

Re: Serious Perl Regular Expression deficiency?

Source: <http://coding.derkeiler.com/Archive/Perl/comp.lang.perl.misc/2005-12/msg01896.html>

- *From:* castillo.bryan@xxxxxxxxx
 - *Date:* 23 Dec 2005 20:13:08 -0800
-

robic0 wrote:

- > I don't see a solution to this problem that
- > regular expressions can't exclude a string when
- > processing. It can exclude individual characters
- > fine. I started doing Perl 2 years ago and have
- > run into this nagging problem several times.
- >
- > After extensive read on the Perl docs on re's
- > (especially in the last 2 days) I have come to the
- > conclusion that regular expressions have a serious
- > deficiency. This is serious because the not string
- > is a fundamental basic logic idea in a search from
- > a touted master search engine or should be.
- > To a degree it works with a known subset, but it
- > won't work to the degree shown below. This is a
- > serious flaw in regular expressions!
- >
- > I hope you masters can prove me wrong! I really do.
- > If not I would hope that the Perl authors can provide
- > some insight on when this construct can be fixed,
- > aka implemented.
- >
- > Beat this code if you can (you can't). Don't look
- > at the code in this example, look instead at the
- > output.
- > Don't comment on any code syntax because thats not
- > welcome or the point.
- > Instead, refer you comments to the output ID's.
- >
- > If you know of a way Perl regex can do this
- > please reply. I'm almost %99 sure Perl regex
- > can't do this. In fact the %1 is thrown out here
- > to either verify that or prove otherwise.
- >

Its not clear what "this" is. Are you asking if perl can do a negative match on a string, pull out XML comments with a regex, or both?

If you are wondering about a negative string match, look at the perlre

Re: Serious Perl Regular Expression deficiency?

documentation, specifically negative lookahead and lookbehind assertions.

If you want to pull out the contents of XML comments you could do this.

```
sub test_xml_comment_parse {
my ($xml) = @_;
print "XML\n", '-' x 40, "\n", $xml, "\n", '-' x 40, "\n";
while ($xml =~ s/<!--(.*?)-->//ms) {
print "Comment [$1]\n"
}
print "\n", '-' x 40, "\n\n";
}

my $gabbage1 = '
<big name="asdf" date="33" >
asdf
<!-- howdy folks -->
<in2>jjjj</in2>
<!-- and still more -->
asdfb
</big>
';

my $gabbage2 = '
<big name="asdf" date="33" >
asdf
<!-- howdy folks %SYSTEM is down <who cares?> -->
<in2>jjjj</in2>
<!-- and still more -->
asdfb
</big>
';

test_xml_comment_parse($_) foreach ($gabbage1,$gabbage2);
```

output:

XML

```
<big name="asdf" date="33" >
asdf
<!-- howdy folks -->
<in2>jjjj</in2>
<!-- and still more -->
asdfb
</big>
```

Re: Serious Perl Regular Expression deficiency?

Comment [howdy folks]
Comment [and still more]

XML

```
<big name="asdf" date="33" >
asdf
<!-- howdy folks %SYSTEM is down <who cares?> -->
<in2>jjjj</in2>
<!-- and still more -->
asdfb
</big>
```

Comment [howdy folks %SYSTEM is down <who cares?>]
Comment [and still more]

There is a problem though. If you need to retrieve data from xml documents, you should generally use an XML parser instead of using your own regular expressions.

Here is 1 case where the code I posted above would pull out the text "not really a comment", that isn't really a comment.

```
<test_xml>
<value>
<![CDATA[ <!-- not really a comment --> ]]>
</value>
</test_xml>
```

-
- **Follow-Ups:**
 - ◆ **Re: Serious Perl Regular Expression deficiency?**
◇ From: robic0

- **References:**

Re: Serious Perl Regular Expression deficiency?

◆ ***Serious Perl Regular Expression deficiency?***

◇ *From:* robic0

- Prev by Date: ***Re: Serious Perl Regular Expression deficiency?***
- Next by Date: ***Re: Apache and Perl in Windows***
- Previous by thread: ***Re: Serious Perl Regular Expression deficiency?***
- Next by thread: ***Re: Serious Perl Regular Expression deficiency?***
- Index(es):
 - ◆ ***Date***
 - ◆ ***Thread***