

XASSOCCNT

Associated counters stepping simultaneously

Version 0.2

11/14/2015

Author: Christian Hupfer[†]

Contents

1	Introduction	3
2	Requirements and incompatibilities	4
2.1	Required packages	4
2.2	Incompatibilities	4
2.3	Package options	4
3	Documentation of Macros	5
3.1	Association macros	5
3.2	Driver macros	9
3.3	Query macros	9
3.3.1	\IfIsDocumentCounter-Queries	12
3.4	Macros about the reset list	13
3.5	Information macros	14
4	Suspending and Resuming	16
5	Additions to standard commands	18
6	To - Do list	20

[†]christian.hupfer@yahoo.de

7 Acknowledgments	21
8 Version history	22
A Total number of sections	23
B Subsection with suspension	23
B.1 First dummy subsection	23
B.2 Second dummy subsection	23
B.3 Third dummy subsection after removing the associated counter	23
B.4 Suspension of a non-associated counter	24
Index	25

Preface

This package is the successor and a complete rewrite of `assoccnt`. Not all features of that package are implemented yet – if some functionality of your document depends on `assoccnt`, continue using the older version and shift gradually to `xassoccnt` please.

! Most times class and package authors will benefit of this package, but there might be usual documents that need the features of `—xassoccnt—`

1 Introduction

The aim of this package is to provide some additional support for example for a package like `totcount`. For example, the total number of pages in a document could be achieved by using

```
%
\regtotcounter{page}
...
The number of pages in the document is \number\totvalue{page} page(s) -- but in fact 2
    {it has \total{totalpages} pages.

... The number of pages in the document is 26 page(s) – but in fact it has 25 pages.
```

This will work, as long there is no reset of the page counter, as it might happen in the case of `\setcounter` or `\pagenumbering` being applied in the document. The result is a false page counter total value.

This package provides associate counters, i.e. counters that are increased simultaneously with a driver counter and are not influenced by a resetting of the driver counter, as long as not being added to the reset list by definition of the counter or explicitly by `\@addtoreset`.

This package defines some macros to handle associated counters. The only interception to the standard behaviour is within the redefined commands `\addtocounter` and `\stepcounter`. The usual commands still work, as there is code added to their definition. In a previous version `\refstepcounter` was redefined, but since these use `\addtocounter` effectively, it was decided to use the basic command.




Internally, the associated counters are stored in one list per counter – it is not recommended to operate on those lists directly.

Please note, that this package does not provide means for simultaneous stepping of counters defined by plain T_EX `\newcount` command.











2 Requirements and incompatibilities




2.1 Required packages



-  `xcolor`
-  `xparse`
-  `l3keys2e`

2.2 Incompatibilities

- This package does not work really well with the  `calc` package if that one is loaded after  `xassoccnt`. Load  `calc` before  `xassoccnt`!

 As of version 0.2  `xassoccnt` will abort compilation if  `calc` is loaded after this package, but will issue a warning only if  `calc` is loaded first.

-  `xassoccnt` and  `perpage` are not compatible completely. As of version 0.2 it is not possible to use the command `\AddAbsoluteCounter` from  `perpage` – this feature is provided already by this package.

 It is not recommended to use counters under control of  `perpage` with the association method!

 Updated:
2015-11-10

New:
2015-11-10

2.3 Package options

As of version 0.2  `xassoccnt` supports the option

`autodefinecounters`= $\langle true, false \rangle$ (initially false)

Setting this package option to `true`, all counters used with the special package macros will be autodefined, unless disabled locally. See the commands

- `\DeclareAssociatedCounters`^{→P.5}
- `\AddAssociatedCounters`^{→P.6}

3 Documentation of Macros

The main purpose of this package is co-stepping of counters, but there are some helper commands in addition to macros provided $\text{\LaTeX 2}_{\epsilon}$ already, see section Additions to standard commands.

- Section Association macros describes the most important macros for setting up associated counters
- Section Driver macros informs about the macros for setting up, removing or clearing driver counters
- Section Query macros deals with query command sequences about counters being a driver or an associated counters
- Section Information macros contains routines that show which counters have been changed last

3.1 Associated counters commands

All macros have the general rule, that the driver counter is specified as 1st mandatory argument to the macro, which is in almost case the 2nd argument of the macro

`\DeclareAssociatedCounters` [*options*] {*driver counter*} {*associated counters list*}

This command is the main macro of the package. It declares the counter names being specified in comma - separated - list (CSV) which should be stepped simultaneously when the driver counter is increased by `\stepcounter`. If only counter is to be associated, omit a trailing " , "!

