VSI SSL111 Version 1.1-1I for OpenVMS Installation Guide and Release Notes

December 2020

This document contains hardware and software prerequisites,
installation instructions, post-installation tasks,
instructions for building your application, the VSI SSL111 directory
structure, and release notes for VSI SSL111 Version 1.1-1I for OpenVMS.

The information in this file applies to VSI SSL111 running on
OpenVMS Integrity servers.

VSI SSL111 Version 1.1-1I for OpenVMS is based on Open Source OpenSSL version
1.1.1i from OpenSSL.org.

Vulnerabilities CVE/CAN:

Please refer OpenSSL websites: https://www.openssl.org/news/vulnerabilities.html

---
Installation Requirements and Prerequisites

The following sections list hardware and disk space requirements, and
software prerequisites.

- Hardware Prerequisites - Disk Space Requirements

  The VSI SSL111 for OpenVMS kit requires approximately 200,000 blocks of
  working disk space to install. Once installed, the software occupies
  approximately 170,000 blocks of disk space.

- Software Prerequisites

  VSI SSL111 V1.1-1I for OpenVMS requires the following software.

- Operating System

  VSI OpenVMS Integrity server Version 8.4-1H1 or later.

- Account Quotas and System Parameters

  There are no specific requirements for account quotas and system
  parameters for installing or using VSI SSL111 for OpenVMS.

New Features in VSI SSL111 Version 1.1-1I for OpenVMS

VSI SSL111 Version 1.1-1I for OpenVMS is based on Open Source
OpenSSL Version 1.1.1i stream.

Coexistence and major changes between VSI SSL V1.4, VSI SSL1, and VSI SSL111 V1.1

The SSL product name has been changed to SSL111 to allow VSI SSL V1.4, (based on
OpenSSL 0.9.8 stream), SSL1 based on OpenSSL 1.0.2 stream), and VSI SSL111 V1.1
(based on OpenSSL 1.1.1 stream) to coexist on the same system.

VSI recommends that VSI SSL111 V1.1, VSI SSL1, and VSI SSL V1.4 products be
installed until any applications dependent on VSI SSL have been recompiled and
relinked against VSI SSL111.

Once all the dependent products or components have been successfully migrated
to VSI SSL111 V1.1, the earlier VSI SSL V1.4 and SSL1 kits can be removed.

Following is a snapshot of co-existence:

```
$ PROD SHOW PROD SSL*
------------------------------------ ----------- ---------
PRODUCT                              KIT TYPE    STATE
------------------------------------ ----------- ---------
VSI I64VMS SSL V1.4-503              Full LP     Installed
VSI I64VMS SSL1 V1.0-2UA             Full LP     Installed
VSI I64VMS SSL111 V1.1-1IA           Full LP     Installed
------------------------------------ ----------- ---------
3 items found
```

Logical names:

All the logical names associated with VSI SSL111 V1.1 are prefixed with SSL111$. The following is a comparison of system-level logical names that are defined for VSI SSL V1.4 and VSI SSL111 V1.1 (a similar comparison can be made between SSL111 and SSL1):

VSI SSL V1.4-503 Logicals	VSI SSL111 V1.1-1I Logicals
"OPENSSL" = "SSL$INCLUDE:"	"OPENSSL" = "SSL111$INCLUDE:"
"SSL$CERT" = "SSL$ROOT:[DEMOCA.CERTS]"	"SSL111$CERT" = "SSL111$ROOT:[DEMOCA.CERTS]"
"SSL$CERTS" = "SSL$ROOT:[DEMOCA.CERTS]"	"SSL111$CERTS" = "SSL111$ROOT:[DEMOCA.CERTS]"
"SSL$COM" = "SSL$ROOT:[COM]"	"SSL111$COM" = "SSL111$ROOT:[COM]"
"SSL$CONF" = "SSL$ROOT:[DEMOCA.CONF]"	"SSL111$CONF" = "SSL111$ROOT:[DEMOCA.CONF]"
"SSL$CRL" = "SSL$ROOT:[DEMOCA.CRL]"	"SSL111$CRL" = "SSL111$ROOT:[DEMOCA.CRL]"
"SSL$EXAMPLES" = "SYS$COMMON:[SYSHLP.EXAMPLES.SSL]"	"SSL111$EXAMPLES" = "SYS$COMMON:[SYSHLP.EXAMPLES.SSL111]"
"SSL$EXE" = "SSL$ROOT:[IA64_EXE]"	"SSL111$EXE" = "SSL111$ROOT:[IA64_EXE]"
"SSL$INCLUDE" = "SSL$ROOT:[INCLUDE]"	"SSL111$INCLUDE" = "SSL111$ROOT:[INCLUDE]"
"SSL$KEY" = "SSL$ROOT:[DEMOCA.CERTS]"	"SSL111$KEY" = "SSL111$ROOT:[DEMOCA.CERTS]"
"SSL$KEYS" = "SSL$ROOT:[DEMOCA.CERTS]"	"SSL111$KEYS" = "SSL111$ROOT:[DEMOCA.CERTS]"
"SSL$PRIVATE" = "SSL$ROOT:[DEMOCA.PRIVATE]"	"SSL111$PRIVATE" = "SSL111$ROOT:[DEMOCA.PRIVATE]"
"SSL$ROOT" = "SYS$SYSDEVICE:[VMS$COMMON.SSL.]"	"SSL111$ROOT" = "SYS$SYSDEVICE:[VMS$COMMON.SSL111.]"

