another article begins. To avoid problems with the paragraph numbering, we interrupt the **contract** environment.

```
\Clause[title={Types of Members},dummy]
```

The first clause of the next article also has been deleted.

```
\Clause[title={Becoming a Member}]
```

Everyone can purchase a membership from one of the associations listed in \refClause{p.maier}.\label{a.preis}

'S To become a member, an informal application is required'. This application should be submitted in green ink on pink paper.

Membership applications cannot be rejected.

Here we have a real clause again. We cross reference one of the deleted clauses and also set a label.

```
\SubClause[title={Amendment to the Previous Clause}]
```

```
'S With the repeal of \refClause{p.maier}, \ref{a.preis} has become impractical'. In its place, memberships can be inherited.
```

Once more, this is a special kind of clause. This time we have not removed a clause but added one without renumbering the following clauses. To do so, we use \SubClause. Therefore the clause number is the same like the previous one but with an appended "a".

```
\Clause[title={Termination of Membership}]
```

'S Membership ends with one's life'. For non-living members, membership does not end.

```
\Clause[title={General Meeting}]
```

A general meeting shall take place twice per year.

The interval between two general meetings shall be no more than 6-months, 1-week, and 2-days.

The invitation to the next general meeting shall be sent no earlier than 6-months from the previous general meeting.

\SubClause[title={Amendment to the General Meeting}]

The general meeting may be held at the earliest 2~weeks after the invitation is received. \end{contract}

The other clauses of this article are very usual. You already know all the features used for them.

```
\section{Validity}
```

```
\begin{contract}
\Clause[title={Effective Date}]

These articles will enter into force on 11.\,11.\,2011 at 11:11~am.

'S If any provision of these by-laws is in conflict with any other, the by-laws will be repealed on 11.\,11.\,2011 at 11:11~am and 11~seconds'. The club is considered to be dissolved in this case.
\end{contract}
```

There follows another article no special features.

```
\end{document}
```

Then the LATEX document ends. You can see first three pages in Figure 1.

9 From scrjura to contract

If you have been using the scrjura package and are now switching to the contract package, there are a few things you need to be aware of:

• Instead of

use \contractSetup.

- Instead of using \KOMAoptions or \KOMAoption to set \langle options \rangle for the contract package,
- The \Clause and \SubClause commands now have an optional argument instead of a mandatory argument. This makes more sense because it is allowed to use \Clause or \SubClause without an argument. Instead of braces, the list of properties must be given in square brackets, for example:

```
\Clause[title={title of clause}]
```

Therefore, even more care should be taken to ensure that the values of each property are enclosed in braces. Otherwise, there may be problems not only with commas, but also with square brackets in the values.

Check your code if you have used or redefined any internal macros as several of them
have been renamed.

A. Overview By-Laws § 1 Name, Legal Form, Headquarters **CfCH** (1) The name of this club shall be the "Club for Club Hoppers" and is not registered in any club register. Club for Club Hoppers (2) 1 The club is a non-economic, useless club. 2 It has no headquarters because its members heads are in their hindquarters. (3) The fiscal year is from March 31st through April 1st. 11.11.2011 § 2 Purpose of the Club Contents (1) $^1{\rm The}$ club is pointless but not useless. $^2{\rm Rather},$ it should put the serious handling of seriousness on a sound footing. (2) For this purpose, the club members can
a) pick their noses,
 A. Overview
 2

 § 1. Name, Legal Form, Headquarters
 2

 § 2. Purpose of the Club
 2

 § 3. Club Officers
 2
 b) crack nuts, B. Membership

§ 6. Becoming a Member

§ 6a. Amendment to the Previous Clause

§ 7. Termination of Membership

§ 8. General Meeting

§ 8a. Amendment to the General Meeting (3) The club is selfish and stands by it. (4) The club has no financial means. § 3 Club Officers (1) The club officers hold honorary positions. (2) 1 If the club had resources (see § 2 paragraph 4 sentence 1), it could afford a full-time manager. 2 Without the necessary funds, this is not possible. Preamble The club landscape in England is diverse. But we have unfortunately been forced to conclude that it often suffers seriously when dealing with seriousness. § 6 Becoming a Member § 8a Amendment to the General Meeting B. Membership § 8a Amendment to the General Meeting The general meeting may be held at the earliest 2 weeks after the invitation is received § 6 Becoming a Member (1) Everyone can purchase a membership from one of the associations listed in § 4. C. Validity (2) ¹To become a member, an informal application is required. ²This application should be submitted in green ink on pink paper. (3) Membership applications cannot be rejected § 9 Effective Date (1) These articles will enter into force on 11.11.2011 at 11:11 am. § 6a Amendment to the Previous Clause (2) 1 If any provision of these by-laws is in conflict with any other, the by-laws will be repealed on 11.11.2011 at 11:11 am and 11 seconds. 2 The club is considered to be dissolved in this case. 1 With the repeal of \S 4, \S 6 paragraph 1 sentence 1 has become impractical. 2 In its place, memberships can be inherited. § 7 Termination of Membership ¹Membership ends with one's life. ²For non-living members, membership does not § 8 General Meeting

§ 1 Name, Legal Form, Headquarters

Figure 1: The four pages of the CfCH example of section 8

(1) A general meeting shall take place twice per year.

(2) The interval between two general meetings shall be no more than 6 months, 1 week, and 2 days.

(3) The invitation to the next general meeting shall be sent no earlier than 6 months from the previous general meeting.

10 State of Development

Since KOMA-Script 3.24, the scripura package has shared the version number of the classes and other important packages of KOMA-Script. Package contract on the other hand now has a new version number independent from KOMA-Script.

Independent from the version number you should note that so far, the interaction of the contract environment with the many different settings possible with other LATEX environments, packages, or classes has not been tested. The main reason for this is that contract is very specialized and far beyond the author's ordinary practice. So the author mostly relies on detailed user feedback.

Implementation

```
1 (*package)
```

11 Cooperation with hyperref

If hyperref has already loaded before contract the package cannot work correctly. So we throw an error. Maybe it would be a good idea to make this error fatal. But currently it is only an error.

```
2 \( \sinit \)
3 \@ifpackageloaded{hyperref}{%
4 \PackageError{contract}{Package hyperref already loaded}{%
5    If you want to use package contract with package hyperref, you have
6    to\MessageBreak
7    load package contract before package hyperref.\MessageBreak
8    To solve the problem, you just should move the loading of package
9    hyperref\MessageBreak
10    behind the loading of package contract.}%
11 }
12 \( \setminus \)
11 init\
```

12 Prerequisites

We need package scrkbase. We could load this also together with tocbasic, which is loaded later. But loading it on its own, we can require a minimum version.

Todo: Package contract is no longer a KOMA-Script package. So it should not use the internal KOMA-Script package scrkbase. For keys of contract environments we can just load scrbase. For package options we should either also switch to scrbase or use native key-value options of IATEX. After doing so we would be able to replace scrkbase at least by scrextend. With this package we still could use commands like \setkomafont. Theoretically we would also be able to still use KOMA-Script options, but this would still not be recommended.

```
13 \(\sinit\)
14 \RequirePackage{scrkbase} [2013/03/26]
```

```
15 (/init)
And now, tocbasic.
16 (*init)
17 \RequirePackage{tocbasic}
18 (/init)
```

13 Options

\cpar@indent (ilen.)

contract Option contract can be used to make the whole document to be a contract. But in this case \if@documentcontract you are not allowed to reuse the contract environment in the document. Nor is it allowed \@documentcontractfalse to stop or restart the contract. With older LATEX this is done by adding \contract to the \@documentcontracttrue end of \document. With an up to date LATEX we us a hook.

```
19 (*options)
20 \KOMA@ifkey{contract}{@documentcontract}
21 \IfLTXAtLeastTF{2020/10/01}{%
22
    \AddToHook{begindocument/end}{%
      \RelaxFamilyKey[.contract.sty]{KOMA}{contract}%
23
      \if@documentcontract\expandafter\contract\fi
24
   }%
25
26 }{%
    \g@addto@macro\document{%
27
      \RelaxFamilyKey[.contract.sty]{KOMA}{contract}%
28
      \if@documentcontract\expandafter\contract\fi
29
30
31 }
32 (/options)
```

juratotoc Allow to set the toc level of the entries. Value true is the same like 2, value false is the same like \maxdimen.

```
\if@juratotoc
     juratoclevel (cnt.)
                          33 (*options)
         \toclevel@cpar
                          34 \KOMA@key{juratotoc}[true]{%
                               \KOMA@set@ifkey{juratotoc}{@tempswa}{#1}%
                               \ifx\FamilyKeyState\FamilyKeyStateProcessed
                          37
                                 \if@tempswa
                                   \DeclareTOCStyleEntry[level=2]{default}{cpar}%
                          38
                          39
                                 \else
                                   \DeclareTOCStyleEntry[level=\maxdimen]{default}{cpar}%
                          40
                                 \fi
                          41
                               \else
                          42
                                 \DeclareTOCStyleEntry[level=#1]{default}{cpar}%
                          43
                          44
                               \KOMA@kav@xreplacevalue{contract.sty}{juratotoc}{\cpartocdepth}%
                          45
                          47 \KOMA@kav@xadd{contract.sty}{juratotoc}{\cpartocdepth}%
                          48 (/options)
     juratocnumberwidth Indent and number width of the toc entries.
          juratocindent
\cpar@numberwidth (ilen.)
```

Todo: Since we are using package tocbasic for the ToC entries these options (and lengths) are not needed any longer, but users should use \DeclareTOCStyleEntry to setup the number width and indent of the entries.

```
49 (*options)
50 \KOMA@key{juratocnumberwidth}{%
    \DeclareTOCStyleEntry[numwidth=#1]{default}{cpar}%
    \FamilyKeyStateProcessed
    \KOMA@kav@replacevalue{contract.sty}{juratocnumberwidth}{#1}%
53
54 }
55 \ \texttt{KOMA@kav@add} \{ \texttt{contract.sty} \} \{ \texttt{juratocnumberwidth} \} \{ \texttt{2em} \}
56 \KOMA@key{juratocindent}{%
    \DeclareTOCStyleEntry[indent=#1]{default}{cpar}%
    \FamilyKeyStateProcessed
58
    \KOMA@kav@replacevalue{contract.sty}{juratocindent}{#1}%
59
60 }
61 \KOMA@kav@add{contract.sty}{juratocindent}{1.5em}%
62 (/options)
```

juratitlepagebreak The options sets the boolean \if@juratitlepagebreak.

\@juratitlepagebreaktrue ommended). \@juratitlepagebreakfalse

\if@juratitlepagebreak If the boolean is \iftrue page breaks inside clause headings are allowed (which is not rec-

Todo: Re-implementation either using a native key-value option (easy) or a new scrbase family.

```
63 (*options)
64 \KOMA@ifkey{juratitlepagebreak}{@juratitlepagebreak}
65 (/options)
```

parnumber The options switches the (automatic) paragraph numbering.

Todo: Re-implementation either using a native key-value option (easy) or a new scrbase family.

```
66 (*options)
67 \newif\ifparnumber
68 \KOMA@key{parnumber}[true]{%
    \Ifstr{#1}{auto}{%
      \AutoPar
70
      \FamilyKeyStateProcessed
71
      \KOMA@kav@remove{contract.sty}{parnumber}{manual}%
72
73
      \KOMA@kav@remove{contract.sty}{parnumber}{auto}%
      \KOMA@kav@add{contract.sty}{parnumber}{auto}%
74
75
      \Ifstr{#1}{manual}{%
76
77
        \ManualPar
        \FamilyKeyStateProcessed
78
        \KOMA@kav@remove{contract.sty}{parnumber}{manual}%
79
        \KOMA@kav@remove{contract.sty}{parnumber}{auto}%
80
        \KOMA@kav@add{contract.sty}{parnumber}{manual}%
81
      }{%
82
        \KOMA@set@ifkey{parnumber}{parnumber}{#1}%
83
```

```
\KOMA@kav@replacebool{contract.sty}{parnumber}{parnumber}%
85
      }%
86
    }%
87 }
88 \KOMA@kav@add{contract.sty}{parnumber}{true}
89 \KOMA@kav@add{contract.sty}{parnumber}{auto}
90 (/options)
```

clausemark

paragraphmark Todo: Re-implementation either using a native key-value option or a new scrbase family (easy).

markboth

markright Todo: Remove deprecated options, because they need an internal KOMA-Script macro.