#1 [*options*]:

`autodefine`=*choice* (initially none)

This choice - key can be specified if the specified counters should be defined if they not already available. Possible values are

- none – no counter is autodefined
- all – all counters will be autodefined
- driver – only driver counters will be autodefined
- associated – only associated counters will be autodefined

Default is none

`sloppy`

If `autodefine` key is used, the `sloppy` key disables the check whether a counter is defined already.

#2 {*driver counter*}

Holds the name of the driver counter to which the list of counters should be associated

#3 `{\associated counters list}`

A comma separated list of counter names that should be associated to the driver counter

- This command is a preamble command, i.e. it can be used in the preamble of the document or within other packages or class files only.
- This command should be used as early as possible, i.e. in the preamble of the document, since the driven counters are not increased as long as they are not associated to the driver counter. On the hand, it is possible or may be required to control the starting point of the association at any position in the body of the document, when the association should start later on. Use the command `\AddAssociatedCounters` if counters should be associated within the document body.

```

%%% The association of another total pages in this example just takes place here,
    (so the stepping of the counter will start from here and providing a 'wrong'
    value.
%%%
\DeclareAssociatedCounters{page}{totalpages,another totalpages}%
This document has \number\totvalue{totalpages} (note:
    (\number\totvalue{another totalpages}) pages.

```

This document has 25 (note: 6) pages.

- Current version state:
 - No checking whether the 2nd and 3rd arguments hold counter names is applied.
 - Mutually cross - association of two counters is not supported! The compilation will stop on this!
- A self-association of the driver counter to itself is ignored internally as this would lead to inconsistent counter values.
- The order of the specification of associated counters in the 2nd arguments is of no importance.
- Specifying an associated counter name multiple times has no effect, only the first occurrence of the name will be used.

`\AddAssociatedCounters` [*options*] `{\driver counter}` `{\associated counters list}`

The usage of this macro is similar to `\DeclareAssociatedCounters`^{→P.5}; if it is called in the document preamble (or in package file), `\AddAssociatedCounters` falls back to

`\DeclareAssociatedCounters`^{→P.5},

having the same optional argument functionality with `autodefine`^{→P.5} and `sloppy`^{→P.5}; if it is called in the document body, this command adds some counters to the associated counter list for a specific driver counter – if this list does not exist, the \LaTeX run will issue a warning, but add the driver counter to the driver list and the associated counters analogously.

Using `\AddAssociatedCounters`^{→P.6} in the document body automated generation of counters is disabled.



Description of arguments of command `\AddAssociatedCounters`

#1 [*options*]: As of version 0.2, the optional argument [*options*] are the same as for `\DeclareAssociatedCounters`^{→P.5}, see `autodefine`^{→P.5} and `sloppy`^{→P.5}.

#2 {*driver counter*}

Holds the name of the driver counter to which the list of counters should be associated

#3 {*associated counters list*}

A comma separated list of counter names that should be associated to the driver counter

`\RemoveAssociatedCounter`{*driver counter*}{*associated counter*}

This command removes a counter from the existing list for a driver counter, i.e. the counter will not be increased any longer by `\stepcounter`. It can be increased however manually, of course.

```
\RemoveAssociatedCounter{page}{anothertotalpages}
This document has \number\totvalue{totalpages} (beware: 2
  \number\totvalue{anothertotalpages}) pages.
```

This document has 25 (beware: 6) pages.

`\RemoveAssociatedCounters`{*driver counter*}{*list of associated counters*}

This command removes the comma-separated-value list of counters from the existing list for a driver counter, i.e. the counters will not be increased any longer by `\stepcounter`. They can be increased however manually, of course.

Take care not to confuse the commands `\RemoveAssociatedCounters` and `\RemoveAssociatedCounter`

`\ClearAssociatedCounters`[*options*]{*driver counter*}

This command clears the internal list for all counters associated to the {*driver counter*}. The counters will not be increased automatically any longer.