These logical names get defined by invoking SYS$STARTUP:SSL$STARTUP.COM and SYS$STARTUP:SSL111$STARTUP.COM startup command procedures respectively.

The logical name "OPENSSL" is mainly used to identify the OpenSSL header file location for building a product against OpenSSL. When VSI SSL V1.4, VSI SSL1, and VSI SSL111 V1.1 versions co-exist, the "OPENSSL" logical name will be pointed to the version of product that was started last.

If there are any custom command procedures on your system using "SSL$..." or "SSL1$..." logical names, ensure that they are modified to use "SSL111$..." logical names when migrating from VSI SSL V1.4 or VSI SSL1 to VSI SSL111 V1.1.

Directory names:

The top level directory structure for VSI SSL111 V1.1 is SYS$SYSDEVICE:[VMS$COMMON.SSL111]. The top level directory structures for VSI SSL V1.4 and VSI SSL1 (if installed) remain as SYS$SYSDEVICE:[VMS$COMMON.SSL] and SYS$SYSDEVICE:[VMS$COMMON.SSL1], respectively.

VSI SSL111 V1.1 example programs are located in SYS$COMMON:[SYSHLP.EXAMPLES.SSL111] directory.

If there are any custom command procedures on your system referencing the "[SSL]" or "[SSL1]" directories, ensure that they are modified to use the new "[SSL111]" directory when migrating from VSI SSL V1.4 or VSI SSL1 to VSI SSL111 V1.1.

Command procedure names:

The relevant command procedure names are prefixed with "SSL111" for
the VSI SSL111 V1.1 product. For example:

SYS$STARTUP:SSL111$STARTUP.COM
SSL111$COM:SSL111$CERT_TOOL.COM

Command procedures for VSI SSL V1.4 and VSI SSL1 are prefixed with
"SSL" and "SSL1", respectively.

If there are any custom command procedures on your system invoking
"SSL$..." or "SSL1$..." command procedures, ensure that they are
modified to invoke "SSL111$..." command procedures when migrating from
VSI SSL V1.4 or VSI SSL1 to VSI SSL111 V1.1.

Library names:

Library names for VSI SSL111 V1.1 are prefixed with SSL111$ as follows:

 SYS$SHARE:SSL111$LIBSSL_SHR.EXE
 SYS$SHARE:SSL111$LIBCRYPTO_SHR.EXE
 SYS$SHARE:SSL111$LIBSSL_SHR32.EXE
 SYS$SHARE:SSL111$LIBCRYPTO_SHR32.EXE

Library names for VSI SSL V1.4 and VSI SSL1 remain unchanged:

 SYS$SHARE:SSL$LIBSSL_SHR.EXE
 SYS$SHARE:SSL$LIBCRYPTO_SHR.EXE
 SYS$SHARE:SSL$LIBSSL_SHR32.EXE
 SYS$SHARE:SSL$LIBCRYPTO_SHR32.EXE

 SYS$SHARE:SSL1$LIBSSL_SHR.EXE
 SYS$SHARE:SSL1$LIBCRYPTO_SHR.EXE
 SYS$SHARE:SSL1$LIBSSL_SHR32.EXE
 SYS$SHARE:SSL1$LIBCRYPTO_SHR32.EXE

Applications that are linked with VSI SSL V1.4 or VSI SSL1 will continue
using VSI SSL V1.4 or VSI SSL1 libraries and applications that are
linked with VSI SSL111 V1.1 product will use the new libraries shipped
with VSI SSL111 product.