\Clausemark The options are used to activate either \markright or \markboth for clauses. \Clausemark expects not only the title but also the number. So it differs from, e.g., \chaptermark, which uses the counter automatically. But maybe I will change this some time.

```
91 (*options)
92 \mbox{ } 1]{}
93 \KOMA@key{clausemark}{%
     \begingroup
94
       \KOMA@set@ncmdkey{clausemark}{@tempa}{%
95
         {false}{0},{off}{0},{no}{0},%
96
97
         {forceright}{1},%
         {forceboth}{2},%
99
         {right}{3},%
100
         {both}{4}%
101
       }{#1}%
     \ifx\FamilyKeyState\FamilyKeyStateProcessed
102
       \ifcase\number\@tempa
103
104
         \endgroup
         \let\Clausemark\@gobble
105
106
         \endgroup
107
         \renewcommand*{\Clausemark}[1]{%
108
109
           \markright{\csname MakeMarkcase\endcsname{##1}}}%
110
         \endgroup
111
         \renewcommand*{\Clausemark}[1]{%
112
           \markboth{\csname MakeMarkcase\endcsname{##1}}%
113
                     {\csname MakeMarkcase\endcsname{##1}}}%
114
       \or
115
116
         \endgroup
         \renewcommand*{\Clausemark}[1]{%
117
118
             \@mkboth\@gobbletwo
119
120
             \markright{\csname MakeMarkcase\endcsname{##1}}%
121
           fi}%
122
       \or
123
124
         \endgroup
125
         \renewcommand*{\Clausemark}[1]{%
           \@mkboth{\csname MakeMarkcase\endcsname{##1}}%
126
```

```
{\csname MakeMarkcase\endcsname{##1}}}%
127
128
       \else
         \endgroup
129
130
       \fi
131
       \FamilyKeyStateProcessed
132
     \else
       \endgroup
133
       \FamilyKeyStateUnknownValue
134
135
     \KOMA@kav@xreplacevalue{contract.sty}{clausemark}{#1}%
136
137 }
138 \KOMA@kav@add{contract.sty}{clausemark}{false}
139 \@ifundefined{KOMA@DeclareDeprecatedOption}{}{%
     \KOMA@DeclareDeprecatedOption[contract] {markright} {clausemark=forceright}%
     \KOMA@DeclareDeprecatedOption[contract]{markboth}{clausemark=forceboth}%
141
142 }
143 \KOMA@key{paragraphmark}{%
     \PackageWarningNoLine{contract}{%
144
       You've used obsolete option 'paragraphmark'.\MessageBreak
145
       Usage of this option is deprecated.\MessageBreak
146
147
       You should simply replace 'paragraphmark'\MessageBreak
       by 'clausemark'%
148
149
150
     \KOMAExecuteOptions[.contract.sty]{clausemark=#1}%
151 }
152 (/options)
```

ref **Todo:** Re-implementation either using a native key-value option or a new scrbase family (easy).

parcitename The formatting of the references of paragraphs and sentences. There are a long a short and sentencecitename a numeric form.

Todo: Remove deprecated options, because they need an internal KOMA-Script macro.

 $\protect\$ Default is the long form. Corresponding values of the two helper macros are: 0 = long, 1 = long,

```
153 (*options)
154 \newcommand*{\parcite@format}{0}
155 \newcommand*{\sentencecite@format}{0}
The options can be used to change the defau
```

The options can be used to change the default.

```
156 \KOMA@key{ref}{%
     \begingroup
157
       \KOMA@set@ncmdkey{ref}{@tempa}{%
158
159
         {parlong}{1},{longpar}{1},{ParL}{1},%
160
         {parshort}{2},{shortpar}{2},{ParS}{2},%
         {parnumeric}{3}, {numericpar}{3}, {ParN}{3}, %
161
         {paroff}{4}, {nopar}{4}, %
162
         {sentencelong}{10}, {longsentence}{10}, {SentenceL}{10}, %
163
         {sentenceshort}{20}, {shortsentence}{20}, {SentenceS}{20}, %
164
         {sentencenumeric}{30}, {numericsentence}{30}, {SentenceN}{30}, %
165
```

```
{sentenceoff}{40},{nosentence}{40},%
166
167
          {long}{11},%
168
          {short}{22},%
169
          {numeric}{33},%
          {paragraphonly}{44},{onlyparagraph}{44},%
170
          {ParagraphOnly}{44}, {OnlyParagraph}{44}%
171
       }{#1}%
172
        \ifx\FamilyKeyState\FamilyKeyStateProcessed
173
          \aftergroup\FamilyKeyStateProcessed
174
          \@tempcnta=\@tempa\relax
175
          \ensuremath{\texttt{0tempcntb=}\z0}
176
          \@whilenum \@tempcnta>9 \do{%
177
178
            \advance\@tempcnta -10\relax
179
            \advance\@tempcntb \@ne\relax
180
         }%
          \ifcase \@tempcnta
181
182
          \or
            \aftergroup\def\aftergroup\parcite@format
183
            \aftergroup{\aftergroup0\aftergroup}%
184
185
            \aftergroup\def\aftergroup\parcite@format
186
            \aftergroup{\aftergroup1\aftergroup}%
187
          \or
188
189
            \aftergroup\def\aftergroup\parcite@format
190
            \aftergroup{\aftergroup2\aftergroup}\%
191
            \aftergroup\def\aftergroup\parcite@format
192
            \aftergroup{\aftergroup-\aftergroup1\aftergroup}%
193
194
          \fi
195
          \ifcase \@tempcntb
196
          \or
            \aftergroup\def\aftergroup\sentencecite@format
197
198
            \aftergroup{\aftergroup0\aftergroup}%
199
            \aftergroup\def\aftergroup\sentencecite@format
200
201
            \aftergroup{\aftergroup1\aftergroup}%
202
            \verb|\aftergroup\ef\aftergroup\sentencecite@format| \\
203
            \aftergroup{\aftergroup2\aftergroup}%
204
205
206
            \aftergroup\def\aftergroup\sentencecite@format
207
            \aftergroup{\aftergroup-\aftergroup1\aftergroup}%
208
          \fi
209
        \else
210
          \aftergroup\FamilyKeyStateUnknownValue
211
        \fi
212
     \endgroup
     \verb|\familyKeyState| FamilyKeyState| Processed|
213
        \KOMA@kav@removekey{contract.sty}{ref}%
214
       \ifcase\parcite@format
215
216
          \KOMA@kav@add{contract.sty}{ref}{parlong}%
217
        \or
```

```
\KOMA@kav@add{contract.sty}{ref}{parshort}%
218
219
220
         \KOMA@kav@add{contract.sty}{ref}{parnumeric}%
221
       \or
         \KOMA@kav@add{contract.sty}{ref}{paroff}%
222
       \fi
223
       \ifcase\sentencecite@format
224
         \KOMA@kav@add{contract.sty}{ref}{sentencelong}%
225
226
         \KOMA@kav@add{contract.sty}{ref}{sentenceshort}%
227
228
         \KOMA@kav@add{contract.sty}{ref}{sentencenumeric}%
229
230
         \KOMA@kav@add{contract.sty}{ref}{sentenceoff}%
231
       \fi
232
     \fi
233
234 }
235 \KOMA@kav@add{contract.sty}{ref}{parlong}%
236 \KOMA@kav@add{contract.sty}{ref}{sentencelong}%
237 \@ifundefined{KOMA@DeclareDeprecatedOption}{}{%
     \KOMA@DeclareDeprecatedOption[contract]{parcitename}{ref=parlong}
     \KOMA@DeclareDeprecatedOption[contract]{sentencecitename}{ref=sentencelong}
240 }
241 (/options)
 Execute the options.
242 (*postoptions)
243 \KOMAProcessOptions\relax
244 (/postoptions)
```

14 Deprecated Paragraph Commands

All these commands are deprecated and now generate error messages instead of code.

```
\Paragraph Todo: Remove deprecated commands, because they already throw errors for about nine
               \SubParagraph
                                   years.
               \refParagraph
                               245 (*body)
              \refParagraphN
        \DeprecatedParagraph
                              247
\ParagraphCompatibilityHacks
                              248
                               249
                                      0.7a.\MessageBreak
                               250
                               251
                                      depending\MessageBreak
```

```
246 \providecommand*{\DeprecatedParagraph}{%
     \PackageError{contract}{modification of old document needed}{%
       It seem that this document was made for scrjura up to version
       Since scrjura version 0.9 \string\Paragraph, \string\SubParagraph, and all
       commands, options, and counters have been renamed.\MessageBreak
252
253
       You should replace the terms 'Paragraph' and 'paragraph' by 'Clause'
       and\MessageBreak
254
       'clause' if they are part of the name of a contract feature, otherwise
255
257
       document may produce severall additional error messages and maybe the
258
       wrong\MessageBreak
```

```
result. Sorry for the inconvenience.%
260
261
     \ParagraphCompatibilityHacks
262 }
263 \newcommand*{\ParagraphCompatibilityHacks}{%
     \PackageWarning{contract}{compatibility hacks for '\string\Paragraph'
264
265
       executed.\MessageBreak
       There is no support for documents using these hacks!\MessageBreak
266
       There is no warranty for real compatibility!\MessageBreak
267
       Even if the LaTeX run of the document doesn't report\MessageBreak
268
       any error, the result may be completely wrong.\MessageBreak
269
       Therefore it is recommended to solve the problem,\MessageBreak
270
       instead of trying to work around using the \MessageBreak
271
272
       compatibility hacks%
    }%
273
     \gdef\Paragraph{\Clause}%
274
275
     \gdef\SubParagraph{\SubClause}%
     \gdef\c@Paragraph{\c@Clause}%
276
     \gdef\cl@Paragraph{\cl@Clause}%
277
     \gdef\c@SubParagraph{\c@SubClause}%
278
279
     \gdef\cl@SubParagraph{\cl@SubClause}%
     \gdef\theParagraph{\theClause}%
280
     \gdef\theSubParagraph{\theSubClause}%
     \gdef\refParagraph{\refClause}%
283
     \gdef\refParagraphN{\refClauseN}%
284
     \aliaskomafont{Paragraph}{Clause}%
285
     \scr@ifundefinedorrelax{Paragraphmark}{}{%
286
       \global\let\Clausemark\Paragraphmark
287
     }%
288 }
289 \providecommand*{\Paragraph}{\DeprecatedParagraph\Paragraph}
290 \providecommand*{\SubParagraph}{\DeprecatedParagraph\SubParagraph}
291 \providecommand*{\refParagraph}{\DeprecatedParagraph\refParagraph}
292 \providecommand*{\refParagraphN}{\DeprecatedParagraph\refParagraphN}
293 (/body)
```

15 Contracts, Clauses, Paragraphs and Sentences

\contract@env@type This macro shows the currently active contract environment.

```
294 \langle *body \rangle
295 \newcommand*{\contract@env@type}{}
296 \langle /body \rangle
```

