

## Ruby trunk - Feature #15724

### Optionally suppress output from IRB for assignment expressions

03/22/2019 02:42 PM - onlnone (Steven Willis)

<b>Status:</b>	Open
<b>Priority:</b>	Normal
<b>Assignee:</b>	
<b>Target version:</b>	
<b>Description</b>	
<p>Most REPLs I've used don't print out the output of assignment expressions like <code>a = 1</code>. Being able to suppress this output is very useful when you have a string which is essentially a blob, an object with a large string representation, or even an array of several objects like this. This happens to me often with rails' ActiveRecord where a query might return hundreds of objects that I just want to assign to a variable and don't want to inspect yet. It's even worse with rails because queries are run lazily, just because I did <code>a = User.where(name='Sally')</code> doesn't mean I actually wanted to execute that query against the db yet. I might want to do <code>a = a.where('age &gt; 35')</code>. But irb doing the echo forces the evaluation.</p> <p>I wrote this patch to irb: <a href="https://github.com/ruby/irb/pull/12">https://github.com/ruby/irb/pull/12</a> which would suppress irb's echo for assignment expressions when a new config setting: <code>context.echo_on_assignment</code> is false. It uses the new <code>RubyVM::AbstractSyntaxTree</code> to determine whether an expression is an assignment expression, so it only works on ruby <code>&gt;= 2.6.0</code>. I'd very much welcome some feedback and any ideas on how to get it to work for earlier ruby versions.</p> <p>I had also written a first-pass proof-of-concept that used the parser gem and would work for ruby <code>&gt;= 2.0.0</code> (<a href="https://github.com/ruby/irb/pull/11">https://github.com/ruby/irb/pull/11</a>). This version obviously can't be merged because of the external dependency, but perhaps it can be treated like <code>readline</code>: use it if it's already installed, but don't depend on it?</p>	

#### History

##### #1 - 03/22/2019 03:49 PM - alanwu (Alan Wu)

You can type `a = 1; nil` to stop irb from calling `inspect` on the assignment expression result, just FYI.

##### #2 - 03/22/2019 05:05 PM - shevegen (Robert A. Heiler)

While I use a similar approach suggested by Alan Wu - and there is nothing wrong with that - I think it may still be useful to have a simpler toggle-way for silencing output by default (or, via a toggle).

I don't know if the original suggestion is good as is, but I think it is an interesting idea to support (unless there is already some config option for irb that does this, but I am not aware of it).

I don't think an external dependency can be merged, though. For enabling additional functionality, as suggested by `readline`, I guess this has to be decided by the ruby core team whether that approach is good; the better approach would obviously be to have this a possibility for everyone. Perhaps there is another way to find out whether something is an assignment expression - that could actually be helpful in general. There are a few ruby projects that have to parse input and find out whether it is valid ruby syntax or not; and then also determine what this token is doing. I have had a similar need to do so in one project that aimed to write a shell-wrapper for ruby (and then re-use it in e. g. ruby bindings to `vte`, or just as a long loop `{}` to continually use `readline` to fetch input, and act on that input).

So from this point of view I think this may be a useful suggestion.

##### #3 - 03/22/2019 09:14 PM - onlnone (Steven Willis)

alanwu (Alan Wu) wrote:

You can type `a = 1; nil` to stop irb from calling `inspect` on the assignment expression result, just FYI.

Yes, this feature is so that one doesn't have to remember to do that after each expression they wouldn't normally want echo'd.

##### #4 - 03/22/2019 09:18 PM - onlnone (Steven Willis)

shevegen (Robert A. Heiler) wrote:

I don't know if the original suggestion is good as is, but I think it is an interesting idea to support (unless there is already some config option for irb that does this, but I am not aware of it).

The only option is @context.echo which turns all echo-ing on or off.

I don't think an external dependency can be merged, though.

Yes. As mentioned, that version with the parser gem was just my first-pass proof-of-concept to show what could be done. The first link I gave (<https://github.com/ruby/irb/pull/12>) has no external dependencies, but it's only available for >= 2.6.0.

Perhaps  
there is another way to find out whether something is an assignment expression - that could actually be helpful in general.

That's exactly what I'm hoping the PR and this feature suggestion would help me find out.

**#5 - 03/27/2019 08:43 PM - onlynone (Steven Willis)**

Thanks to some feedback on the [PR](#), I've updated it to use [Ripper](#), so it doesn't rely on internals of CRuby ([RubyVM](#)) or an external library ([parser](#)).

Is there anything else I can do to get this merged?