
Subject: Re: why no virtual destructor for RegExp?

Posted by [navi](#) on Fri, 25 Jan 2013 16:37:28 GMT

[View Forum Message](#) <> [Reply to Message](#)

Quote:IMHO the pattern parameter should not be there, it is kind of confusing and against the whole idea that RegExp object represents the regexp pattern. If you want to have a chance to specify the replace pattern a separate global function might be better for this... It could use the RegExp object internally, so it's not much work for you

You are right. I think its best to get rid of pattern from arguments. Actually I was trying to match preg_replace style. which is not particularly suited for regexp class.

Quote:Why not simply

```
int ReplaceGlobal(String& t, const String r, offset=0, bool backref=false);
```

well you see, the functions Match() and GlobalMatch() has no option for offset. in order for offset to work I had to device my own function using Execute(). however then I realize if written using GlobalMatch() even-though no option for offset, but it is much more solid and more compatible then to my own version with offset. now I wanted to have the best of both hence two different function and not option=0 as default value. I believe we can still mimic the default value effect if we write the following:

```
int ReplaceGlobal(String& t, const String r, bool backref=false);  
int ReplaceGlobal(String& t, const String r, offset, bool backref=false);
```

essentially if offset is not given then the function using GlobalMatch() will be called. if offset given, then the version using Execute() will be called instead. or perhaps drop the whole idea of starting offset, like the author of Match() and GlobalMatch()? I cant get my head around function GlobalMatch(). if I could understand it fully, then I can rewrite the exact thing using Execute() alone and we then could have one function for like you pointed out.

Quote:PS: I forgot... About the returning, I think it is okay to modify the referenced string. If necessary it should be simple to add a another method with separate string for output, that would not modify the input.

I agree, I have no problem doing either way. the only issue being if we wish to return the output text then, we will have 2 return values, 1: the replace count, 2: the output text. one of which have to go, or go in to function argument as reference variable.

Thanks for all your help & pointers.

Regards

Navi