\ellipsispar Count one or more paragraphs given by the optional argument but print \parellipsis \parellipsis instead of a real paragraph. The default is either \dots or \textellipsis if available.

```
297 \*body\
298 \newcommand*{\ellipsispar}[1][1]{%
299 \begingroup
300 \KOMAoptions{parnumber=manual}\parellipsis\par
301 \addtocounter{par}{#1}%
```

```
305
                                                            {\csname the\contract@env@type
                              306
                                                              AbsoluteClause\endcsname}%
                              307
                                                            {\thepar}%
                                        }%
                              308
                                      \fi
                              309
                                    \endgroup
                              310
                                    \addtocounter{par}{-1}\refstepcounter{par}%
                              311
                                    \ignorespaces
                              314 \newcommand*{\parellipsis}{%
                                    \scr@ifundefinedorrelax{textellipsis}{\dots}{\textellipsis}%
                              316 }
                              317 (/body)
                    contract It is not allowed to nest the contract environments, but you can end them and start them
  \if@contract@skiphyperref new. But this would not end the contract and start a new contract but only delay it for
        contractClause (cnt.) some other code.
          \thecontractClause
                              318 (*body)
     \contract@Clauseformat 319 \newenvironment{contract}{%
               \Clauseformat 320
                                   \ifx\contract@env@type\@empty
                                      \let\@doendpe\contract@doendpe
           \paragraphformat 321
                                      \let\Clause\contract@paragraph
     \verb|contractSubClause| (cnt.) \quad ^{322}
                              323
                                      \let\c@Clause\c@contractClause
      \thecontractSubClause
                                      \edef\cl@Clause\cl@contractClause}%
                              324
contractAbsoluteClause (cnt.)
                              325
                                      \let\SubClause\contract@subparagraph
                                      \let\c@SubClause\c@contractSubClause
                              326
                              327
                                      \edef\cl@SubClause{\cl@SubClause\cl@contractSubClause}%
                              328
                                      \let\Sentence\contract@sentence
                                      \renewcommand*{\contract@env@type}{contract}%
                              329
                                      \aliaskomafont{Clause}{contract.Clause}%
                              330
                              331
                                      \PackageError{contract}{nested 'contract' detected}{%
                              332
                                        You may not use a 'contract' environment inside\MessageBreak
                              333
                                        a '\contract@env@type' environment or after loading\MessageBreak
                              334
                                        package 'contract' with option '\contract@env@type'!}%
                              335
                                   \fi
                              336
                              337 }{}
                              338 \let\if@contract@skiphyperref\iftrue
                              339 \let\cl@Clause\@empty
                              340 \let\cl@SubClause\@empty
                              341 \newcounter{contractClause}
                              342 \renewcommand*{\thecontractClause}{%
                                   {\contract@Clauseformat{\arabic{Clause}}}}
                              344 \DeclareRobustCommand*{\contract@Clauseformat}[1]{\Clauseformat{#1}}
                              345 \mbox{ } \mbox{clauseformat}[1]{\mbox{\sc } }
                              346 \newcounter{contractSubClause}
                              347 \@addtoreset{SubClause}{Clause}
                              348 \renewcommand*{\thecontractSubClause}{%
                              349 {\theClause\alph{SubClause}}}
```

\if@filesw

303

304

\protected@write\@auxout{}{%

\string\newmaxpar{\contract@env@type}%

```
350 \newcounter{contractAbsoluteClause}
351 (/body)
```

\DeclareNewJuraEnvironment Using \\c_atsign_strdefjuraenvironment to define a new juristic environment. This can be done only in document preamble.

```
352 (*body)
353 \newcommand*{\DeclareNewJuraEnvironment}[1]{%
                         \@ifundefined{#1}{\expandafter\let\csname #1\expandafter\endcsname
                                   \csname end#1\endcsname}{}%
355
                         \verb|\defined{#1}{\left| \text{let}\ensuremath} | \text{$0$} 
356
357
                                   \PackageError{contract}{ignorring declaration of '#1'}{%
358
                                           You've tried to declare jura environment '#1', but
359
                                           environment\MessageBreak
360
                                            '#1' or command
361
                                            \expandafter\string\csname #1\endcsname\space or
                                           \expandafter\string\csname end#1\endcsname\MessageBreak
362
363
                                           already exists.\MessageBreak
364
                                           Declaration will be ignored}%
365
                                   \long\def\reserved@defjuraenvironment##1[##2]##3##4{}%
366
                       }%
                         \kernel@ifnextchar [%]
367
368
                                   {\reserved@defjuraenvironment{#1}}{\reserved@defjuraenvironment{#1}[]}%
369 }
370 \@onlypreamble\DeclareNewJuraEnvironment
```

392

\@defjuraenvironment This command is used to define a new contract environment like contract. Several options are provided (see the user manual for details).

```
371 \DefineFamily{KOMAarg}
372 \DefineFamilyMember{KOMAarg}
373 \newcommand{\@defjuraenvironment}{}
374 \long\def\@defjuraenvironment#1[#2]#3#4{%
      \let\reserved@defjuraenvironment\relax
The counters:
```

```
376
     \newcounter{#1Clause}%
377
     \newcounter{#1AbsoluteClause}%
     \newcounter{#1SubClause}%
378
     \FamilyCSKey[.contract.sty]{KOMAarg}{Clause}{#1@Clause}%
379
     \FamilyCSKey[.contract.sty]{KOMAarg}{SubClause}{#1@SubClause}%
380
381
     \FamilyCSKey[.contract.sty]{KOMAarg}{Sentence}{#1@Sentence}%
382
     \DefineFamilyKey[.contract.sty]{KOMAarg}{ClauseNumberFormat}{%
383
       \expandafter\def\csname #1@Clauseformat \endcsname####1{##1{###1}}%
384
       \expandafter\edef\csname #1@Clauseformat\endcsname{%
         \noexpand\protect\expandafter\noexpand\csname #1@Clauseformat \endcsname
385
386
387
       \FamilyKeyStateProcessed
     }
388
     \DefineFamilyKey[.contract.sty]{KOMAarg}{ClauseFont}{%
389
       \IfExistskomafont{#1.Clause}{%
390
391
         \IfIsAliaskomafont{#1.Clause}{%
```

\expandafter\let\csname scr@fnt@instead@#1.Clause\endcsname\relax

```
\newkomafont{#1.Clause}{##1}%
 393
 394
          }{\setkomafont{#1.Clause}{##1}}%
        }{%
 395
396
          \newkomafont{#1.Clause}{##1}%
        }%
397
      }
398
      \FamilyExecuteOptions[.contract.sty]{KOMAarg}{#2}%
399
      \RelaxFamilyKey[.contract.sty]{KOMAarg}{ClauseFont}%
 400
      \RelaxFamilyKey[.contract.sty]{KOMAarg}{ClauseNumberFormat}%
 401
      \RelaxFamilyKey[.contract.sty]{KOMAarg}{Sentence}%
 402
      \RelaxFamilyKey[.contract.sty]{KOMAarg}{SubClause}%
 403
      \RelaxFamilyKey[.contract.sty]{KOMAarg}{Clause}%
 404
      \@ifundefined{#1@Clauseformat}{%
 405
        \expandafter\DeclareRobustCommand\expandafter*%
 406
        \csname #1@Clauseformat\endcsname[1]{\Clauseformat{##1}}%
 407
 408
      }{}%
      \expandafter\renewcommand\expandafter*\csname the#1Clause\endcsname{%
 409
        \protect\@nameuse{#1@Clauseformat}{\arabic{#1Clause}}}%
410
Environment:
411
      \newenvironment{#1}{%
412
        \par
        \ifx\contract@env@type\@empty
413
          \edef\contract@env@type{#1}%
 414
 415
          \let\@doendpe\contract@doendpe
416
          \expandafter\let\expandafter\c@Clause\csname c@#1Clause\endcsname
417
          \edef\cl@Clause{\clause\csname cl@#1Clause\endcsname}%
418
          \expandafter\let\expandafter\c@SubClause
            \csname c@#1SubClause\endcsname
419
          \edef\cl@SubClause{\cl@SubClause
420
            \csname cl@#1SubClause\endcsname}%
 421
          \@ifundefined{#1@Clause}{%
 422
            \let\Clause\contract@paragraph
 423
 425
            \expandafter\let\expandafter\Clause
 426
            \csname #1@Clause\endcsname
 427
          ጉ%
          \@ifundefined{#1@SubClause}{%
428
            \verb|\label{contract@subparagraph|} \label{contract@subparagraph} |
 429
430
            \expandafter\let\expandafter\SubClause
 431
            \csname #1@SubClause\endcsname
 432
433
          \@ifundefined{#1@Sentence}{%
 434
            \let\Sentence\contract@sentence
 435
          }{%
 436
            \expandafter\let\expandafter\Sentence\csname #1@Sentence\endcsname
 437
          }%
 438
          \@ifundefined{\contract@env@type @everypar}{%
 439
            \expandafter\let
 440
            \csname \contract@env@type @everypar\endcsname
 441
```

442

\contract@everypar

```
443 }{}%
```

Font alias for Clause. If neither a font not an alias is defined for the new environment contract.Clause is used.

```
\IfExistskomafont{#1.Clause}{%
444
            \IfIsAliaskomafont{#1.Clause}{%
445
              \aliaskomafont{Clause}{\csname scr@fnt@instead@#1.Clause\endcsname}%
446
            }{%
447
              \aliaskomafont{Clause}{#1.Clause}%
448
            }%
449
450
         }{%
            \aliaskomafont{Clause}{contract.Clause}%
         }%
452
         #3%
453
       \else
454
          \PackageError{contract}{nested contract environments detected}{%
455
            You must not use a '#1' environment inside\MessageBreak
456
            a '\contract@env@type' environment or after loading\MessageBreak
457
            package 'contract' with option '\contract@env@type'!}%
458
459
       \fi
     }{%
460
       #4%
461
462
        \par
463
     }%
464 }
465 \langle \mathsf{/body} \rangle
```

\contract@paragraph This is the \Clause used by contracts. A contract consists (usually) of several clauses. Each clause has optional elements managed by $\langle key \rangle = \langle value \rangle$ pairs handled by scrkbase and last but not least by keyval.

title Title, running head and toc entry of the clause. The title is the default for running head and head toc entry. But you can also use an empty value for each of them or use the no... options nohead to switch them off.

```
entry
           466 (*body)
   noentry
            467 \define@key{contract}{title}{%
  tocentry
            468
                 \def\contract@title{#1}%
                 \ifx\contract@entry\relax\def\contract@entry{\contract@title}\fi
notocentry
            469
                 \ifx\contract@head\relax\def\contract@head{\contract@title}\fi
            470
            471 }
            472 \define@key{contract}{entry}{%
            473
                 \PackageWarning{contract}{deprecated option 'entry'.\MessageBreak
            474
                   You should use option 'tocentry' instead of \MessageBreak
            475
                   option 'entry'%
            476
                 }%
                 \def\contract@entry{#1}}
            477
            478 \define@key{contract}{tocentry}{\def\contract@entry{#1}}
            479 \define@key{contract}{noentry}[]{%
                 \verb|\PackageWarning{contract}{deprecated option `noentry'.} \\|
            480
                   You should use option 'notocentry' instead of \MessageBreak
            481
            482
                   option 'noentry'%
            483
                 }%
```

```
484 \let\contract@entry\relax}
485 \define@key{contract}{notocentry}[]{\let\contract@entry\relax}
486 \define@key{contract}{head}{\def\contract@head{#1}}
487 \define@key{contract}{nohead}[]{\let\contract@head\relax}
```

number The number can be changed manually. But clauses without numbers are not allowed. So if you use an empty value, the number is automatically set.

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

\contract@preskip The options are used to specify the distance before and after the clause. The preset value \contract@postskip of these options are the global settings done by \setkeys{contract}{...}.

```
{\tt preskip} \quad {\tt 489 \ \ less command*{\ \ \ \ \ }} \ \{2\ \ \ \ \ \}} \ \{2\ \ \ \ \ \ \}
postskip 490 \newcommand*{\contract@postskip}{\baselineskip}
            491 \define@key{contract}{preskip}{\def\contract@preskip{#1}}
            492 \define@key{contract}{postskip}{\def\contract@postskip{#1}}
```

dummy The option switches the boolean \ifcontract@dummy.