The optional argument is not used as of version 0.2.

Please note that the driver counter is not removed from the list of driver counters – this simplifies reassociating of (other) counters to this one later on with the macro `\AddAssociatedCounters`^{→P.6} and suppress the relevant warning.

If the driver counter and all its associated counters should be removed, use `\RemoveDriverCounter`^{→P.9} instead.

3.2 Driver counter commands

`\AddDriverCounter` [*<options>*] {*<driver counter name>*}

Description of arguments of command `\AddDriverCounter`

#1 [*<options>*]: As of 0.2, the optional argument [*<options>*] is not used so far, but is reserved for later purposes.

#2 {*<driver counter name>*}

Holds the name of the driver counter that should be added to the list of driver counters.

`\RemoveDriverCounter` [*<options>*] {*<driver counter>*}

This command clears the internal list for all counters associated to the {*<driver counter>*}. The counters will not be increased automatically any longer.

The optional argument is not used as of version 0.2.

If all driver counters should be unregistered, use `\ClearDriverCounters` instead!

`\ClearDriverCounters` [*<options>*]

This clears completely the list of driver counters, such that no counters are regarded as being associated – i.e. no driver is hold as being a driver counter.

The optional argument is not used as of version 0.2.

3.3 Commands for queries

Sometimes it might be necessary to get information, whether a counter is regarded as a driver or as an associated counter. This section describes some query macros in order to obtain this information.

`\IsAssociatedToCounter` {*<driver counter>*} {*<associated counter>*} {*<True branch>*} {*<False branch>*}

This macro checks, whether a counter is associated to a particular given driver counter and expands the corresponding branch. If the internal driver counter list does not exist, the false branch will be used, since this also means, that the possibly associated counter is not associated at all.

Description of arguments of command `\IsAssociatedToCounter`**#1** `{\langle driver counter \rangle}`

Holds the name of the driver counter to which `{\langle associated counter \rangle}` the could possibly be associated.

#2 `{\langle associated counter \rangle}`

Contains the name of the possibly associated counter.

#3 `{\langle True branch \rangle}`

This code is expanded if the counter is associated to the driver, otherwise it is ignored.

#4 `{\langle True branch \rangle}`

This code is expanded if the counter is **not** associated to the driver, otherwise it is ignored.

```
% Remove associated counter first for demonstration purposes
\RemoveAssociatedCounter{page}{anothertotalpages}
\IsAssociatedToCounter{page}{totalpages}{Yes, totalpages is associated}{No, }
  {totalpages is not associated}

\IsAssociatedToCounter{page}{anothertotalpages}{Yes, anothertotalpages is }
  {associated}{No, anotherpages is not associated}

Yes, totalpages is associated
No, anotherpages is not associated
```

See also

- `\IsAssociatedCounter`^{→P.11} for checking whether a counter is associated
- `\IsDriverCounter`^{→P.11} in order to check whether a counter is a driver.
- `\GetDriverCounter` returns the driver counter name for a given associated counter name

`\GetDriverCounter``{\langle counter name \rangle}`

This commands returns the driver counter to which the counter name of the first argument is connected to. If the counter is not defined, the macro returns nothing.

- No check whether the counter name is defined is performed

- No check whether the counter is associated at all is performed. Usage of this command in conjunction with `\IsAssociatedCounter`^{→P.11} is strongly encouraged.

```
%
totalpages is associated to the ›
    {\textcolor{blue}{\textbf{\GetDriverCounter{totalpages}}}} counter.
% Try with an undefined counter name
humptydumpty is associated to the ›
    {\textcolor{blue}{\textbf{\GetDriverCounter{humptydumpty}}}} counter.
```

totalpages is associated to the **page** counter. humptydumpty is associated to the – counter.

`\IsAssociatedCounter`{*<counter name>*}{*<True branch>*}{*<False branch>*}

This commands tests, whether a given counter name is an associated counter and expands correspondingly the true or the false branch. The command does not tell to which driver the counter it is associated – this information can be obtained by `\GetDriverCounter`^{→P.10}.

