Ruby trunk - Feature #12543

explicit tail call syntax: foo() then return

07/02/2016 05:24 PM - mame (Yusuke Endoh)

Status:	Closed
Priority:	Normal
Assignee:	matz (Yukihiro Matsumoto)
Target version:	

Description

How about introducing a new syntax for tail call?

```
def foo()
  foo()
end
foo() #=> stack level too deep


def bar()
  bar() then return
end
bar() #=> infinite loop
```

- no new keyword (cf. goto foo())
- no conflict with any existing syntax
- an experimental patch is available (attached)
- no shift/reduce nor reduce/reduce conflict in parse.y

--
Yusuke Endoh mame@ruby-lang.org

Related issues:

Related to Ruby trunk - Feature #6602: Tail call optimization: enable by defa...	**Feedback**

History

#1 - 07/02/2016 05:24 PM - mame (Yusuke Endoh)

- Subject changed from explicit tal call syntax: foo() then return to explicit tail call syntax: foo() then return

#2 - 07/02/2016 05:25 PM - mame (Yusuke Endoh)

- Related to Feature #6602: Tail call optimization: enable by default? added

#3 - 07/19/2016 06:53 AM - ko1 (Koichi Sasada)

mame-san:

Do you have use-cases?

#4 - 07/19/2016 09:10 AM - matz (Yukihiro Matsumoto)

- Status changed from Open to Closed

I am not positive. This may not work under tracing. I am for adding tail-call optimization, but Koichi do not love the idea either.

Matz.

Files

then_return.patch	9.18 KB	07/02/2016	mame (Yusuke Endoh)