\ifcontract@dummy If the boolean is \iftrue the clause will not be printed. But note: you cannot use this to remove the paragraphs or sentences of the clause. But you can use this option to generate holes in the numbering without manually manipulating the counters.

```
493 \newif\ifcontract@dummy
494 \define@key{contract}{dummy}[true]{\csname contract@dummy#1\endcsname}
```

contract.Clause (font) Correctly this macro should be named \contract@paragraph@format. But it is already used by \contract@paragraph@font some users for ugly tricks. So I will not rename it to avoid problems for existing documents. Additionally it would be better to use a new macro per environment. However the same reason not to change this.

```
496 \verb|\newcommand*{\contract@paragraph@font}{\usekomafont{Clause}} \% 
                       \@hangfrom}
@AbsClause (cnt.)
\theH@AbsClause _{498} % Here we have some not good tested code for \pkg{hyperref}.
    \theHClause 499 \newcounter{@AbsClause}
 \theHSubClause 500 \def\theH@AbsClause{P-\arabic{@AbsClause}}
                  501 \def\theHClause{\theH@AbsClause}
```

502 \def\theHSubClause{\theH@AbsClause}

495 \newkomafont{contract.Clause}{\sffamily\bfseries\large}

For the headings we use manual paragraph numbering, because we don't want any paragraph numbering inside the heading. After initializing the options they are processed.

```
503 \NewDocumentCommand\contract@paragraph {o} {%
     \stepcounter{\contract@env@type AbsoluteClause}%
504
     \ManualPar\parnumbertrue
505
     \let\contract@title\relax
506
507
     \let\contract@entry\relax
     \let\contract@head\relax
508
     \let\contract@number\relax
509
     \contract@dummyfalse
510
     \IfValueT{#1}{\setkeys{contract}{#1}}%
511
```

Unless this is a dummy clause, the headings will be initialized and vertical skips will be done.

```
512 \ifcontract@dummy\else
513 \par
514 \@afterindentfalse
515 \addvspace{\contract@preskip}%
516 \fi
```

If there isn't a manual number, we use the next number. If there is a manual number, this number is printed and we take care that labels and hyperref also use the manual number.

```
\ifx\contract@number\relax
518
        \let\p@Clause\@empty
        \expandafter\let\expandafter\theClause
519
520
          \csname the\contract@env@type Clause\endcsname
521
        \refstepcounter{Clause}%
522
        \begingroup
523
524
          \let\@elt\@stpelt
525
          \cl@Clause
526
        \endgroup
        \protected@edef\theClause{%
527
          \protect\@nameuse{\contract@env@type @Clauseformat}{\contract@number}%
528
529
        \protected@edef\@currentlabel{\theClause}%
530
531
       \def\@currentcounter{Clause}%
532
     \stepcounter{@AbsClause}%
533
     \begingroup\expandafter\expandafter\expandafter\endgroup
534
     \expandafter\ifx\csname if@skiphyperref\endcsname\relax
535
536
        \expandafter\let\csname if@contract@skiphyperref\expandafter\endcsname
537
538
        \csname if@skiphyperref\endcsname
539
540
     \if@contract@skiphyperref\else
        \hyper@refstepcounter{@AbsClause}%
               \typeout{absolute Number: \the@AbsClause^^JLabel: '\@currentHref'}%
542 \langle +trace \rangle
543
```

For simplification we use the code of clauses for sub-clauses.

544 \let\theSubClause\theClause

Unless for dummy clauses, the heading is printed, the toc entry is done and also the running head.

```
\ifcontract@dummy\else
545
       \begingroup
546
547
         \if@juratitlepagebreak\else\interlinepenalty\@M\fi
548
         \contract@paragraph@font{\theClause
           \ifx\contract@title\relax\else\enskip\fi}%
549
         \contract@title
550
         \ifx\contract@entry\relax\else
551
           \expandafter\addxcontentsline\expandafter{\ext@toc}%
552
553
           {cpar}[\theClause]\contract@entry
```

```
\addxcontentsline{cpa}{cpar}[\theClause]\contract@entry
                         555
                         556
                                  \ifx\contract@head\relax\else
                         557
                                     \expandafter\Clausemark\expandafter{%
                         558
                                       \expandafter\theSubClause\expandafter\enskip\contract@head}%
                                  \fi
                         559
                         560
                                  \par
                                \endgroup\nobreak\vskip\contract@postskip
                         561
                        Last but not least paragraph numbering is initialized.
                                \contract@afterheading
                         562
                              \fi
                         563
                         564 }
                        565 (/body)
\contract@subparagraph This is almost the same like \contract@paragraph.
                         566 (*body)
                         567 \NewDocumentCommand \contract@subparagraph {o}{%
                              \stepcounter{\contract@env@type AbsoluteClause}%
                         568
                              \ManualPar\parnumbertrue
                         569
                              \let\contract@title\relax
                         570
                         571
                              \let\contract@entry\relax
                         572
                              \let\contract@head\relax
                              \let\contract@number\relax
                         573
                              \contract@dummyfalse
                              \IfValueT{#1}{\setkeys{contract}{#1}}%
                         576
                              \ifcontract@dummy\else
                         577
                                \par
                                \@afterindentfalse
                         578
                                \vskip\contract@preskip
                         579
                         580
                              \ifx\contract@number\relax
                         581
                                \let\p@SubClause\@empty
                         582
                                \let\theSubClause\thecontractSubClause
                         583
                                \refstepcounter{SubClause}%
                         584
                         585
                         586
                                \begingroup
                         587
                                  \let\@elt\@stpelt
                         588
                                  \cl@SubClause
                         589
                                \endgroup
                                \protected@edef\theSubClause{\theClause\contract@number}%
                         590
                                \protected@edef\@currentlabel{\theSubClause}%
                         591
                                \def\@currentcounter{SubClause}%
                         592
                         593
                              \stepcounter{@AbsClause}%
                         594
                              \begingroup\expandafter\expandafter\expandafter\endgroup
                         595
                              \expandafter\ifx\csname if@skiphyperref\endcsname\relax
                         596
                         597
                         598
                                \expandafter\let\csname if@contract@skiphyperref\expandafter\endcsname
                         599
                                \csname if@skiphyperref\endcsname
                         600
                              \if@contract@skiphyperref\else
                         601
```

```
\hyper@refstepcounter{@AbsClause}%
603 (+trace)
               \typeout{absolute Number: \the@AbsClause^^JLabel: '\@currentHref'}%
604
     \fi
605
     \ifcontract@dummy\else
606
       \begingroup
         \if@juratitlepagebreak\else\interlinepenalty\@M\fi
607
         \contract@paragraph@font{\theSubClause
608
           \ifx\contract@title\relax\else\enskip\fi}%
609
         \contract@title
610
         \ifx\contract@entry\relax\else
611
           \expandafter\addxcontentsline\expandafter{\ext@toc}%
           {cpar}[\theSubClause]\contract@entry
           \addxcontentsline{cpa}{cpar}[\theSubClause]\contract@entry
614
615
         \verb|\ifx\contract@head\relax\else| \\
616
           \expandafter\Clausemark\expandafter{%
617
618
              \expandafter\theSubClause\expandafter\enskip\contract@head}%
619
         \fi
         \par
620
       \endgroup
621
       \nobreak\vskip\contract@postskip
622
       \contract@afterheading
623
624
     \fi
625 }
626 (/body)
```

\AutoPar Switching between automatic or manual paragraph numbers for all contract environments.

```
\ManualPar 627 \langle \text{body} \\
628 \newcommand*{\AutoPar}{\%} \\
629 \expandafter\let\expandafter\contract@used@everypar \\
630 \csname \contract@env@type @everypar\endcsname \\
631 \rangle \\
632 \newcommand*{\ManualPar}{\%} \\
633 \let\contract@used@everypar\relax \\
634 \rangle \\
635 \langle \/
635 \langle \/
635 \langle \/
635 \langle \/
636 \\
637 \langle \/
638 \\
638 \langle \/
```

\contract@afterheading Similar to \afterheading but with automatic paragraph numbers.

ToDo: Test if this can be done using LATEX hooks, depending on the LATEX release.

```
636 (*body)
637 \CheckCommand*{\@afterheading}{%
     \@nobreaktrue
     \everypar{%
639
640
       \if@nobreak
641
         \@nobreakfalse
         \clubpenalty \@M
642
         \if@afterindent \else
643
           {\setbox\z@\lastbox}%
644
645
         \fi
646
       \else
```

```
\clubpenalty \@clubpenalty
          \everypar{}%
648
        \fi}%
649
650 }
651 \newcommand*{\contract@afterheading}{%
     \@nobreaktrue
652
     \everypar{%
653
        \if@nobreak
654
          \@nobreakfalse
655
656
          \clubpenalty \@M
          \if@afterindent \else
657
            {\setbox\z@\lastbox}%
658
          \fi
659
660
        \else
          \clubpenalty \@clubpenalty
661
          \everypar{%
662
663
            \contract@used@everypar
         }%
664
        \fi
665
        \contract@used@everypar
666
667
     \AutoPar
668
669 }
```

\contract@used@everypar

The macro to be used at the very beginning of every paragraph to add the number. To be used only inside contract environments, so empty outside.

670 \newcommand*{\contract@used@everypar}{}

\@doendpe IATFX used this macro, to reset all paragraph actions at the end of environments. To avoid unwanted switching-off of the paragraph number it will be reinitialized.

\EndIncludeInRelease

\contract@doendpe From IATEX 2015/01/01 a different definition of \\c_atsign_strdoendpe is used. So we also \IncludeInRelease have to use different versions depending on the release. We do so with some tricks. Maybe \@gobble@IncludeInRelease this should be replaced by usage of \IfLTXAtLeastTF from already loaded scrbase.

Todo: Using \everypar is evil and should be replaced by using a generic paragraph hook.

```
671 \providecommand*{\IncludeInRelease}[3]{%
     \PackageInfo{contract}{temporary definition of \string\IncludeInRelease}%
672
     \Ifstr{#1}{0000/00/00}{%
673
       \let\IncludeInRelease\@undefined
674
       \def\EndIncludeInRelease{\let\EndIncludeInRelease\@undefined}%
675
676
       \let\EndIncludeInRelease\relax
677
       \long\def\@gobble@IncludeInRelease##1\EndIncludeInRelease{%
678
         \let\@gobble@IncludeInRelease\@undefined
679
680
       \expandafter\@gobble@IncludeInRelease
681
682
683 }
684 \IncludeInRelease{2015/01/01}{\@doendpe}{clubpenalty fix}
685 \CheckCommand*\@doendpe{\@endpetrue
        \def\par{\@restorepar
```

```
\clubpenalty\@clubpenalty
687
688
                  \everypar{}\par\@endpefalse}\everypar
689
                   {{\setbox\z@\lastbox}%
                    \everypar{}\@endpefalse}}
690
691
   \newcommand*{\contract@doendpe}{%
     \@endpetrue
692
     \def \par{%
693
        \@restorepar
694
        \clubpenalty\@clubpenalty
695
696
        \everypar{%
          \csname contract@used@everypar\endcsname
697
698
699
        \par\@endpefalse
700
     \everypar{%
701
       {\setbox\z@\lastbox}\everypar{%
702
703
          \csname contract@used@everypar\endcsname
704
        \@endpefalse
705
     }%
706
707 }
708 \EndIncludeInRelease
709 \IncludeInRelease{0000/00/00}{\@doendpe}{clubpenalty fix}
710 \CheckCommand*\@doendpe{\@endpetrue
     \def\par{\@restorepar\everypar{}\par\@endpefalse}\everypar
711
712
     {{\setbox\z@\lastbox}\everypar{}\@endpefalse}}
713 \newcommand*{\contract@doendpe}{%
     \@endpetrue
714
715
     \def\par{%
716
       \@restorepar\everypar{%
          \csname contract@used@everypar\endcsname
717
718
       \par\@endpefalse
719
720
721
     \everypar{%
722
       {\setbox\z@\lastbox}\everypar{%
723
          \csname contract@used@everypar\endcsname
724
725
        \@endpefalse
726
     }%
727 }
728 \EndIncludeInRelease
729 (/body)
```

16 Entry to Table of Contents

\lambda@cpar Toc entry of contract clauses. This is done using tocbasic. The definition has to be part of the initialization of the package, otherwise package options wouldn't be able to change the setting.

```
730 (*init)
```

```
731 \DeclareTOCStyleEntry[%
732 indent=1.5em,
733 numwidth=2em,
734 level=\maxdimen
735 ]{default}{cpar}
736 \/init\
```

