Ruby master - Feature #17752

Enable -Wundef for C extensions in repository

03/26/2021 05:04 PM - Eregon (Benoit Daloze)

Status: Closed
Priority: Normal
Assignee: Eregon (Benoit Daloze)
Target version: 3.1

Description
I would like to enable -Wundef for C extensions built/bundled with CRuby.

From https://gcc.gnu.org/onlinedocs/gcc/Warning-Options.html

-Wundef
Warn if an undefined identifier is evaluated in an #if directive. Such identifiers are replaced with zero.

I found this warning to be quite useful, notably when investigating why a given C extension did not include some code I expected, and then building those extensions on TruffleRuby.

There are a couple places not respecting this currently but they seem trivial to fix, I can do that.

For instance a confusing case is:
https://github.com/ruby/ruby/blob/9143d21b1bf216b1e847d569a588510726d8860/ext/nkf/utf8/nkf.h#L19

#if DEFAULT_NEWLINE == 0x0D0A

which without -Wundef would just exclude the code without any warning if DEFAULT_NEWLINE is not defined.

I'm not sure if we should/can enable it for C extensions in general (installed as gems), as if a C extensions uses -Werror and would have such a warning it would no longer build.

I can make a PR for this.
I'm not sure where to add -Wundef though, should it be in https://github.com/ruby/ruby/blob/9143d21b1bf216b1e847d569a588510726d8860/configure.ac#L620, or maybe in mkmf.rb?

Related issues:
Related to Ruby master - Bug #17850: `PAGE_SIZE` is no longer a constant for ...

Closed

Associated revisions
Revision 8b32de2e - 05/04/2021 12:56 PM - Eregon (Benoit Daloze)
Add -Werror=undef to default warnflags for core

- See [Feature #17752]
- For external extensions it's transformed to just warn and not error (-Wundef) like other other -Werror in warnflags.

Revision 9e2483ee - 05/04/2021 12:56 PM - Eregon (Benoit Daloze)
vm_dump.c: rename HAVE_BACKTRACE to USE_BACKTRACE

- HAVE_ macros should only be defined or undefined, not used for their value.
- See [Feature #17752]

Co-authored-by: xtkoba (Tee KOBAYASHI) xtkoba+ruby@gmail.com

Revision 229cbeba - 05/04/2021 12:56 PM - Eregon (Benoit Daloze)
Fix -Wundef warnings in coroutine/*/Context.h

- See [Feature #17752]

Co-authored-by: xtkoba (Tee KOBAYASHI) xtkoba+ruby@gmail.com

Revision 68d6bd08 - 05/04/2021 12:56 PM - Eregon (Benoit Daloze)
Fix trivial -Wundef warnings

- See [Feature #17752]

Co-authored-by: xtkoba (Tee KOBAYASHI) xtkoba+ruby@gmail.com

Revision 11d3c5da - 05/04/2021 12:56 PM - Eregon (Benoit Daloze)
Add RBIMPL_RVALUE_EMBED_LEN_MAX needed by internal/bignum.h

- It evaluated to 0 before, revealed by -Wundef
- See [Feature #17752]

Co-authored-by: xtkoba (Tee KOBAYASHI) xtkoba+ruby@gmail.com

Revision 59a92a84 - 05/04/2021 12:56 PM - Eregon (Benoit Daloze)
Fix -Wundef warnings in core extensions

- See [Feature #17752]

Revision fa7a712d - 05/04/2021 12:56 PM - Eregon (Benoit Daloze)
Fix -Wundef warnings for HAVE_RB_EXT_RACTOR_SAFE

- See [Feature #17752]

Revision 0764d323 - 05/04/2021 12:56 PM - Eregon (Benoit Daloze)
Fix -Wundef warnings for patterns #if HAVE

- See [Feature #17752]

Revision 108bc32 - 05/04/2021 12:56 PM - Eregon (Benoit Daloze)
HAVE_* macros should not be defined with value 0

- See [Feature #17752]

Revision d0998850 - 05/04/2021 12:56 PM - Eregon (Benoit Daloze)
Fix -Wundef warnings for RBIMPL_HAS_BUILTIN

- See [Feature #17752]
- Defining explicitly to 0 seems the best solution, see https://github.com/ruby/ruby/pull/4428
- For example: ./include/ruby/internal/has/builtin.h:49:33: error: "RBIMPL_HAS_BUILTIN_builtin_assume" is not defined, evaluates to 0
[-Werror=undef] 49 | # define RBIMPL_HAS_BUILTIN( builtin_assume) |
                                | __________________~
./include/ruby/internal/assume.h:75:7: note: in expansion of macro 'RBIMPL_HAS_BUILTIN' 75 | #elif RBIMPL_HAS_BUILTIN( builtin_assume) |
                                | ~~~~~~~~~

Revision 4868d4b4 - 02/19/2022 05:25 AM - nagachika (Tomoyuki Chikanaga)
merge revision(s) fa7a712d460dc904f8a836bb22b54d457d95ba8e:

Fix -Wundef warnings for HAVE_RB_EXT_RACTOR_SAFE

* See [Feature #17752]

---
ext/cgi/escape/escape.c | 2 +-  
ext/monitor/monitor.c | 2 +-  
ext/racc/cparse/cparse.c | 2 +-  
ext/zlib/zlib.c | 2 +-  
4 files changed, 4 insertions(+), 4 deletions(-)

History

#1 - 03/27/2021 03:25 AM - xtkoba (Tee KOBAYASHI)
I think it's just a matter of coding style to allow evaluating undefined identifiers, whether it is good or bad.

