Notes on Valgrind

Valgrind is a test harness that includes many tools such as memcheck, which is commonly used to check for memory leaks, etc. The default tool run by Valgrind is memcheck. There are other tools available, but this will focus on memcheck.

Valgrind runs programs in a virtual machine, this means OpenSSL unit tests run under Valgrind will take longer than normal.

Requirements

1. Platform supported by Valgrind See http://valgrind.org/info/platforms.html
2. Valgrind installed on the platform See http://valgrind.org/downloads/current.html
3. OpenSSL compiled See INSTALL.md

Running Tests

Test behavior can be modified by adjusting environment variables.

`EXE_SHELL`

This variable is used to specify the shell used to execute OpenSSL test programs. The default wrapper (`util/wrap.pl`) initializes the environment to allow programs to find shared libraries. The variable can be modified to specify a different executable environment.

```
EXE_SHELL=\
"`/bin/pwd`/util/wrap.pl valgrind --error-exitcode=1 --leak-check=full -q"
```

This will start up Valgrind with the default checker (`memcheck`). The `--error-exitcode=1` option specifies that Valgrind should exit with an error code of 1 when memory leaks occur. The `--leak-check=full` option specifies extensive memory checking. The `-q` option prints only error messages. Additional Valgrind options may be added to the `EXE_SHELL` variable.

`OPENSSL_ia32cap`

This variable controls the processor-specific code on Intel processors. By default, OpenSSL will attempt to figure out the capabilities of a processor, and use it to its fullest capability. This variable can be used to control what capabilities OpenSSL uses.

As of valgrind-3.15.0 on Linux/x86_64, instructions up to AVX2 are supported. Setting the following disables instructions beyond AVX2:

`OPENSSL_ia32cap=":0"`

This variable may need to be set to something different based on the processor and Valgrind version you are running tests on. More information may be found in doc/man3/OPENSSL_ia32cap.pod.

Additional variables (such as `VERBOSE` and `TESTS`) are described in the file test/README.md.

Example command line:

```
$ make test EXE_SHELL="`/bin/pwd`/util/wrap.pl valgrind --error-exitcode=1 \
    --leak-check=full -q" OPENSSL_ia32cap=":0"
```

If an error occurs, you can then run the specific test via the `TESTS` variable with the `VERBOSE` or `VF` or `VFP` options to gather additional information.

```
$ make test VERBOSE=1 TESTS=test_test EXE_SHELL="`/bin/pwd`/util/wrap.pl \
   valgrind --error-exitcode=1 --leak-check=full -q" OPENSSL_ia32cap=":0"
```