Description of arguments of command `\IfAssociatedCounter`

#1 {*<counter name>*}

Contains the name of the possibly associated counter

#2 {*<True branch>*}

This code is expanded if the counter is associated to a driver, otherwise it is ignored

#3 {*<True branch>*}

This code is expanded if the counter is **not** associated a driver, otherwise it is ignored

```
\IsAssociatedCounter{section}{Yes, section is an associated counter}{No, section ›
    {counter does not have the associated counter properties}
\IsAssociatedCounter{totalpages}{Yes, totalpages is an associated counter}{No, ›
    {totalpages counter does not have the associated counter properties}}
```

No, section counter does not have the associated counter properties Yes, totalpages is an associated counter

`\IsDriverCounter`{*<driver counter name>*}{*<True branch>*}{*<False branch>*}

This commands tests, whether a given counter name is a driver counter and expands correspondingly the true or the false branch.

Description of arguments of command `\IfDriverCounter`**#1** `{\langle driver counter name \rangle}`

Contains the name of the possible driver counter

#2 `{\langle True branch \rangle}`

This code is expanded if the counter is a driver, otherwise it is ignored

#3 `{\langle True branch \rangle}`This code is expanded if the counter is **not** a driver, otherwise it is ignored

```
\IsDriverCounter{section}{Yes, section is a driver counter}{No, section counter 2
  {does not have driver properties}}
```

Yes, section is a driver counter

`\IsSuspendedCounter{\langle counter name \rangle}{\langle true branch \rangle}{\langle false branch \rangle}`

See Suspending and Resuming on this topic.

This command checks, whether a counter is suspended, i.e. not updated at all and expands the corresponding branches.



#1 `{\langle counter name \rangle}`

Contains the name of counter presumed to be suspended

#2 `{\langle True branch \rangle}`

This code is expanded if the counter is suspended, otherwise it is ignored

#3 `{\langle True branch \rangle}`This code is expanded if the counter is **not** suspended, otherwise it is ignored**3.3.1 Commands checking whether a name refers to a counter**

 **xassoccnt** provides three commands that are quite similar – all check whether `{\langle name \rangle}` is an already defined $\text{\LaTeX} 2_{\epsilon}$ counter (name), in good tradition with the  **xparse** - syntax:

- `\IfIsDocumentCounterTF[\langle \rangle]{\langle name \rangle}{\langle true branch \rangle}{\langle false branch \rangle}`

This macro performs the full branching

- `\IfIsDocumentCounterT[⟨⟩]{⟨name⟩}{⟨long⟩ true branch}`

This command executes only if the name is a counter.

`\IfIsDocumentCounterF[⟨⟩]{⟨name⟩}{⟨true branch⟩}`

This command executes only if the name is not a counter.

The optional argument is not used as of version 0.2 for none of those three commands.

3.4 Information macros about the counter reset list

Sometimes it might be necessary or convenient to know how many counters are on a reset list of some other counters, i.e. added by `\newcounter{counter}[resetting counter]` or `\NewDocumentCounter`^{→ P.18}. There are some macros that provide this information:

`\countersresetlistcount{⟨counter name⟩}`

This macro determines the number of counters being in the reset list of the counter specified as mandatory argument.

Please note: This command isn't expandable. The number is stored internally to another macro, which can be accessed with `\getresetlistcount`, which returns a pure integer number.

`\getresetlistcount`

This macro returns the number of counters being in the reset list of the counter specified as mandatory argument. It needs a previous call of `\countersresetlistcount` first!

If the counter has no other counters in its reset list, the value of 0 is returned.

`\IfInResetListTF[⟨⟩]{⟨resetting counter⟩}{⟨reset counter⟩}{⟨true branch⟩}{⟨false branch⟩}`

This command sequence tests whether the counter `⟨reset counter⟩` is in the reset list of `⟨resetting counter⟩` and expands the relevant branch then. See the short-circuit commands `\IfInResetListT` and `\IfInResetListF` as well.

`\IfInResetListT[⟨⟩]{⟨resetting counter⟩}{⟨reset counter⟩}{⟨true branch⟩}`

This command sequence tests whether the counter `⟨reset counter⟩` is in the reset list of `⟨resetting counter⟩` and expands to the true branch. See the related commands `\IfInResetListTF` and `\IfInResetListF` as well.

