

Forum:[Hack Forums](#)

Thread: [Get lines between lines](#)

sososerious - September 30, 2017, 12:31 pm

Kinda weird but yeah...I tried few methods but without luck,as iv been coding pretty much whole day my mind just dosnt want to work anymore lol...

So what i need is something like this
example:

```
Line1
Line2
[start]
sdfk
sdfok
sdfk
[finish]
text
and
text
```

I want to remove lines from start to finish. Any suggestions?

0xDEAD10CC - September 30, 2017, 1:15 pm

Remove them and keep them or discard them from a resultant collection? You need to be more specific, the simplest way is obviously to just iterate over each line and ignore from start to end while adding each string to a collection of some type.

Owl - September 30, 2017, 2:02 pm

I like Dead's suggestion. Something like:

```
Code var skip = false;

foreach (var line in lines)
{
    if (line == "[start]" || line == "[finish]")
        skip = !skip;
    else if (!skip)
        // proceed
}
```

You could also index the start and finish points and interest them into a standard for loop.

0xDEAD10CC - September 30, 2017, 5:01 pm

(Sep 30, 2017 02:02 PM)Orka Wrote: [\[CITE\]](#) I like Dead's suggestion. Something like:

```
Code var skip = false;

foreach (var line in lines)
{
```

```

if (line == "[start]" || line == "[finish]")
    skip = !skip;
else if (skip)
    continue;
else
    // proceed
}

```

You could also index the start and finish points and interest them into a standard for loop.

This is pretty much what I had in mind, but you might want to be careful about the adverse effect if having a potential duplicate starting or end tag somewhere in the data. If [start] is meant to only indicate the start of the ignore phase, your logic would trigger an unignore if a duplicate [start] tag was found when you may only want to start that unignore flag when you see the [end] tag. Any assumptions past this however, becomes a fine line bordering on the premise of "uneducated guesswork" given the limited amount of details provided by OP, so I'll only add this as a mention for now.

As pseudocode this is a clear representation though.

```

Code bool skip = false;

```

```

foreach (var line in lines)
{
    if (line.Equals("[start]", StringComparison.Ordinal))
    {
        skip = true;
    }
    else if (line.Equals("[finish]", StringComparison.Ordinal))
    {
        skip = true;
    }
    else if (!skip)
    {
        // relevant data
    }
}

```

sososerious - October 1, 2017, 12:11 am

(Sep 30, 2017 05:01 PM)0xDEAD10CC Wrote: [CITE](#) This is pretty much what I had in mind, but you might want to be careful about the adverse effect if having a potential duplicate starting or end tag somewhere in the data. If [start] is meant to only indicate the start of the ignore phase, your logic would trigger an unignore if a duplicate [start] tag was found when you may only want to start that unignore flag when you see the [end] tag. Any assumptions past this however, becomes a fine line bordering on the premise of "uneducated guesswork" given the limited amount of details provided by OP, so I'll only add this as a mention for now.

As pseudocode this is a clear representation though.

```

Code bool skip = false;

```

```

foreach (var line in lines)
{
    if (line.Equals("[start]", StringComparison.Ordinal))
    {

```

```
        skip = true;
    }
    else if (line.Equals("[finish]", StringComparison.Ordinal))
    {
        skip = true;
    }
    else if (!skip)
    {
        // relevant data
    }
}
```

I have used ur code and it works. I like the way you code man.

Btw thanks to both of you guys. I had something similar in mind but i was just too tired yesterday to make it work. You guys did it with some elegance in it :) Thanks again.