The example of ext/nkf/nkf-utf8/nkf.h seems intentional, just not bothering to write:

```c
#ifndef DEFAULT_NEWLINE
#define DEFAULT_NEWLINE 0
#endif
```
If this practice should be error-prone, we might have to change the style. I am neutral for now.

#2 - 03/27/2021 11:43 AM - Eregon (Benoit Daloze)
The #if UNDEFINED_IDENTIFIER seems fairly rare (56 vs 637), so it also seems more consistent to always use #ifdef if the identifier might not always be defined.

One example is:

```c
#if HAVE_RB_EXT_RACCTOR_SAFE
vs
#endif HAVE_RB_EXT_RACCTOR_SAFE
```

The second seems better to me, and is more explicit about the fact this might not be defined.
The value of the macro ultimately does not matter for all HAVE_ macros.

#3 - 03/27/2021 12:25 PM - xtkoba (Tee KOBAYASHI)
- File ruby-USE_BACKTRACE.patch added

The value of the macro ultimately does not matter for all HAVE_ macros.

There is a counterexample: HAVE_BACKTRACE in vm_dump.c, which is redefined as 0 when BROKEN_BACKTRACE is defined. I feel this to be an abuse of HAVE_ macros, and I would like to define a new macro USE_BACKTRACE indicating whether backtrace should be used or not, as in the attached patch.

#4 - 03/27/2021 12:48 PM - Eregon (Benoit Daloze)
Indeed, and that patch looks good to me.

#5 - 03/29/2021 04:49 AM - shyouhei (Shyouhei Urabe)
As far as the effect of -Wundef do not leak to 3rd party extension libraries, yes I'm in favor of it. It sounds a bit too harsh for 3rd parties.

#6 - 04/07/2021 01:11 AM - xtkoba (Tee KOBAYASHI)
- File ruby-trivial-undefined-macros.patch added
- File ruby-COROUTINE_LIMITED_ADDRESS_SPACE.patch added
- File ruby-BIGNUM_EMBED_LEN_MAX.patch added

Using -Wundef I saw that the definitions of BIGNUM_EMBED_LEN_MAX and COROUTINE_LIMITED_ADDRESS_SPACE might be incorrect. Patches are attached to fix them.

Also attached is another patch to calm down -Wundef warnings that seem to be trivial.

#7 - 04/07/2021 12:40 PM - xtkoba (Tee KOBAYASHI)
I noticed that -Wundef is explicitly disabled for GCC at include/ruby/internal/token_paste.h:40:

```c
#if RBIMPL_COMPILER_SINCE(GCC, 4, 2, 0)
    # define RBIMPL_TOKEN_PASTE(x, y) TOKEN_PASTE(x, y)
# /* GCC is one of such compiler who cannot write `_Pragma` inside of a `#if`.
# * Cannot but globally kill everything. This is of course a very bad thing.
# * If you know how to reroute this please tell us. */
# * http://gcc.godbolt.org/z/K2xr7X */
# pragma GCC diagnostic ignored "-Wundef"
#elif > warning: "symbol" is not defined, evaluates to 0 [-Wundef] */
```

The problem is that GCC is a major compiler that is widely used in CI, and so we will still be missing a lot of potential problems even if -Wundef is enabled by default.

#8 - 04/09/2021 01:32 AM - shyouhei (Shyouhei Urabe)
xtkoa (Tee KOBAYASHI) wrote in #note-7:

I noticed that -Wundef is explicitly disabled for GCC at include/ruby/internal/token_paste.h:40:
Ah yes I remember. This is because for instance TOKEN_PASTE(re, strict) issues undefined macro warning for "re", in spite of the rendered token restrict being valid.

PS. TOKEN_PASTE is a macro defined by configure.

#9 - 04/09/2021 09:04 AM - shyouhei (Shyouhei Urabe)
My attempt https://github.com/ruby/ruby/pull/4371

#10 - 04/13/2021 12:50 PM - Eregon (Benoit Daloze)
xtkoba (Tee KOBAYASHI) Your 4 patches look good to me, could you commit them?

#11 - 04/14/2021 01:00 AM - xtkoba (Tee KOBAYASHI)
- File ruby-trivial-undefind-macros-0002.patch added

Thanks to shyouhei (Shyouhei Urabe)'s work, we are now able to notice undefined identifiers evaluated in #if directives using GCC with -Wundef.

I attach an additional patch that fixes undefined identifiers from being evaluated.

