Auto Multiple Choice - Feature # 338: Help us avoid bad question identifiers

Status:	Closed	**Priority:**	Normal
Author:	Akim Demaille	**Category:**	LaTeX
Created:	01/14/2015	**Assignee:**	Alexis Bienvenüe
Updated:	04/29/2015	**Due date:**	
Description:	In an exam of mine some students have very weird scores for a question: "-12", "-1-1" etc. Besides, it's about a question that does not exist. It turns out I've been using question identifiers such as "foo.1", "foo.2". Which is documented as "don't do that".		
	I _knew_ about this a year ago, but forgot about it, and fell.		
	This is really annoying, I wish AMC would use another convention to do that, something less like to be in conflict with what a user would do (for instance it could append '.amc' or prepend 'amc.' to "scope" the generated identifiers). But if it does not, at least it should warn the user when an invalid id is used.		
	Thanks!		

History

01/14/2015 12:01 pm - Alexis Bienvenüe

These are not _invalid_ id's. @foo.1@ and @foo.2@ are concatenated to build a @foo@ string. For example, this is used to build the student's id from the questions written by @AMCcode@ to get the digits.

So if you use for example @foo.1@ and @foo.2@, this is quite difficult to guess if this is an intended or not usage of this "feature".

To scope the identifiers, perhaps you would prefer the @scope:id@ naming scheme, as this allows to request a per-scope sum of scores column in the OpenDocument export.

01/14/2015 12:07 pm - Akim Demaille

Hi Alexis,

Well, there are not "invalid", they are "reserved". And I do scope my question identifiers, I had:

```
<pre>
   29: \begin{question}{urbi.naive}
   99: \begin{question}{urbi.assoc.1}
  118: \begin{question}{urbi.assoc.1}
  141: \begin{question}{urbi.disamb}
  154: \begin{question}{urbi.ll1}
  167: \begin{question}{urbi.ll2}
  181: \begin{question}{urbi.ll.impl}
  195: \begin{question}{urbi.yacc}
  209: \begin{question}{urbi.sr}
</pre>
```

But I have been wrecked because of urbi.assoc.1 and urbi.assoc.2 that are no longer understood as two questions.

I do understand the usefulness for AMC to use such IDs _internally_, yet when {question} or {questionmulti} is used, it's certainly not valid. The user API should warn about implementation details.

Granted, using @:@ instead of @.@ will help in the future. Except that @.@ is fairly common as a separator of identifiers, so it is natural to first think about it. Any piece of software should be easy to use, and warn users in dangerous situations rather than expecting that the user just read again the documentation cover to cover and remembers every detail.

01/14/2015 12:20 pm - Alexis Bienvenüe

- Target version set to 1.3.0

> AMC to use such IDs internally

In some situations (maybe to get some other codes), the end-user can also use this feature.

Maybe

* AMC should use @id[1]@, @id[2]@, etc. instead of @id.1@, @id.2@ to concatenate strings
* AMC should give the user the choice between @.@ and @:@ for scoping

01/14/2015 12:30 pm - Akim Demaille

Alexis Bienvenüe wrote:

> > AMC to use such IDs internally

> In some situations (maybe to get some other codes), the end-user can also use this feature.

Yes, but I expect that in that situation it is not {question} which is used. That some internal macro accept such identifier is fair. I'll only referring to accidental uses in {question} and {questionmulti} for instance.

>

> Maybe

> * AMC should use @id[1]@, @id[2]@, etc. instead of @id.1@, @id.2@ to concatenate strings

That would be great, especially if amc warns the user *against* using @[@ and @]@.

> * AMC should give the user the choice between @.@ and @:@ for scoping

That would be great too. Or just support both: @[.:]@.

01/14/2015 05:07 pm - Alexis Bienvenüe

> Yes, but I expect that in that situation it is not {question} which is used.

If @\AMCcode@ is not exactly what the user needs, he can use multiple @{question}@'s to build a multiple-characters code input.

04/29/2015 10:59 am - Alexis Bienvenüe

- % Done changed from 0 to 100

- Status changed from New to Closed

The two changes mentionned in note#3 are in AMC since revision commit:d24cfb078a8b.