17 Numbering of Paragraphs and Sentences

```
\contract@separator Used to make it possible to remove white spaces at the beginning or end.
                     737 (*body)
                     738 \DeclareRobustCommand*{\contract@separator}[1]{#1}
                     739 (/body)
  \contract@usetype By default it is robust but does only call \contract@@usetype with the only argument.
 \contract@usetype This second command is not robust and can easily be redefined. But by default it also does
                    nothing but eating the argument.
                     742 \newcommand*{\contract@@usetype}[1]{}
 \contract@everypar The \contract@everpar used by contracts.
       \ifparnumber The boolean defines if paragraph numbers have to be used. If they are deactivated also
     \parnumbertrue manual paragraph numbers are deactivated and the paragraphs are not counted. Otherwise
    \parnumberfalse the paragraphs are numbered using \thepar. It is important to reset the paragraph counter
           par (cnt.) with every clause and sub-clause. And for labels the parent object the clause has to be used.
            \thepar 744 \ \langle *body \rangle
           \parformat 746 \renewcommand*{\thepar}{\arabic{par}}
\parformatseparation 747 \def\theHpar{\theH@AbsClause-\Roman{par}}
             \p@par 748 \rightarrow 748 \rightarrow 748
  \verb|\withoutparnumber| 749 \\ | \text{lowcommand*{parformatseparation}{\nobreakspace}}| \\
                     750 \newkomafont{parnumber}{}
                     751 \renewcommand*\p@par{{\contract@usetype{\contract@env@type}\theSubClause\contract@separator
                     752 \@addtoreset{par}{Clause}
                     753 \@addtoreset{par}{SubClause}
                     754 \newcommand*{\withoutparnumber}{}
                     755 (/body)
                     756 (*body)
                     757 \newcommand*{\contract@everypar}{%
                          \ifparnumber
                            \ifx\contract@special@par\relax
                     759
                     760
                               \ifx\contract@special@reset@par\relax\else
                     761
                                \global\let\thepar\contract@special@reset@par
                                \global\let\contract@special@reset@par\relax
                               \refstepcounter{par}%
                     764
```

```
\refstepcounter{sentence}%
                        765
                       766
                       767
                                 \ifx\contract@special@reset@par\relax
                       768
                                   \global\let\contract@special@reset@par\thepar
                       769
                                 \fi
                                 \global\let\thepar\contract@special@par
                       770
                                 \global\let\contract@special@par\relax
                       771
                                 \setcounter{sentence}{0}\refstepcounter{sentence}%
                       772
                       773
                       774
                               \begingroup
                                 \if@filesw
                       775
                                   \protected@write\@auxout{%
                       776
                        777
                                     \expandafter\let\csname \contract@env@type @Clauseformat\endcsname
                       778
                                     \@firstofone
                                   }{%
                       779
                                     \string\newmaxpar{\contract@env@type}%
                       780
                       781
                                                       {\csname the\contract@env@type
                                                         AbsoluteClause\endcsname}%
                       782
                                                       {\thepar}%
                       783
                                   }%
                       784
                        785
                                 \fi
                                 \getmaxpar\@tempa{\contract@env@type}%
                       786
                        787
                                                   {\csname the\contract@env@type AbsoluteClause\endcsname}%
                        788 (+trace)
                                         \typeout{Stored max is \@tempa}%
                       789
                                 790
                                 \afterassignment\reserved@a\@tempcnta=0\@tempa\relax\@nnil
                       791
                                 \ifnum \@tempcnta>\@ne
                                   {\usekomafont{parnumber}{\parformat\parformatseparation}}%
                       792
                       793
                                 \else
                       794
                                   \def\reserved@a{\relax}%
                       795
                                   \ifx\@tempa\reserved@a
                                     \withoutparnumber
                       796
                       797
                                     {\usekomafont{parnumber}{\parformat\parformatseparation}}%
                        798
                        799
                                   \fi
                       800
                                 \fi
                       801
                               \endgroup
                       802
                             \else
                               \begingroup\withoutparnumber\endgroup
                       803
                               \setcounter{sentence}{-1}\refstepcounter{sentence}%
                       804
                       805
                             \fi
                       806 }
                       807 (/body)
       \thisparnumber You can use this for manual paragraph numbering. But the number has to be fully expand-
\contract@special@par able!
                       808 \langle *body \rangle
```

809 \newcommand*{\thisparnumber}[1]{%\def\contract@special@par{#1}%

812 \newcommand*{\contract@special@par}{}

811 }

\contract@special@reset@par

```
813 \let\contract@special@par\relax
814 \newcommand*{\contract@special@reset@par}{}
815 \let\contract@special@reset@par\relax
816 \let\contract@special@reset@par\relax
```

18 Referencing

```
\refL Similar to \ref but always the long form.
\ensuremath{\mbox{\sc VrefQL}} 817 \ensuremath{\mbox{\sc 817}}
         818 \newcommand*{\refL}{\kernel@ifstar {\text{CL}}}}
         819 \newcommand*{\ref@L}[2]{\%
         820 \quad \verb|\begingroup|
                 \def\parcite@format{0}%
         821
                 \let\sentencecite@format\parcite@format
         822
                 \ref#1{#2}%
         823
         824
             \endgroup
         825 }
 \refS Similar to \ref but always the short form.
\label{lem:command*{refS}} $$ $$ 826 \end{substitute} $$ 826 \end{substitute} $$ $$ $$ $$ $$ $$ $$
         827 \newcommand*{\ref@S}[2]{%
              \begingroup
                 \def\parcite@format{1}%
                 \let\sentencecite@format\parcite@format
         831
                 \ref#1{#2}%
         832
              \endgroup
         833 }
 \refN Similar to \ref but always the numerical form.
\label{lem:command*{refN}} $$ 1834 \encommand*{refN}{\kappa {\refN}} $$
         835 \newcommand*{\ref@N}[2]{%
         836
              \begingroup
                 \def\parcite@format{2}%
         837
                 \let\sentencecite@format\parcite@format
         838
                 \ref#1{#2}%
         839
         840
              \endgroup
         841 }
```

\refClause Reference only the clause of a clause, paragraph or sentence. For better compatibility with \refClause hyperref there is also a star version if hyperref is used. Without hyperref the star version is nonsense.

```
842 \newcommand*{\refClause}{%

843 \kernel@ifstar {\ref@Clause*}{\ref@Clause{}}

844 }

845 \newcommand*{\ref@Clause}[2]{%

846 \expandafter\ifx\csname r@#2\endcsname\relax

847 \ref#1{#2}%

848 \else

849 \begingroup
```

Copy all parts of the reference but the first one to \\c_atsign_strtempb.

```
\expandafter\expandafter\expandafter
850
         \expandafter\expandafter\def
851
         \expandafter\expandafter\expandafter
852
         \expandafter\expandafter\@tempb
853
         \expandafter\expandafter\expandafter
854
         \expandafter\expandafter\expandafter{%
855
           \expandafter\expandafter\expandafter\@gobble\csname r@#2\endcsname}%
856
Copy the first part of the reference to \\c_atsign_strtempa.
         \def\@tempc##1##2\@nil{##1}%
857
         \let\contract@separator\@gobble
858
         \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
859
           \csname r@#2\endcsname\noexpand\@nil}%
860
Copy the first part of \\c_atsign_strtempa to \\c_atsign_strtempb.
861
         \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
862
           \@tempa\@nil}%
863
         \let\@@protect\protect
         \let\protect\noexpand
864
         \expandafter\edef\csname r@#2\endcsname{{\@tempa}\@tempb}%
865
         \let\protect\@@protect
866
         \ref#1{#2}%
867
       \endgroup
868
     \fi
869
870 }
```

\refClauseN Reference only the clause number of a clause, a paragraph or a sentence. For improved \refClauseN compatibility with hyperref there is also a star version if hyperref is used. Without hyperref the star version is nonsense.

```
871 \newcommand*{\refClauseN}{%
872 \kernel@ifstar {\ref@ClauseN*}{\ref@ClauseN{}}
873 }
874 \newcommand*{\ref@ClauseN}[2]{%
875 \begingroup
876 \let\Clauseformat\relax
877 \ref@Clause{#1}{#2}%
878 \endgroup
879 }
```

\refPar References only the paragraph of a paragraph or sentence. For improved compatibility with \ref@Par hyperref there is also a star version if hyperref is used. Without hyperref the star version is nonsense.

```
880 \newcommand*{\refPar}{%
881 \kernel@ifstar {\ref@Par*}{\ref@Par{}}
882 }
883 \newcommand*{\ref@Par}[2]{%
884 \expandafter\ifx\csname r@#2\endcsname\relax
885 \ref#1{#2}%
886 \else
887 \begingroup
```

```
\expandafter\expandafter\expandafter
          888
           889
                    \expandafter\expandafter\def
          890
                    \expandafter\expandafter\expandafter
                    \expandafter\expandafter\expandafter\@tempb
          891
                    \expandafter\expandafter\expandafter
          892
                    \expandafter\expandafter\expandafter{%
          893
                      \expandafter\expandafter\expandafter\@gobble\csname r@#2\endcsname}%
          894
          Copy the first part of the reference to \\c_atsign_strtempa.
                    \def\@tempc##1##2\@nil{##1}%
          895
                    \let\contract@separator\@gobble
          896
                    \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
          897
                      \csname r@#2\endcsname\noexpand\@nil}%
          898
          Copy the second part of \\c_atsign_strtempa to \\c_atsign_strtempa ablegen.
                    \def\@tempc##1##2##3\@nil{##2}%
          899
                    \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
          900
                      \@tempa{%
          901
                        \protect\G@refundefinedtrue
          902
                        \nfss@text{\reset@font\bfseries ??}%
          903
                        \@latex@warning{Reference '#2' on page \thepage \space
          904
          905
                          with undefined par number}%
                      }\noexpand\@nil}%
          907
                    \let\@@protect\protect
          908
                    \let\protect\noexpand
                    \expandafter\edef\csname r@#2\endcsname{{\@tempa}\@tempb}%
          909
                    \let\protect\@@protect
          910
                    \ref#1{#2}%
          911
                  \endgroup
          912
                \fi
          913
          914 }
\refParL The same but long.
\ref@ParX
          915 \newcommand*{\refParL}{%
          916
               \kernel@ifstar {\ref@ParX0*}{\ref@ParX0{}}
          917 }
          918 \newcommand*{\ref@ParX}[3]{%
          919
               \begingroup
                  \def\parcite@format{#1}%
          920
          921
                  \let\sentencecite@format\parcite@format
          922
                  \ref@Par{#2}{#3}%
                \endgroup
          923
          924 }
\refParS The same but short.
          925 \newcommand*{\refParS}{%
               \kernel@ifstar {\ref@ParX1*}{\ref@ParX1{}}
          927 }
```

Copy all parts of the reference but the first one to \\c_atsign_strtempb.

```
\refParN The same but numerical.
\ref@ParN
           928 \newcommand*{\refParN}{%
\ref@@ParN
                 \kernel@ifstar {\ref@ParN2*}{\ref@ParN2{}}
            930 }
            931 \newcommand*{\ref@ParN}[2]{%
                 \kernel@ifnextchar [%]
            932
                   {\ref@@ParN{#1}{#2}}%
            933
                   {\ref@ParX{#1}{#2}}%
            934
            935 }
            936 \newcommand*{\ref@@ParN}{}
            937 \def\ref@@ParN#1#2[#3]#4{%
                 \begingroup
                   \renewcommand*{\parnumericformat}[1]{%
            940
                     \csname @#3\endcsname{\number ##1\relax}%
            941
                   \ref@ParX{#1}{#2}{#4}%
            942
                 \endgroup
            943
            944 }
```

