Notes for the OpenVMS platform

- Requirement details
- About ANSI C compiler
- About ODS-5 directory names and Perl
- About MMS and DCL
- About debugging
- Checking the distribution

Requirement details

In addition to the requirements and instructions listed in INSTALL.md, this are required as well:

- At least ODS-5 disk organization for source and build. Installation can be done on any existing disk organization.

About ANSI C compiler

An ANSI C compiled is needed among other things. This means that VAX C is not and will not be supported.

We have only tested with DEC C (aka HP VMS C / VSI C) and require version 7.1 or later. Compiling with a different ANSI C compiler may require some work.

Please avoid using C RTL feature logical names `DECC$*` when building and testing OpenSSL. Most of all, they can be disruptive when running the tests, as they affect the Perl interpreter.

About ODS-5 directory names and Perl

It seems that the perl function canonpath() in the `File::Spec` module doesn't treat file specifications where the last directory name contains periods very well. Unfortunately, some versions of VMS tar will keep the periods in the OpenSSL source directory instead of converting them to underscore, thereby leaving your source in something like `[.openssl-1^.1^.0]`. This will lead to issues when configuring and building OpenSSL.

We have no replacement for Perl's canonpath(), so the best workaround for now is to rename the OpenSSL source directory, as follows (please adjust for the actual source directory name you have):

```
$ rename openssl-1^.1^.0.DIR openssl-1_1_0.DIR
```

About MMS and DCL

MMS has certain limitations when it comes to line length, and DCL has certain limitations when it comes to total command length. We do what we can to mitigate, but there is the possibility that it's not enough. Should you run into

issues, a very simple solution is to set yourself up a few logical names for the directory trees you're going to use.

About debugging

If you build for debugging, the default on VMS is that image activation starts the debugger automatically, giving you a debug prompt. Unfortunately, this disrupts all other uses, such as running test programs in the test framework.

Generally speaking, if you build for debugging, only use the programs directly for debugging. Do not try to use them from a script, such as running the test suite.

The following is not available on Alpha

As a compromise, we're turning off the flag that makes the debugger start automatically. If there is a program that you need to debug, you need to turn that flag back on first, for example:

```
$ set image /flag=call_debug [.test]evp_test.exe
```

Then just run it and you will find yourself in a debugging session. When done, we recommend that you turn that flag back off:

```
$ set image /flag=nocall_debug [.test]evp_test.exe
```

Checking the distribution

There have been reports of places where the distribution didn't quite get through, for example if you've copied the tree from a NFS-mounted Unix mount point.

The easiest way to check if everything got through as it should is to check that this file exists:

```
[.include.openssl]configuration^.h.in
```

The best way to get a correct distribution is to download the gzipped tar file from ftp://ftp.openssl.org/source/, use `GZIP -d` to uncompress it and `VMSTAR` to unpack the resulting tar file.

Gzip and VMSTAR are available here:

http://antinode.info/dec/index.html#Software

Should you need it, you can find UnZip for VMS here:

http://www.info-zip.org/UnZip.html

How the value of 'arch' is determined ————————————————-

'arch' is mentioned in INSTALL. It's value is determined like this:

```
arch = f$edit( f$getsyi( "arch_name"), "upcase")
```