

Ruby master - Feature #16945

Enable TCO by use of special form

06/10/2020 09:38 AM - x3qt (Yury Polyakov)

Status: Open	
Priority: Normal	
Assignee:	
Target version:	
Description I would like to propose to create some way to enable TCO in ruby by use of special form, like (recur)[[https://clojure.org/reference/special_forms#recur]]. AFAIK TCO is not enabled by default because it messes with stack traces, so by implementing some special form we will avoid changing default behavior and provide opportunity to use it at the same time. Example: <pre>def fact(n, acc = 1) return acc if n <= 1 recur(n - 1, n * acc) # recursive call with TCO enabled end</pre>	
Related issues: Is duplicate of Ruby master - Feature #12543: explicit tail call syntax: foo(...)	
	Assigned

History

#1 - 06/10/2020 09:48 AM - x3qt (Yury Polyakov)

- Backport deleted (2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN)
- Tracker changed from Bug to Feature

#2 - 06/10/2020 11:58 AM - nobu (Nobuyoshi Nakada)

- Is duplicate of Feature #12543: explicit tail call syntax: foo() then return added

#3 - 06/10/2020 11:58 AM - nobu (Nobuyoshi Nakada)

We don't want to introduce a new reserved word only for this purpose.

#4 - 06/11/2020 12:55 AM - shyouhei (Shyouhei Urabe)

Tell us how is it better than [#12543](#), which is receiving negative feedbacks?

#5 - 06/11/2020 01:15 AM - nobu (Nobuyoshi Nakada)

I thought that we had more discussions for tail-call syntax, but couldn't find others.

#6 - 06/11/2020 10:32 AM - x3qt (Yury Polyakov)

shyouhei (Shyouhei Urabe) wrote in [#note-4](#):

Tell us how is it better than [#12543](#), which is receiving negative feedbacks?

```
foo() then return
```

Syntax proposed in [#12543](#) does not look good to me, specifically because it is not a single word and using already existing constructions, so can be confusing because it is a special meaning.