\refSentence Reference only the sentence of a sentence. For improved compatibility with hyperref there \ref@Sentence is also a star version if hyperref is used. Without hyperref the star version is nonsense.

```
945 \newcommand*{\refSentence}{%
     \kernel@ifstar {\ref@Sentence*}{\ref@Sentence{}}
946
947 }
948 \newcommand*{\ref@Sentence}[2]{%
     \expandafter\ifx\csname r@#2\endcsname\relax
950
        \ref#1{#2}%
951
     \else
952
        \begingroup
Copy all parts of the reference to \\c_atsign_strtempb.
953
          \expandafter\expandafter\expandafter
954
         \expandafter\expandafter\def
955
         \expandafter\expandafter\expandafter
956
          \expandafter\expandafter\expandafter\@tempb
957
          \expandafter\expandafter\expandafter
          \expandafter\expandafter\expandafter{%
958
            \expandafter\expandafter\expandafter\@gobble\csname r@#2\endcsname}%
959
Copy the first part of the reference to \\c_atsign_strtempa.
         \def\@tempc##1##2\@nil{##1}%
960
         \let\contract@separator\@gobble
961
          \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
962
            \csname r@#2\endcsname\noexpand\@nil}%
963
Copy the third part of \\c_atsign_strtempa to \\c_atsign_strtempa.
          \def\@tempc##1##2##3##4\@nil{##3}%
964
          \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
965
            \@tempa{}{%
966
              \protect\G@refundefinedtrue
967
968
              \nfss@text{\reset@font\bfseries ??}%
              \@latex@warning{Reference '#2' on page \thepage \space
969
```

```
with undefined sentence number}%
                    971
                                }\noexpand\@nil}%
                              \let\@@protect\protect
                    972
                    973
                              \let\protect\noexpand
                    974
                              \expandafter\edef\csname r@#2\endcsname{{\@tempa}\@tempb}%
                    975
                              \let\protect\@@protect
                              \ref#1{#2}%
                    976
                    977
                            \endgroup
                    978
                         \fi
                    979 }
     \refSentenceL The same but long.
    \ref@SentenceX
                    980 \newcommand*{\refSentenceL}{%
                          \kernel@ifstar {\ref@SentenceX0*}{\ref@SentenceX0{}}
                    981
                    982 }
                    983 \newcommand*{\ref@SentenceX}[3]{%
                          \begingroup
                    984
                            \def\parcite@format{#1}%
                    985
                            \let\sentencecite@format\parcite@format
                            \ref@Sentence{#2}{#3}%
                    987
                    988
                          \endgroup
                    989 }
     \refSentenceS The same but short.
                    990 \newcommand*{\refSentenceS}{%
                          \kernel@ifstar {\ref@SentenceX1*}{\ref@SentenceX1{}}
                    992 }
     \refSentenceN The same but numeric.
                    993 \newcommand*{\refSentenceN}{%
                         \kernel@ifstar {\ref@SentenceX2*}{\ref@SentenceX2{}}
                    995 }
                    996 (/body)
\contract@sentence Numbering of sentences.
    sentence (cnt.) The counter is used for numbering the sentences. It is important to add the paragraph as
      \thesentence parent object to labels. The original method to make it possible to use \thesentence as
     \theHsentence an argument of \p@sentence does not work any longer using LATFX 2019-10-01 or newer. It
       \p@sentence would result in an error message. So the code has to be adapted to the new definition of
                   \refstepcounter in LATEX 2019-10-01. From this version it uses \labelformat. Don't ask me,
                   what I think about the fact, that every new versions of LATEX can break existing packages
                   and package authors have to find out such incompatibilities on their own.
                    997 (*body)
                    998 \newcounter{sentence}[par]
```

999 \renewcommand*{\thesentence}\{\arabic{sentence}\} 1000 \def\theHsentence{\theHpar-\arabic{sentence}\} 1001 \scr@ifundefinedorrelax{labelformat}{%}

1002 \renewcommand*{\p@sentence}{\expandafter\p@@sentence}

```
\newcommand*{\p@@sentence}[1]{\p@par{{\par@cite{\thepar}}%
                                  \contract@separator{\nobreakspace}}{\sentence@cite{#1}}}%
                       1004
                       1005 }{%
                       1006
                             \labelformat{sentence}{\p@par{{\par@cite{\thepar}}%
                       1007
                                  \contract@separator{\nobreakspace}}{\sentence@cite{#1}}}%
                       1008 }
                       1009 \newcommand*{\contract@sentence}{%
                       For the numbering it is important not to increase the paragraph number at the very be-
                       ginning, because the paragraph already does so. To make this work, the paragraph has to
                       start before we print the number. But immediately after a minipage, a list or a \parbox we
                       should behave as not being at the beginning of a paragraph.
                       1010
                             \ifvmode
                       1011
                                \if@endpe
                       1012
                                  \refstepcounter{sentence}%
                       1013
                                \else
                                  \leavevmode
                       1014
                                \fi
                       1015
                       1016
                                \refstepcounter{sentence}%
                       1017
                       1018
                              {\usekomafont{sentencenumber}{\sentencenumberformat}}%
                       1019
                              \nobreak\hskip\z@
                       1020
                       1021 }
 sentencenumber (font) Formatting an font can be changed using font element sentencenumber and command
\sentencenumberformat \sentencenumberformat. The last has the preset \textsuperscript.
                       1022 \mbox{newkomafont{sentencenumber}{}}
                       1023 \verb|\newcommand*{\sentencenumberformat}{\textsuperscript{\thesentence}}|
                       1024 (/body)
            \par@cite Reference style for paragraphs.
       \parciteformat _{1025} \langle *body \rangle
                       1026 \verb|\DeclareRobustCommand*{\par@cite}[1]{\parciteformat{\#1}}}
                       1027 \newcommand*{\parciteformat}[1]{%
                             \ifcase \parcite@format
                       1028
                                \expandafter\parlongformat
                       1029
                       1030
                                \expandafter\parshortformat
                       1031
                       1032
                                \expandafter\parnumericformat
                       1033
                       1034
                                \unskip\expandafter\@gobble
                       1035
                             \fi
                       1036
                       1037
                              {#1}%
                       1038 }
       \sentence@cite Reference style for sentences. Preset is \\c_atsign_strarabic.
  \space{10} \ \sentenceciteformat \space{10} \ \DeclareRobustCommand*{\sentence@cite} [1] {\sentenceciteformat{#1}}
                       1040 \newcommand*{\sentenceciteformat}[1]{%
                       1041
                             \ifcase \sentencecite@format
```

\expandafter\sentencelongformat

1042

```
1043
                                   \expandafter\sentenceshortformat
                          1044
                          1045
                          1046
                                   \expandafter\sentencenumericformat
                          1047
                                 \else
                          1048
                                   \unskip\expandafter\@gobble
                                 \fi
                          1049
                                 {#1}%
                          1050
                          1051 }
         \parlongformat The six formattings.
        \label{lem:loss} $$ \operatorname{loss}_{1052} \end{*{\scriptstyle parlongformat}[1]_{\scriptstyle parname~\#1}} $$
     \parnumericformat 1053 \newcommand*{\parshortformat}[1]{\parshortname~#1}
   \sentencelongformat 1054 \newcommand*{\parnumericformat}[1]{\@Roman{\number #1\relax}}
  \sentenceshortformat 1055 \newcommand*{\sentencelongformat}[1]{\sentencename~#1}
\verb|\sentencenumericformat| 1056 \verb|\newcommand*{\sentenceshortformat}| [1] {\sentenceshortname~\#1} |
                          1057 \newcommand*{\sentencenumericformat}[1]{\@arabic{\number #1\relax}.}
                          1058 (/body)
```

19 Language Dependent Names

```
\parname The names of paragraphs and sentences and their short versions. The English names are
       \parshortname donated by "m.eik".
       \sentencename _{1059} \sentencename _{1059}
  \sentenceshortname 1060 \newcommand*{\parname}{Paragraph}
\contract@lang@error 1061 \AtBeginDocument{%
                            \providecaptionname{german,ngerman,austrian,naustrian}\parname{Absatz}%
                      1062
                            \providecaptionname{german,ngerman,austrian,naustrian}\parshortname{Abs.}%
                      1063
                            \providecaptionname{german,ngerman,austrian,naustrian}\sentencename{Satz}%
                      1064
                      1065
                            \providecaptionname{german,ngerman,austrian,naustrian}\sentenceshortname{S.}%
                            \providecaptionname{english,american,british,canadian,%
                      1066
                              USenglish, UKenglish, usenglish, ukenglish}\parname{paragraph}%
                      1067
                            \providecaptionname{english,american,british,canadian,%
                      1068
                      1069
                              USenglish, UKenglish, usenglish, ukenglish}\parshortname{par.}%
                      1070
                            \providecaptionname{english,american,british,canadian,%
                      1071
                              USenglish, UKenglish, usenglish, ukenglish}\sentencename{sentence}%
                            \providecaptionname{english,american,british,canadian,%
                      1072
                      1073
                              USenglish,UKenglish,usenglish,ukenglish}\sentenceshortname{sent.}%
                      1074 }
                      1075 \providecommand*{\parname}{\contract@lang@error{\parname}}
                      1076 \providecommand*{\parshortname}{\contract@lang@error{\parshortname}}
                      1077 \providecommand*{\sentencename}{\contract@lang@error{\sentencename}}
                      1078 \providecommand*{\sentenceshortname}{\contract@lang@error{\sentenceshortname}}
                      1079 \newcommand*{\contract@lang@error}[1]{%
                      1080
                            \PackageError{contract}{%
                      1081
                              current language not supported%
                            }{%
                      1082
                              Currently contract only supports languages 'german', 'ngerman',
                      1083
                              'austrian',\MessageBreak
                      1084
                              'naustrian', 'english', 'american', 'british', 'canadian',
                      1085
```

```
'USenglish',\MessageBreak
        'UKenglish', 'usenglish', and 'ukenglish'.\MessageBreak
1087
        It seems, that you are using another language (maybe '\languagename') or
1088
1089
        that\MessageBreak
        your language selection isn't compatible to package 'babel'.\MessageBreak
1090
        Because of this you have to define '\string#1' by yourself!\MessageBreak
1091
        It would be nice if you'll send your definitions to the author.%
1092
     ጉ%
1093
      \textbf{??}%
1094
1095 }
1096 (/body)
```

20 Using Values from last LATEX Run

\newmaxpar Two helper macros, to save a counter in a aux-file and get the value back or another value \getmaxpar of it is not in the aux-file.

```
1097 (*body)
1098 \newcommand*{\newmaxpar}[3]{%
      \begingroup
1099
        \expandafter\let\csname #1@Clauseformat\endcsname\@firstofone
1100
        \protected@edef\@tempa{#2}\@onelevel@sanitize\@tempa
1101
        \expandafter\xdef\csname max@#1@\@tempa\endcsname{#3}%
1102
      \endgroup
1103
1104 }
1105 \newcommand*{\getmaxpar}[3]{%
      \begingroup
1106
        \expandafter\let\csname #2@Clauseformat\endcsname\@firstofone
1107
1108
        \protected@edef\@tempa{#3}%
1109
        \@onelevel@sanitize\@tempa
        \expandafter\ifx \csname max@#2@\@tempa\endcsname\relax
1110
          \edef\@tempa{\endgroup\edef\noexpand#1{\expandafter\the\value{par}}}%
1111
1112
        \else
          \edef\@tempa{\endgroup
1113
            \edef\noexpand#1{\csname max@#2@\@tempa\endcsname}}%
1114
1115
        \fi
      \@tempa
1116
1117 }
```

Because some users remove contract from their documents without deleting the aux-file, we add a fallback definition of \newmaxpar to the aux-file. This avoids error messages because of undefined \newmaxpar.

```
1118 \AtBeginDocument{%
1119 \if@filesw
1120 \immediate\write\@auxout{%
1121 \string\providecommand*\string\newmaxpar[3]{}
1122 }%
1123 \fi
1124 }
1125 \( /body \)
```