`\IfInResetListF[⟨⟩]{⟨resetting counter⟩}{⟨reset counter⟩}{⟨false branch⟩}`

This command sequence tests whether the counter `⟨reset counter⟩` is not in the reset list of `⟨resetting counter⟩` and expands to the false branch. See the related commands `\IfInResetListTF` and `\IfInResetListT` as well.

3.5 Information on counters

On occasions it might be important to have some information which counter has been changed last. Since there are four commands manipulating counter values, there are four corresponding routines for this:

`\LastAddedToCounter`

This command has no arguments and expands to the name of the counter which was used last in `\addtocounter`. There is no further typesetting done with the countername.

```
\newcounter{SomeCounter}
```

```
\addtocounter{SomeCounter}{10}
```

```
The last counter something added to was \LastAddedToCounter.
```

```
The last counter something added to was totalpages.
```



Please note that `\LastAddedToCounter` might fail!

`\LastSteppedCounter`

This command has no arguments and expands to the name of the counter which was stepped last using `\stepcounter`. There is no further typesetting done with the countername.

```
\stepcounter{SomeCounter}
```

```
The last counter being stepped was \LastSteppedCounter.
```

```
The last counter being stepped was SomeCounter.
```

`\LastRefSteppedCounter`

```
\begin{equation}
```

```
  E = mc^2 \label{eq:einstein}
```

```
\end{equation}
```

```
% \stepcounter{SomeCounter}
```

```
The last counter being refstepped was \LastRefSteppedCounter.
```

$$E = mc^2 \tag{1}$$

```
The last counter being refstepped was equation.
```

`\LastSetCounter`

This command has no arguments and expands to the name of the counter which was set last using `\setcounter`. There is no further typesetting done with the countername.

```
\setcounter{SomeCounter}{21}%
```

The last counter being set was `\LastSetCounter`.

The last counter being set was SomeCounter.

`\LastCounterValue`

This command has no arguments and expands to the value of the very last change of a counter, i.e. using `\setcounter` etc.

```
\setcounter{SomeCounter}{100}%
```

The last counter being set was `\LastSetCounter` and it had the value 2
`\LastCounterValue{}` then, where as `\stepcounter{equation}` will yield 2
`\fbox{\LastSteppedCounter}` and `\LastCounterValue!`

The last counter being set was SomeCounter and it had the value 100 then, where as will yield `equation` and 2!

The usage of `\LastSetCounter` is best together with one of the other `\Last...` macros.



All of the `\Last...` macros are expandable, i.e. it is possible to store the value to an macro defined with `\edef`

```
\setcounter{SomeCounter}{50}%
```

```
\edef\lastcounterset{\LastSetCounter}
```

```
\edef\lastcountervalue{\LastCounterValue}
```

```
\setcounter{equation}{81}%
```

The last counter being set was `\fbox{\LastSetCounter}` and it had the value 2
`\LastCounterValue{}` then, but we changed `\lastcounterset{}` earlier and it had 2
the value `\lastcountervalue{}` then.

The last counter being set was `equation` and it had the value 81 then, but we changed SomeCounter earlier and it had the value 50 then.

! Please note, that all of these commands are only working in the current run of compilation, i.e. after there has been some operation on the counters. They can't be used for information on the last changed counter in a previous run.

4 Suspending and resuming (associated) counters

Rather than removing an associated counter from the list, it is possible to suspend the automatic stepping for a while and then resume it (or completely drop it), for example, if the value of a counter should not be stepped within a specific chapter etc.

`\SuspendCounters` [*<options>*] {*<counters list>*}

Description of arguments of command `\SuspendCounters`

#1 [*<options>*]

Not used so far, reserved for later usages

#2 {*<counters list>*}

Contains the name of counters to be suspended, separated by commas (CSV - list)

`\ResumeSuspendedCounters` [*<options>*] {*<counters list>*}

This command revokes the suspension of the counters in the {*<counters>*} list.