Note that there must still remain a whole bunch of #if HAVE__ that I do not aware of because it is defined in my environment.

Eregon (Benoit Daloze) I am not sure these patches conform to the coding style. Especially I doubt if the macro of the name __RVALUE_EMBED_LEN_MAX is allowed. Could anyone brush up and commit them in place of me?

#12 - 04/29/2021 01:27 PM - Eregon (Benoit Daloze)
- Target version set to 3.1
- Assignee set to Eregon (Benoit Daloze)

I made a PR with xtkoba (Tee KOBAYASHI)'s patches and additional fixes, as well as enabling undef warnings:
https://github.com/ruby/ruby/pull/4428

xtkoba (Tee KOBAYASHI) Importing patches is quite time-consuming, it would be better if you can already make commits, with a PR or just on a branch next time.

#13 - 04/29/2021 01:39 PM - xtkoba (Tee KOBAYASHI)
Eregon (Benoit Daloze) Thank you very much. I have just learnt how to use GitHub. I will do it by myself next time.

#14 - 05/04/2021 12:57 PM - Eregon (Benoit Daloze)
- Status changed from Open to Closed

Applied in changeset git|8b32de2ec9b72d4c9ede19b70ec9497718fb25a6.

Add -Werror=undef to default warnflags for core

- See [Feature #17752]
- For external extensions it's transformed to just warn and not error (-Wundef) like other other -Werror in warnflags.

#15 - 05/04/2021 10:56 PM - mame (Yusuke Endoh)
- Status changed from Closed to Assigned

I think this change broke the following servers on RubyCI.

- Ubuntu armv7l eabihf
- CentOS 7.6 (1810) x86_64
- RHEL 7.1 x86_4
- FreeBSD 12.2 x64
- macOS Big Sur (ARM)

Eregon (Benoit Daloze) Could you please fix them?

#16 - 05/04/2021 10:58 PM - mame (Yusuke Endoh)
Maybe icc 2021.2 x64 is also broken.

#17 - 05/04/2021 11:41 PM - xtkoba (Tee KOBAYASHI)
I propose -Wundef warnings not be made into errors for the time being.

PR: https://github.com/ruby/ruby/pull/4455

#18 - 05/05/2021 12:57 AM - Eregon (Benoit Daloze)
I'll take a look tomorrow.
If we need a fix faster, I think it's OK to merge xtkoba (Tee KOBAYASHI)'s PR.

#19 - 05/05/2021 01:20 AM - xtkoba (Tee KOBAYASHI)
The remaining problems seem due to insufficient simulation of RBIMPL_HAS_* in include/ruby/internal/has/* for GCC <= 4.x and ICC.
My temporary patch would not be necessary if errors did not manifest themselves one after another.

#20 - 05/05/2021 03:57 AM - xtkoba (Tee KOBAYASHI)
I missed one more error, for which a new ticket is open now (#17850).

#21 - 05/05/2021 04:09 AM - mame (Yusuke Endoh)

- nobu (Nobuyoshi Nakada) fixed the error on old GCC: 5bbe2e61db8148c5a7974f640aee38be60bf368
- I think I fixed the error on icc (but not confirmed yet): e71c9ca529f1dce2c3816853c974ce786eea7d8
- nobu (Nobuyoshi Nakada) is now creating a patch for M1 macOS.

#22 - 05/05/2021 11:55 AM - Eregon (Benoit Daloze)
- Related to Bug #17850: `PAGE_SIZE` is no longer a constant for macOS added

#23 - 05/05/2021 11:55 AM - Eregon (Benoit Daloze)
https://rubyci.org/ looks good now except macOS Big Sur (ARM), thank you mame (Yusuke Endoh), nobu (Nobuyoshi Nakada), xtkoba (Tee KOBAYASHI), peterzhu2118 (Peter Zhu) for the fixes, and sorry for the breakage (the CI on GitHub passed).

#24 - 05/05/2021 12:38 PM - mame (Yusuke Endoh)
- Status changed from Assigned to Closed

Eregon (Benoit Daloze) No worries. This change revealed an actual issue #17850 anyway, great! I think other issues than #17850 are all addressed, so I'm closing this ticket.

**Files**

- ruby-USE_BACKTRACE.patch
  - 1.21 KB
  - 03/27/2021
  - xtkoba (Tee KOBAYASHI)
- ruby-BIGNUM_EMBED_LEN_MAX.patch
  - 950 Bytes
  - 04/07/2021
  - xtkoba (Tee KOBAYASHI)
- ruby-COROUTINE_LIMITED_ADDRESS_SPACE.patch
  - 711 Bytes
  - 04/07/2021
  - xtkoba (Tee KOBAYASHI)
- ruby-trivial-undefined-macros.patch
  - 4.35 KB
  - 04/07/2021
  - xtkoba (Tee KOBAYASHI)
- ruby-trivial-undefind-macros-0002.patch
  - 1.44 KB
  - 04/14/2021
  - xtkoba (Tee KOBAYASHI)