References

- Javier Bezos López and Johannes L. Braams. babel Multilingual support for LATEX, LualATEX, XALATEX and Plain TEX. Version 24.1. This package manages culturally-determined typographical (and other) rules for a wide range of languages. A document may select a single language to be supported, or it may select several, in which case the document may switch from one language to another in a variety of ways. Babel uses contributed configuration files that provide the detail of what has to be done for each language, as well as .ini files for about 300 languages from around the World, including many written in non-Latin and RTL scripts. Many of them work with pdflATEX, as well as with XALATEX and LualATEX, out of the box. A few even work with plain formats. Jan. 7, 2024. URL: https://ctan.org/pkg/babel (visited on 01/08/2024).
- [Koh20a] Markus Kohm. KOMA-Script. Eine Sammlung von Klassen und Paketen für $L^{A}T_{E}X$ 2_{ε} . 7. Aufl. Edition DANTE. Print-Ausgabe. Berlin: Lehmanns Media, 2020. ISBN: 978-3-96543-097-6.
- [Koh20b] Markus Kohm. KOMA-Script. Eine Sammlung von Klassen und Paketen für $L^{A}T_{E}X$ 2_{ε} . 7. Aufl. Edition DANTE. eBook-Ausgabe. Berlin: Lehmanns Media, 2020. ISBN: 978-3-96543-103-4.
- [Koh23a] Markus Kohm. KOMA-Script. Die Anleitung. 16. Juni 2023. URL: http://mirrors.ctan.org/macros/latex/contrib/koma-script/scrguide-de.pdf (besucht am 04.07.2023).
- [Koh23b] Markus Kohm. KOMA-Script. The Guide. June 16, 2023. URL: http://mirrors.ctan.org/macros/latex/contrib/koma-script/scrguide-en.pdf (visited on 07/14/2023).
- [Koh23c] Markus Kohm. KOMA-Script A bundle of versatile classes and packages. Version 3.41. The KOMA-Script bundle provides replacements for the article, report, and book classes with emphasis on typography and versatility. There is also a letter class. July 7, 2023. URL: https://ctan.org/pkg/koma-script (visited on 07/14/2023).
- [MFP21] Frank Mittelbach, Robin Fairbairns, and H. Partl. parskip Layout with zero \parindent, non-zero \parskip. Version 2.0h. Simply changing \parskip and \parindent leaves a layout that is untidy; this package (though it is no substitute for a properly-designed class) helps alleviate this untidiness. Mar. 14, 2021. URL: https://www.ctan.org/pkg/parskip (visited on 01/19/2024).
- [Mit21] Frank Mittelbach. The parskip package. Mar. 14, 2021. URL: http://mirrors.ctan.org/macros/latex/contrib/parskip/parskip.pdf (visited on 01/19/2024).

Change History

scrjura-v0.5b - 2010/04/05	scrjura-v0.5c - 2010/04/26
	\contract@everypar:
iuratitlepagebreak: new	\contract@Clauseformat expands

while writing to its own argument .	38	\paragraphformat: redefinition of	00
first argument of \newmaxpar and		\paragraphformatscrjura-v0.7 - 2013/04/18	28
second argument of $\general{\general}$ second $\general{\general}$ not $\general{\general}$ contractpars $\general{\general}$	38	\contract@everypar: using counter	
\getmaxpar: \\#2@Clauseformat	30	contractAbsoluteClause	38
expands to its argument	47	\contract@paragraph: increase counter	00
\protected@edef replaced by \edef	47	contractAbsoluteClause	32
scrjura-v0.5d – 2010/04/28		\contract@subparagraph: increasing	-
contract: defining \jura@env@type	28	counter contractAbsoluteClause .	34
missing error message added	28	contractAbsoluteClause: new	-
not self-redefining any more	28	(internal) counter for all clauses to	
\contract@dusetype: new	38	make it possible to reset the users	
\contract@env@type: new	27	clause counter	28
\contract@usetype: new	38	scrjura-v0.7 - 2013/04/28	
\p@par: added \jura@usetype with		\contract@paragraph: distance behind	
argument \jura@env@type	38	the number is part of the number .	33
\parciteformat: argument moved	45	\contract@subparagraph: distance after	
\sentenceciteformat: argument moved	45	number	34
scrjura-v0.5d – 2010/06/07		scrjura-v0.7 - 2013/05/02	
ref: new values nopar, nosentence,		\ellipsispar: new	27
OnlyParagraph	24	\parellipsis: new	27
sentencecitename: deprecated	24	scrjura-v0.7 – 2013/05/23	
scrjura-v0.5e – 2011/08/31		\contract@everypar: initializing	
\Clausemark: support for		sentence number for manually	
\MakeMarkcase	23	numbered paragraphs to 0 instead of	
scrjura-v0.6 - 2011/09/29		1, because first action of \Sentence	
\cpar@indent: new	21	is increasing the number	38
\if@juratotoc: replaced by counter	21	recognize paragraph number of	
juratocindent: new	21	\thisparnumber	38
juratoclevel: definition moved	21	$\verb \contract@special@reset@par: new .$	39
parnumber: new	22	\thisparnumber: new	39
\parnumberfalse: moved the definition		scrjura-v0.7 - 2013/06/06	
into the definition of the option	38	\contract@everypar: if paragraph	
scrjura-v0.6 - 2011/09/30		numbers are not completely	
\getmaxpar: argument no 3 is expanded		numerical always set the number	38
using \protected@edef	47	scrjura-v0.7 - 2013/06/07	
noentry: deprecated	31	\parformat: new font element	
notocentry: new	31	parnumber	38
scrjura-v0.6a - 2012/10/08		scrjura-v0.7 - 2013/06/09	
\contract@paragraph: missing \par		\toclevel@cpar: new, because of	
$added \dots \dots$	33	hyperref	21
\contract@subparagraph: missing \par		scrjura-v0.7 - 2013/09/19	
added	34	\contract@lang@error: Usage of	
scrjura-v $0.6a - 2012/10/15$		\PackageError instead of	
parnumber: value mistake message		\PackageErrorNoLine	46
changed	22	scrjura-v0.7 – 2013/11/04	
scrjura-v0.6b – 2013/04/16		clausemark: usage of renewed interface	20
\contract@paragraph: using		with \FamilyKeyState	23
\contract@Clauseformat for	99	juratocindent: usage of renewed	01
manual numbers too	3 3	interface with \FamilyKeyState	21
\newmaxpar: \\#1@Clauseformat	47	juratotoc: usage of renewed interface	21
expands to its arotiment	4/	WILL VEAULIVE AVSTATE	2.1

markboth: deprecated	23	contractSubClause	28
parnumber: usage of renewed interface		\thecontractClause: must use counter	
with \FamilyKeyState	22	Clause instead of contractClause .	28
ref: usage of renewed interface with		scrjura-v0.9c - 2015/05/13	
\FamilyKeyState	24	\@defjuraenvironment : defining	
scrjura-v0.7a - 2014/01/28		\theClause	30
\contract@sentence:		\contract@paragraph:	
\nobreak\hskip\z@added to allow		\thecontractClause replaced by	
hyphenation of the first word after	4.4	environment-dependent macro	33
the sentence mark		scrjura-v0.9e - 2015/11/03	
using \textsuperscript	44	\@gobble@IncludeInRelease: used	
scrjura-v0.7b - 2014/11/03		temporary	
\contract@@usetype: \jura@@usetype renamed	20	1 0	36
\contract@afterheading:	30	\IncludeInRelease: used temporary .	36
\jura@afterheading renamed	35	scrjura-v0.9e - 2015/11/04	
no argument	35	\p@sentence: first start the paragraph	
\contract@env@type: \jura@env@type	50	then print the number	45
renamed	27	scrjura-v0.9f - 2016/02/06	
\contract@everypar: more flexible	21	\contract@everypar: \nobreakspace	
replacement of		replaced by \parformatseparation	38
contractAbsoluteClause	38	\parformatseparation: new	38
\contract@paragraph: more flexible		scrjura-v0.9f - 2016/02/24	
$\langle environment \rangle$ AbsoluteClause	32	\@defjuraenvironment:	
\contract@separator:		\protect\\c_atsign_strnameuse	
\jura@separator renamed	38	instead of \csname\endcsname	30
\contract@used@everypar:		\contract@paragraph:	
\jura@everypar renamed	36	\protect\\c_atsign_strnameuse	22
\contract@usetype: \jura@usetype		instead of \csname\endcsname	99
renamed	38	scrjura-v0.9g - 2015/03/25	
\ellipsispar:		\getmaxpar: write to aux-file \if@filesw	47
\thecontractAbsoluteClause made			41
more flexible	27	scrjura-v0.9g – 2016/03/25	
scrjura-v0.7b - 2014/11/10		\contract@everypar: writing to aux-file \if@filesw	90
\getmaxpar: added fallback code to		\ellipsispar: write to aux-file only	30
aux-file	47	\ifOfilesw	27
scrjura-v0.7b – 2014/11/11		scrjura-v0.9h - 2016/04/11	41
General: General renaming of	00	clausemark: paragraphmark renamed to	
"Paragraph" into "Clause"	20	clausemark	23
$scrjura-v0.9-2014/11/04$ \@defjuraenvironment: new	20		20
	29	\DeclareNewJuraEnvironment:	
scrjura-v0.9 - 2014/11/12 \DeclareNewJuraEnvironment: new	29	\\c_atsign_strifnextchar	
scrjura-v0.9a - 2015/03/09	49	replaced by \kernel@ifnextchar	29
clausemark: internal value storage	23	\ref@ParN: \\c_atsign_strifnextchar	
juratocindent: internal value storage	21	replaced by \kernel@ifnextchar	43
juratoto: internal value storage	21	\\c_atsign_strifstar replaced by	
parnumber: internal value storage	22	\kernel@ifstar	43
ref: internal value storage	24	\refClause: \\c_atsign_strifstar	-0
scrjura-v0.9b – 2015/05/01		replaced by \kernel@ifstar	40
contractSubClause: must use counter		\refClauseN: \\c_atsign_strifstar	
SubClause instead of		replaced by \kernel@ifstar	41

\refL: \\c_atsign_strifstar replaced	scrjura-v3.27 - 2019/02/25	
by \kernel@ifstar 40	\contract@afterheading:	
\refN: \\c_atsign_strifstar replaced	\CheckCommand for	
by \kernel@ifstar 40	\c _atsign_strafterheading 35	5
\refPar: \\c_atsign_strifstar	\cpar@indent: replaced by usage of	
replaced by \kernel@ifstar 41	\DeclareTOCStyleEntry 21	1
\refParL: \\c_atsign_strifstar	juratoclevel: removed 21	1
replaced by \kernel@ifstar 42	\lambda l@cpar: usage of tocbasic 37	7
\refParS: \\c_atsign_strifstar	\toclevel@cpar: handled by tocbasic . 21	1
replaced by \kernel@ifstar 42	scrjura-v3.27 - 2019/10/09	
<pre>\refS: \\c_atsign_strifstar replaced</pre>	\p@sentence: adaption for $L^{A}T_{E}X$	
by \kernel@ifstar 40	2019-10-01	4
\refSentence: \\c_atsign_strifstar	scrjura-v3.28 - 2019/11/18	
replaced by \kernel@ifstar 43	\IncludeInRelease: \ifstr renamed	
\refSentenceL: \\c_atsign_strifstar	into \Ifstr 36	
replaced by \kernel@ifstar 44		2
\refSentenceN: \\c_atsign_strifstar	scrjura-v3.36 - 2022/01/25	
replaced by \kernel@ifstar 44	\@documentcontracttrue: switched to a	
\refSentenceS: \\c_atsign_strifstar	KOMA-Script option 21	1
replaced by \kernel@ifstar 44		
scrjura-v0.9i - 2017/02/23	(because of hooks) 28	8
\contract@paragraph: usage of	scrjura-v3.39 - 2022/11/11	
\ext@toc 33	clausemark: initial dot in member	
\contract@subparagraph: usage of	argument of option storage	_
\ext@toc 34	commands removed 25	3
scrjura-v1.0 – 2014/11/22	juratocindent: initial dot in member	
\ParagraphCompatibilityHacks: throw	argument of option storage	1
error for deprecated commands but	commands removed 21	1
only once	juratotoc: initial dot in member	
scrjura-v3.24 - 2017/05/29	argument of option storage	1
\contract@lang@error: adaption for	commands removed 21	1
language name bug in babel 3.10 46	parnumber: initial dot in member	
scrjura-v3.25 - 2017/12/08	argument of option storage commands removed	9
\@defjuraenvironment:	ref: initial dot in member argument of	_
\reserved@defjuraenvironment set	option storage commands removed . 24	1
back to \relax 29	scrjura-v3.39 – 2022/11/16	±
Font alias fixed	markboth: only with KOMA-Script 3 . 23	2
new command options ClauseFont . 29	sentencecitename: only with	J
using local instead of global command	KOMA-Script 3	1
options		1
\contract@paragraph@font: using	\contract@paragraph: update of	
element Clause instead of	\\c_atsign_strcurrentcounter	
contract.Clause	added	3
scrjura-v3.25 - 2017/12/19	\contract@subparagraph: update of	,
\withoutparnumber: new 38		
scrjura-v3.26 – 2018/07/20	added 34	4
\p@sentence: new font element	\theH@AbsClause: Ulrike Fischer	Ī
sentencenumber	•	
new formatting	\newcommand* 32	2
\sentencenumberformat 45	•	
\sentencenumberformat: new 45		8