! If a driver counter is suspended, all counters associated to it are suspended too!


```
\textbf{This example shows 4 equations, but only two of them are counted}
```

```
\begin{equation}
```

```
E_{0} = mc^2
```

```
\end{equation}
```

Now suspend the totalequations:

```
\SuspendCounters{totalequations}
```

```
\begin{equation}
```

```
E^2 = \left(\text{pc}\right)^2 + E_{2\_0}^2
```

```
\end{equation}
```

```
\begin{equation}
```

```
m(v) = \frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}
```

```
\end{equation}
```

And resume it: \ResumeSuspendedCounters{totalequations}

```
\begin{equation}
```

```
E = h \nu
```

```
\end{equation}
```

There are \number\totvalue{totalequations}~equations in here!

```
%\end{document}
```

This example shows 4 equations, but only two of them are counted

$$E_0 = mc^2 \tag{1}$$

Now suspend the totalequations:

$$E^2 = (pc)^2 + E_0^2 \tag{2}$$

$$m(v) = \frac{m_0}{\sqrt{1 - \frac{v^2}{c^2}}} \tag{3}$$

And resume it:

$$E = h\nu \tag{4}$$

There are 2 equations in here!

5 Additions to standard commands

`\NewDocumentCounter` [*options*] {*counter*} [*resetting counter*]

This command is a new interface to `\newcounter` and behaves effectively the same.

`initial`=*integer value* (initially 0)

This is used for the start value of the new counter.

`\DeclareDocumentCounter` [*options*] {*counter*} [*resetting counter*]

This command is the preamble-only version of `\NewDocumentCounter`.

`\SetDocumentCounter` [*options*] {*counter*} {*counter value*}

This command behaves like the standard macro `\setcounter`, but has an additional optional 1st argument.

Description of arguments of command `\SetDocumentCounter`

#1 [*options*]:

`associatedtoo`=*true/false*

If enabled (*true*), `\SetDocumentCounter` will use the counter value for all counters associated to this driver counter as well. Initially, this option is set to *false*.

`onlycounters`=*comma separated list of counters*

If this key is used, only those associated counters are set as well that are given in the comma separated list.

Names, that are either not referring to counters at all or to counters that are not associated to the given driver counter will be ignored silently.

#2 {*counter*} Holds the name of the (driver) counter to be set.

#3 {*counter value*} Holds the value to be set

Some notes on `\SetDocumentCounter`

- The option keys `associatedtoo` and `onlycounters` are mutually exclusive!
- The counter to be set can be either a driver counter or an otherwise associated counter.

`\CopyDocumentCounters` [*options*] {*source counter*} {*target counter*}

This document copies the counter value from the source counter in argument 2 to the target counter in argument 3.

Description of arguments of command `\CopyDocumentCounters`

- #1 [*options*]: As of version 0.2, this option is not used
- #2 {*source counter*} Holds the name of the source counter.
- #3 {*target counter*} Holds the name of the target counter.

`\SwapDocumentCounters` [*options*] {*counter 1*} {*counter 2*}

This macro swaps the values of the counters given in arguments 2 and 3

Description of arguments of command `\SwapDocumentCounters`

- #1 [*options*]: As of version 0.2, this option is not used
- #2 {*counter 1*} Holds the name of the first counter.
- #3 {*counter 2*} Holds the name of the second counter.

`\SyncCounters` [*options*] {*driver counter*}

This document synchronizes the driver counter value to the associated values. It has the some options as `\SetDocumentCounter`^{→P.18}. If the given counter is no driver counter, nothing is done.

Description of arguments of command `\SyncCounters`

- #1 [*options*]: see `\SetDocumentCounter`^{→P.18}
- #2 {*source counter*} Holds the name of the source counter.

6 To - Do list

- Better counter definition/copy counter routines → another package perhaps
- Check the existence of driver and associated counters
- More examples
- Provide a better internal list organization
- Some macro names might be non-intuitive
- Improve documentation


If you

- find bugs
- errors in the documentation
- have suggestions
- have feature requests

don't hesitate and contact me using my mail address: christian.hupfer@yahoo.de .


7 Acknowledgments

I would like to express my gratitudes to the developpers of fine \LaTeX packages and of course to the users at tex.stackexchange.com, especially to

- Paulo Roberto Massa Cereda
- Enrico Gregorio
- Joseph Wright
- David Carlisle
- Werner
- Gonzalo Medina
- Cosmo Huber (for providing the bug report with the  `calc` package.)

for their invaluable help on many questions on macros.



A special gratitude goes to Prof. Dr. Dr. Thomas Sturm for providing the wonderful  `tcolorbox` package which was used to write this documentation.

