Ruby - Bug #11257

undefined symbol: rb_Digest_*_Init error when attempting to use digest/md5, digest/sha1, digest/sha2, and digest/ after building 2.2.2 form source.

06/12/2015 08:20 PM - alanjf (Alan J. F.)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	2.2.2	**Backport:**	2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN

Description

I ran into this after building ruby-2.2.2 from the source tarball, under Linux, when I was upgrading from ruby 1.x.

```
ruby -rdigest/sha1 -e '1'   -> ... 2.2.0/i686-linux/digest/sha1.so: undefined symbol: rb_Digest_SHA1_Init ...
ruby -rdigest/sha2 -e '1'   -> ... 2.2.0/i686-linux/digest/sha2.so: undefined symbol: rb_Digest_SHA256_Init ...
ruby -rdigest/md5 -e '1'    -> ... 2.2.0/i686-linux/digest/md5.so: undefined symbol: rb_Digest_MD5_Init ...
ruby -rdigest/rmd160 -e '1' -> ... 2.2.0/i686-linux/digest/rmd160.so: undefined symbol: rb_Digest_RMD160_Init ...
```

(digest/bubblebabble actually builds properly.)

I appears the base .c of those four (sha1.c, sha2.c, md5.c, rmd160.c) is not compiled to produce a .o (along with the *init.c and *ossl.c) and thus said missing .o files are not included in their respective .so producing lines. Somehow the compilation still succeeds but when atempting to use those extensions, they emit the errors above.

I'm not sure what the exact cause of this is (perhaps someone who works more closely with the code can shed some light), but here is a work around that I came up with that was able to produce functional .so files.

For example:

```
cd ${ruby_srouce_root}/ext/digest/sha1
```

```
gcc -I. -I../../../.ext/include/i686-linux -I../../.././include -I../../.././ext/digest/sha1 -I../../.././ext/digest/sha1/.. -I${openssl_prefix}/include -DRUBY_EXTCONF_H=\"extconf.h\"  -D_FILE_OFFSET_BITS=64  -fPIC -O3 -fno-fast-math -ggdb3 -Wall -Wextra -Wno-unused-parameter -Wno-parentheses -Wno-long-long -Wno-missing-field-initializers -Wunused-variable -Wpointer-arith -Wwrite-strings -Wdeclaration-after-statement -Wimplicit-function-declaration -Wdeprecated-declarations -Wno-packed-bitfield-compat  -fPIC   -o sha1.o -c sha1.c
```

```
rm -f ../../../.ext/i686-linux/digest/sha1.so ### Make sure the tainted .so is nuked.
```

```
gcc -shared -o ../../../.ext/i686-linux/digest/sha1.so sha1.o sha1init.o sha1ossl.o -L. -L../../.. -L. -fstack-protector -rdynamic -Wl,-export-dynamic -L${openssl_prefix}/lib    -Wl,-R${ruby_prefix}/lib -L${ruby_prefix}/lib -lruby -lcrypto  -lssl -lcrypto -lpthread -lrt -lgmp -ldl -lcrypt -lm   -lc
```

Then just cp resulting ../../../.ext/i686-linux/digest/sha1.so into ${ruby_prefix}/lib/ruby/2.2.0/i686-linux/digest/ adjusting for your arch and versions. Rinse and repeat for sha2, md5 and rmb160.

-ajf

History

#1 - 06/13/2015 01:08 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Feedback

Could you show mkmf.log file?

#2 - 07/23/2019 05:42 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Feedback to Closed