\theHsentence: Ulrike Fischer requested	\contract@special@reset@par:	
to use \def instead of \newcommand* 44	\scrjura@special@par renamed	39
\theHSubClause: Ulrike Fischer	\scrjura@special@reset@par	
requested to use \def instead of	renamed	39
$\verb \newcommand* $	\contract@used@everypar:	
v0.0.1 - 2023-10-10	\scrjura@everypar renamed	36
General: new KOMA-Script spin-off 1	\contract@usetype: \scrjura@usetype	
\contract@usetype:	renamed	38
\scrjura@usetype renamed 38	\if@contract@skiphyperref:	
\contract@afterheading:	\if@scrjura@skiphyperref	
\scrjura@afterheading renamed . 35	renamed	28
\contract@doendpe: \scrjura@doendpe	v0.0.2 - 2024/01/23	
renamed	\contract@paragraph: the argument is	
\contract@env@type:	optional	32
\scrjura@env@type renamed 27	\contract@subparagraph: the argument	
\contract@lang@error:	is optional	34
\scrjura@lang@error renamed 46	v0.9 - 2024-02-02	
\contract@separator:	General: first release as standalone	
\scrjura@separator renamed 38	package	-
J 1	1 0	

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	\contractSetup
@AbsClause (cnt.) 498	\DeclareNewJuraEnvironment 12 , 352
	$\DeprecatedParagraph \dots 245$
\mathbf{A}	\ellipsispar
\AutoPar <u>627</u>	\getmaxpar <u>1097</u>
	\ManualPar <u>627</u>
\mathbf{C}	\newmaxpar <u>1097</u>
\Clause 5	\Paragraph <u>245</u>
clause	$\ParagraphCompatibilityHacks 245$
clause	\paragraphformat <u>318</u>
\Clauseformat	\parciteformat <u>1025</u>
\Clausemark <u>91</u>	\parellipsis
clausemark $(opt.)$ $\underline{91}$	\parformat 8, <u>744</u>
$clausemark=\langle value \rangle \ (opt.) \ \dots \ 7$	\parformatseparation
clausemark=both $(opt.)$	\parlongformat <u>1052</u>
clausemark=false $(opt.)$	\parname
clausemark=forceboth $(opt.)$	\parnumericformat $\underline{1052}$
clausemark=forceright $(opt.)$	\parshortformat <u>1052</u>
clausemark=right $(opt.)$	\parshortname <u>1059</u>
Commands:	\partshortname 12
\AutoPar	\ref 9
\Clause <u>5</u>	\refClause 9, <u>842</u>
\Clauseformat	\refClauseN 9, <u>871</u>
\Clausemark <u>91</u>	\refL 9, <u>817</u>

\refN 9, 834	\DeprecatedParagraph 245
\refPar 10, 880	$\frac{1}{\text{dummy }(opt.)} \dots \frac{3}{493}$
\refParagraph 245	
\refParagraphN 245	${f E}$
\refParL 10, 915	\ellipsispar
\refParN 10, 928	entry $(opt.)$ $\underline{466}$
\refParS 10, 925	Environments:
\refS	contract <u>318</u>
\refSentence	environments:
\refSentenceL 10, 980	contract 5
\refSentenceN	\mathbf{G}
\refSentenceS	\getmaxpar 1097
\Sentence 9	(getmaxpai
\sentenceciteformat 1039	H
\sentencelongformat $\dots \dots \overline{1052}$	head $(opt.)$ $\underline{466}$
\sentencename	, <u> </u>
\sentencenumberformat $\dots \dots g$, $\overline{1022}$	J
\sentencenumericformat $\dots \dots 1052$	juratitlepagebreak $(opt.)$
\sentenceshortformat $\dots \dots \overline{1052}$	$\texttt{juratitlepagebreak=} \langle \textit{boolean} \rangle \ (\textit{opt.}) \ \dots \ \ prop:prop:opt:prop:o$
\sentenceshortname	juratocindent $(opt.)$ $\underline{49}$
\SubClause 5	$juratocindent = \langle length \rangle (opt.) \dots 4$
\SubParagraph <u>245</u>	juratoclevel ($cnt.$) 33
\thecontractClause 318	juratocnumberwidth $(opt.)$ $\underline{49}$
\thecontractSubClause 318	juratocnumberwidth= $\langle length \rangle (opt.)$. 4, 15
\theHClause 498	juratotoc (<i>opt.</i>)
\theHpar <u>744</u>	juratotoc=\langle (opt.) 4
\theHsentence 997	$juratotoc=\langle integer \rangle \ (opt.)$
\theHSubClause <u>498</u>	K
\thepar	KOMA-Script font elements:
\thesentence	contract.Clause
\thisparnumber 808	sentencenumber 1022
\withoutparnumber	<u></u>
contract	${f L}$
$\mathtt{contract}\ (\mathit{env.}) \dots \dots \underline{318}$	Lengths (internal):
$\mathtt{contract}\ (\mathit{opt.})\ \ldots\ldots\ldots\ldots \ 5,\ \underline{19}$	\cpar@indent $\underline{49}$
contract (env.) 5	\cpar@numberwidth $\underline{49}$
contract.Clause $(font)$	M
$contractAbsoluteClause\ (cnt.)\ \dots \ \underline{318}$	\ManualPar 627
$\texttt{contractClause} \ (\textit{cnt.}) \dots \underline{318}$	markboth (opt.)
\contractSetup 3	markright (opt.)
contractSubClause $(cnt.)$ 318	markinght (opt.)
Counters:	${f N}$
@AbsClause <u>498</u>	\newmaxpar 1097
$contractAbsoluteClause \dots 318$	noentry (opt.)
contractClause 318	nohead $(opt.)$ $\underline{466}$
contractSubClause 318	notocentry (opt.) 466
juratoclevel	$\verb"number" (opt.) $
par	
sentence $0, 997$	0
D	Options:
D	clausemark 91
\DeclareNewJuraEnvironment 12, 352	clausemark= $\langle value \rangle$

7	D
clausemark=both 7	P
clausemark=false 7	par (cnt.)
clausemark=forceboth	\Paragraph 245
clausemark=forceright 7	paragraph see clause
clausemark=right 7	numbering
contract 5, <u>19</u>	\ParagraphCompatibilityHacks 245
dummy	\paragraphformat 318
entry	paragraphmark (opt.) 91
head	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
juratitlepagebreak	parcitename (opt .)
juratitlepagebreak=\langle boolean \rangle \docs \docs \frac{7}{2}	\parellipsis
juratocindent <u>49</u>	\parformat
$juratocindent = \langle length \rangle \dots 4$	\parformatseparation
juratocnumberwidth 49	\parlongformat 1052
juratocnumberwidth= $\langle length \rangle$ 4, 15	\parname
juratotoc	parnumber $(opt.)$
juratotoc=\langle boolean \rangle \display \disp	$parnumber=\langle value \rangle \ (opt.) \ \dots \$
$juratotoc = \langle integer \rangle \dots 4$	parnumber=auto (opt.)
markboth <u>91</u>	parnumber=false (opt.)
markright <u>91</u>	parnumber=manual (opt.)
noentry <u>466</u>	parnumber=true (opt.)
nohead $\underline{466}$	\parnumericformat <u>1052</u>
notocentry $\dots \qquad \underline{466}$	\parshortformat <u>1052</u>
number <u>488</u>	\parshortname <u>1059</u>
paragraphmark <u>91</u>	parskip (opt.)
parcitename <u>153</u>	\partshortname 12
parnumber	postskip (opt.)
$parnumber = \langle value \rangle \dots $	preskip (opt.)
parnumber=auto	.
parnumber=false 7	R
parnumber=manual 8	\ref 9
parnumber=true $\dots $ 7	ref (opt.)
parskip	$ref=\langle value \rangle \ (opt.) \ \dots \ 10$
${\tt postskip} \dots \underline{489}$	ref=long (opt.)
preskip <u>489</u>	ref=numeric (opt.)
$\texttt{ref} \dots \underline{153}$	ref=parlong (opt.)
ref=(value) 10	ref=paroff $(opt.)$
ref=long	ref=parshort (opt.)
ref=numeric <u>11</u>	ref=sentencenumeric $(opt.)$
ref=parlong 11, 11	ref=sentenceoff (opt.)
ref=parnumeric <u>11</u>	ref=sentenceshort (opt.)
ref=paroff 11	ref=value (opt.)
ref=parshort <u>11</u>	\refClause
ref=sentencenumeric 11	\refClauseN
ref=sentenceoff 11	\refL 9, 817
ref=sentenceshort 11	\refN
ref=value <u>11</u>	\refPar
sentencecitename <u>153</u>	\refParagraph 245
title <u>466</u>	\refParagraphN 245
tocentry 466	\refParL 10, 915

\fDN	\+
\refParN	\contract@special@reset@par 808
· —	\contract@subparagraph 566
\refS	\contract@used@everypar 670
\refSentence	\contract@usetype
\refSentenceL	\EndIncludeInRelease
\refSentenceN	\if@contract@skiphyperref 318
\refSentenceS 10, 990	\if@documentcontract 19
${f s}$	\if@juratitlepagebreak 63
section see clause	\if@juratotoc 33
\Sentence 9	\ifcontract@dummy 493
sentence	\ifparnumber 744
number	\IncludeInRelease 671
sentence (cnt.)	\1@cpar
\sentenceciteformat 1039	\p@par
sentencecitename $(opt.)$	\p@sentence <u>997</u>
\sentenceCitename ($opt.$) \sentencelongformat \sum \frac{1052}{2}	\par@cite <u>1025</u>
\sentencerongrormat \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	\parcite@format <u>153</u>
sentencenumber $(font)$	\parnumberfalse <u>744</u>
\sentencenumber(yout) \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	\parnumbertrue 744
\sentencenumericformat	\ref@@ParN <u>928</u>
\sentenceshortformat	\ref@Clause <u>842</u>
\sentenceshortromat \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	\ref@ClauseN
\SubClause	\ref@L <u>817</u>
	\ref@N <u>834</u>
\SubParagraph <u>245</u>	\ref@Par
${f T}$	\ref@ParN <u>928</u>
TEX macros (internal):	\ref@ParX <u>915</u>
\@defjuraenvironment 371	\ref@S <u>826</u>
\@documentcontractfalse 19	\ref@Sentence <u>945</u>
\@documentcontracttrue 19	\ref@SentenceX <u>980</u>
\@doendpe 671	\sentence@cite <u>1039</u>
\@gobble@IncludeInRelease 671	\sentencecite@format $\dots 153$
\@juratitlepagebreakfalse 63	$\verb \theH@AbsClause 498 $
\@juratitlepagebreaktrue 63	\toclevel@cpar <u>33</u>
\contract@@usetype 740	\thecontractClause 318
\contract@afterheading 636	\thecontractSubClause 318
\contract@Clauseformat 318	\theHClause 498
\contract@doendpe 671	\theHpar 744
\contract@env@type $\dots \dots \dots$	\theHsentence 997
\contract@everypar $\dots \frac{744}{}$	\theHSubClause 498
\contract@lang@error 1059	\thepar
\contract@paragraph 466	\thesentence
\contract@paragraph@font 495	\thisparnumber <u>808</u>
\contract@postskip 489	title (opt.)
\contract@preskip 489	tocentry (opt.)
\contract@sentence 997	J (· F ·)
\contract@separator	TT 7
	${f W}$
\contract@special@par 808	\withoutparnumber