Ruby master - Feature #5016

Kernel#caller with negative limit should limit result to N initial frames

07/11/2011 04:44 PM - now (Nikolai Weibull)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>ko1 (Koichi Sasada)</td>
</tr>
<tr>
<td>Target version:</td>
<td>2.0.0</td>
</tr>
</tbody>
</table>

**Description**
Calling Kernel#caller with a negative limit should limit result to N initial frames. When using Kernel#caller you often only want the first element of the result to get the information of the immediate caller. Generating the whole backtrace in this case is wasteful and time-consuming. Allowing Kernel#caller to take a negative limit to only return the first \(-N\) initial stack frames would allow for this.

**Related issues:**
- Related to Ruby master - Feature #1906: Kernel#backtrace: Objectifying Kernel...
- Related to Ruby master - Feature #3917: [proposal] called_from() which is muc...

**History**

**#1 - 07/12/2011 05:40 PM - ddebernardy (Denis de Bernardy)**
Dup of #1906 and #3917.

**#2 - 07/12/2011 06:14 PM - now (Nikolai Weibull)**
I wouldn’t say “dup”, but it’s certainly related. This interface can be used instead of adding Kernel#called_from, if we want to keep the number of methods down. Also, even if we objectify the backtrace as per #1906 that still doesn’t mean that we shouldn’t have the limit as per this feature request.

**#3 - 03/25/2012 04:57 PM - mame (Yusuke Endoh)**
- Status changed from Open to Assigned
- Assignee set to ko1 (Koichi Sasada)

**#4 - 03/25/2012 05:23 PM - ko1 (Koichi Sasada)**
Calling Kernel#caller with a negative limit should limit result to N initial frames. When using Kernel#caller you often only want the first element of the result to get the information of the immediate caller. Generating the whole backtrace in this case is wasteful and time-consuming. Allowing Kernel#caller to take a negative limit to only return the first \(-N\) initial stack frames would allow for this.

There are similar proposals.
https://bugs.ruby-lang.org/issues/3917

Maybe there are other proposals. I want to summarize and implement 2.0.
At first, I want to know other alternative proposals. Please let me know.

--
// SASADA Koichi at atdot dot net

**#5 - 06/26/2012 05:25 AM - ko1 (Koichi Sasada)**
- Status changed from Assigned to Closed

See https://bugs.ruby-lang.org/issues/3917