TYPO3 Core - Bug #23826

stdWrap "insertData" does not work anymore if using "override" additionally

2010-10-22 12:54 - Susanne Moog

Status:	Closed	**Start date:**	2010-10-22
Priority:	Should have	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
TYPO3 Version:	4.5	**Complexity:**	
PHP Version:	5.3	**Is Regression:**	
Tags:		**Sprint Focus:**	

Description

Try the following TS:

```
page.100 = TEXT
page.100 {
value = polly wants {field:uid} cracker(s)
override.cObject = TEXT
override.cObject {
value = smells like teen spirit
if.value = 11
if.equals = 123
}
insertData = 1
}
```

Expected output would be something like

polly wants 69 cracker(s)

Output is:
polly wants {field:uid} cracker(s)

Now try the following:

```
page.100 = TEXT
page.100 {
value = polly wants {field:uid} cracker(s)
insertData = 1
}
```

The output is now correct. So as soon as the override is used it does not work anymore (I have currently not tested whether outher properties produce the same fault)

trunk rev 9195

(issue imported from #M16110)

Related issues:

Related to TYPO3 Core - Bug #23913: Regression : Typoscript "required" did no...	**Closed**	**2010-11-02**
Related to TYPO3 Core - Bug #24213: TypoScript: insertData change (#23826) pr...	**Closed**	**2010-11-26**
Has duplicate TYPO3 Core - Bug #24056: innerWrap2 only working if innerWrap e...	**Closed**	**2010-11-15**

History

#1 - 2010-10-22 16:04 - Jo Hasenau

Actually it works with override only, but stops working as soon as an "if" condition is involved, that returns false

So this works:

```
10 = TEXT
```

```
10 {
value = Text with {field:title}
override.cObject = TEXT
override.cObject {
value = Another text with {field:title}
if.value = 11
if.equals = 11
}
insertData = 1
}
```

while this won't:

```
10 = TEXT
10 {
value = Text with {field:title}
override.cObject = TEXT
override.cObject {
value = Another text with {field:title}
if.value = 11
if.equals = 123
}
insertData = 1
}
```

#2 - 2010-10-22 16:25 - Jo Hasenau

I tracked it down to a missing check for recursive or nested stdWrap calls.

So if there is at least one conditional function (if, required, fieldRequired) returning false, the whole rendering process stops after this function and nothing will be executed from that point on.

The attached patch Introduces a variable $this->stdWrapRecursionLevel.
It solves the problem, since the check for $this->stopRendering[$this->stdWrapRecursionLevel] is done within the current stdWrap function call only and doesn't influence the rest of the rendering process anymore.

#3 - 2010-10-25 17:42 - Ernesto Baschny

committed to trunk rev 9195 by Susanne Moog

#4 - 2011-03-29 17:47 - Susanne Moog

- Target version deleted (4.5.0)

Files

tslib_content.diff	2.21 KB	2010-10-22	Administrator Admin