8 Version history

- Version 0.2

Improved `\stepcounter` to remove some incompatibilities with the [perpage](#). This is only partially managed so far.

- Version: 0.1

A major bug fixed due to some error in usage together with [calc](#) when the driven counters are not stepped any longer. The culprit was in [assoccnt](#) that the counter reset list was not really disabled.

Thanks to this question <http://tex.stackexchange.com/questions/269731/calc-breaks-assoccnt> this bug was detected.

This however lead to some internal inconsistencies and it was decided to rewrite [assoccnt](#) with [expl3](#) and the features of the new \LaTeX 3 - Syntax.

Note: The `\DeclareAssociatedCounters` command have to be used in the preamble of the document. It's missing here for the sake of a compact example.

A Example: Total number of sections

In this example, all sections of this document are counted, i.e. the current one as well as all following ones.

```
This document has \total{totalsections} section(s)%
```

```
This document has 10 section(s)
```

B Example: Total number of subsections with suspension

In this example, the subsections of this document are counted but later on, the associatedcounter is removed from the list, so it is frozen.

```
\subsection{First dummy subsection}
SubSection counter: \thesubsection~-- \number\totvalue{totalsubsections}
\subsection{Second dummy subsection}
SubSection counter: \thesubsection~-- \number\totvalue{totalsubsections}

\RemoveAssociatedCounter{subsection}{totalsubsections}%
\subsection{Third dummy subsection after removing the associated counter}

SubSection counter: \thesubsection~-- \number\totvalue{totalsubsections}
```

B.1 First dummy subsection

SubSection counter: B.1 – 10

B.2 Second dummy subsection

SubSection counter: B.2 – 10

B.3 Third dummy subsection after removing the associated counter

SubSection counter: B.3 – 10

B.4 Suspension of a non-associated counter

This example will show the suspension of a non-associated counter

```
\setcounter{equation}{0}%
\SuspendCounters{equation}%
\begin{equation}
E_{0} = mc^2
\end{equation}

\begin{equation}
E^2 = \left(\right)^2 + E_{0}^2
\end{equation}

\begin{equation}
m(v) = \frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}
\end{equation}
```

There are \number\value{equation}~equations in here!

$$E_0 = mc^2 \tag{0}$$

$$E^2 = (pc)^2 + E_0^2 \tag{0}$$

$$m(v) = \frac{m_0}{\sqrt{1 - \frac{v^2}{c^2}}} \tag{0}$$

There are 0 equations in here!

Index

\AddAssociatedCounters, 6
\AddDriverCounter, 9
associatedtoo key, 18
autodefine key, 5
autodefinecounters key, 4

\ClearAssociatedCounters, 7
\ClearDriverCounters, 9
\CopyDocumentCounters, 19
\countersresetlistcount, 13

\DeclareAssociatedCounters, 5
\DeclareDocumentCounter, 18

\GetDriverCounter, 10
\getresetlistcount, 13

\IfInResetListF, 13
\IfInResetListT, 13
\IfInResetListTF, 13
\IfIsDocumentCounterF, 13
\IfIsDocumentCounterT, 13
\IfIsDocumentCounterTF, 12
initial key, 18
\IsAssociatedCounter, 11
\IsAssociatedToCounter, 9
\IsDriverCounter, 11
\IsSuspendedCounter, 12

Keys
 associatedtoo, 18
 autodefine, 5
 autodefinecounters, 4
 initial, 18
 onlycounters, 18
 sloppy, 5

\LastAddedToCounter, 14
\LastCounterValue, 15

\LastRefSteppedCounter, 14
\LastSetCounter, 15
\LastSteppedCounter, 14

\NewDocumentCounter, 18

onlycounters key, 18

Package
 assoccnt, 3, 22
 calc, 4, 21, 22
 expl3, 22
 l3keys2e, 4
 perpage, 4, 22
 tcolorbox, 21
 totcount, 3
 xassoccnt, 3, 4, 12
 xcolor, 4
 xparse, 4, 12

\RemoveAssociatedCounter, 7
\RemoveAssociatedCounters, 7
\RemoveDriverCounter, 9
\ResumeSuspendedCounters, 16

\SetDocumentCounter, 18
sloppy key, 5
\SuspendCounters, 16
\SwapDocumentCounters, 19
\SyncCounters, 19