

Ruby - Feature #20673

Enable native SOCKS support by default

08/10/2024 08:04 PM - MatzFan (Brian Cohen)

|   |          |  |
|---|----------|--|
| Status:   | Feedback |  |
| Priority:   | Normal   |  |
| Assignee:   |          |  |
| Target version:   |          |  |
| <p><b>Description</b></p> <p>I'd like to see future Rubies have native SOCKS support enabled by default.</p> <p>Currently Rubies must be compiled with the --enable-socks=true option in order to provide access to the SOCKSSocket class, through which network connections may be made via the SOCKS protocol.</p> <p>My proposal is simply to deprecate the --enable-socks option IFO of SOCKS support being enabled by default.</p> <p>Use cases: To the best of my knowledge, the most commonly used SOCKS proxies are <a href="#">Dante</a> and <a href="#">Tor</a>. I know little of the former, but the latter is used as an anonymity tool. For example, the Tor Browser (based on Firefox) is built on the tor SOCKS proxy. Users wishing to make requests through the SOCKS protocol (through one of these proxies or otherwise) must compile their own Ruby with SOCKS enabled, or install Dante/Tor and make requests through these via another tool - e.g. a Selenium-driven browser, suitably configured. Alternatively, a lower level library like cURL can be utilized.</p> <p>Discussion: Most users do not build their own Rubies, but instead use the standard Rubies available from Ruby managers like rvm or rbenv (in both production and CI environments). Users needing to use the SOCKS protocol natively with one of these Rubies are presently reliant on the <a href="#">socksify</a> Ruby gem (&gt; 60M downloads). Declaration: I am the gem's current maintainer. The gem was originally built to provide native SOCKS support to users of these 'standard' Rubies via patches to the Net/HTTP and other Ruby core libraries. The current gem patches the private method Net::HTTP#connect in order to achieve this. This is a brittle and highly undesirable state of affairs and all code built using socksify is of course subject to breakage with each new Ruby release. It is difficult to gauge the exact nature and extent of the user need, but of the 60M Socksify downloads 20M have taken place in the last few years since I became maintainer. I consider this a significant user base.</p> <p>Many other languages provide native SOCKS support - e.g. Python has <a href="#">PySocks</a>.</p> <p>Fundamentally, is there any reason NOT to enable SOCKS support - would it break anything? I'd dearly like to retire the Socksify 'hack' in due course and see Ruby join the list of languages that offer SOCKS support by default - in all future Rubies.</p> <p>Many thanks for your consideration.</p> |          |  |

History

#1 - 08/23/2024 02:19 AM - znz (Kazuhiro NISHIYAMA)

Enabling SOCKS support depends SOCKS\_SERVER too now, I think it is unusual.  
So I agree to enable by default.

I created [a pull request](#).

But it cannot enable as before if building environment does not have socks library.

#2 - 09/05/2024 01:20 PM - nobu (Nobuyoshi Nakada)

At the developers' meeting, there was an opinion that we could drop the support for SOCKS if it works well with socksify.

#3 - 09/05/2024 02:36 PM - MatzFan (Brian Cohen)

[@nobu \(Nobuyoshi Nakada\)](#) there has perhaps been a misunderstanding. The socksify library works adequately for HTTP today, but it patches a private method in the Net::HTTP class. This makes any code built using socksify very fragile and potentially subject to breakage with each new Ruby release. It also no longer works for other protocols (e.g. FTP - [since Ruby 2.4](#)). The library is a hack and no substitute for the support for socks which Ruby can provide if the (currently non-default) --enable-socks compile option is used.

If Ruby developers feel that SOCKS support is not something the core language should provide, an alternative may be to make the relevant parts of the various network API's public (i.e. for each protocol that can be used with SOCKS). This way socksify or another library can be used to extend the language to provide SOCKS support properly, for those who require it.

#4 - 09/06/2024 03:46 AM - nobu (Nobuyoshi Nakada)

As for the current SOCKS support at least, if the necessary library and header are found socket.so uses SOCKS functions instead of standard socket functions, and the built socket.so cannot be loaded where SOCKS library is not installed.  
This means "native SOCKS support enabled by default" requires platforms to support it (by default or optionally).

I looked for how to install SOCKS library/headers using package managers, but couldn't find it in Ubuntu, MacPorts, HomeBrew and vcpkg at least.

An alternative I thought out is to make SOCKS::Socket class a separate extension library for libraries/users who want to use SOCKS.

MatzFan (Brian Cohen) wrote in [#note-3](#):

If Ruby developers feel that SOCKS support is not something the core language should provide, an alternative may be to make the relevant parts of the various network API's public (i.e. for each protocol that can be used with SOCKS). This way socksify or another library can be used to extend the language to provide SOCKS support properly, for those who require it.

It sounds a good way.  
Could you make a draft?

**#5 - 10/03/2024 08:53 AM - hsbt (Hiroshi SHIBATA)**

- Status changed from Open to Feedback