The logical name "OPENSSL" is used commonly by VSI SSL111 V1.1, VSI
SSL1, and VSI SSL V1.4. Care must therefore be taken to identify that
this logical name is defined to the appropriate path (SSL111$INCLUDE:,
SSL1$INCLUDE: or SSL$INCLUDE:) before rebuilding applications.

Migrate certificate store from VSI SSL V1.4 or VSI SSL1 to VSI SSL111 V1.1:

 - The top level directory structure of VSI SSL111 V1.1 is modified to
   SYS$SYSDEVICE:[VMS$COMMON.SSL111] from SYS$SYSDEVICE:[VMS$COMMON.SSL] or
   SYS$SYSDEVICE:[VMS$COMMON.SSL1] (Which are the top level directories for
   VSI SSL 1.4 and VSI SSL1 respectively).

   In case there is a certificate store manually created in
   SYS$SYSDEVICE:[VMS$COMMON.SSL.DEMOCA...] or SYS$SYSDEVICE:[VMS$COMMON.SSL1.DEMOCA...],
   copy the certificate store to SYS$SYSDEVICE:[VMS$COMMON.SSL111.DEMOCA...].

 - In a certificate store, the certificate files will have names of the form
   "hash.0" or will have symbolic links to names of this form (where "hash" is
   the hashed certificate subject name; see the -hash option of the openssl x509
   utility).

   From VSI SSL V1.4 or VSI SSL1 to VSI SSL111 V1.1, this hash is modified
   from the MD5 to the SHA-1 algorithm. Due to this modification, validation of
   certificates will fail with SSL111 if we use the same hash names.

Manually rename the certificate file name to use the new hash.

An example of moving a certificate from VSI SSL V1.4 to VSI SSL111 V1.1 is as follows:

a) Assume we have VSI SSL V1.4 installed and had created a certificate store in SSL$ROOT:[DEMOCA.CERTS].

b) Assume we have a certificate file 438F16D6.0 in SSL$ROOT:[DEMOCA.CERTS]. The name "438F16D6" of this certificate file is the MD5 hash of the certificate subject.

```
$ @SSL$COM:SSL$UTILS
$ openssl x509 -hash -in SSL$ROOT:[DEMOCA.CERTS]438F16D6.0
 438F16D6
 -----BEGIN CERTIFICATE-----
 MIIB9zCCAWACCQC1TifkDidaxTANBgkqhkiG9w0BAQUFADBAMQswCQYDVQQGEwJV
 UzELMAkGA1UECgwCSFAxDTALBgNVBAsMBFNUU0QxFTATBgNVBAMMDENBIEF1dGhv
 cml0eTAeFw0xNTExMjYyMTI3NThaFw0yMDExMjQyMTI3NThaMEAxCzAJBgNVBAYT
 AlVTMQswCQYDVQQKDAJIUDENMAsGA1UECwwEU1RTRDEVMBMGA1UEAwwMQ0EgQXV0
 aG9yaXR5MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC3v+0ecrW2nbQ7ASwe
 6hNeCPyixt6FdqnADVTVAws7TG70JFtVPK6pbc81grwJZPbJn1oAxTGMLLiANr/Y
 XPlU73OUG+rrSiirq5fhWjVrD6M+yK9XHo6qnjMVUuwXITc8Sxr1xzDb/nOBX1+L
 qkzGIX/4hvc4ko4OZ8mhKkEauwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAJetkXxW
 YSi/crNHg+vSPiK1QA/KwLKDSNFDNazyvM9toswa9yA6U6ZBal0WCTj9efOi8Rbd
 l1AH7HEUXUTccIrjlzOVsO4safWGt/wpyHNMZGAxA25Dd8fQbf9GpAvooaSPrdJU
 u23fgeoXF3GcLYd/hog/yhpOq1w+BsA+nVi+
 -----END CERTIFICATE-----
$
```

b) Now after installing VSI SSL111 V1.1, executing the "openssl x509 -hash" command from SSL111 gives "37d8de08" which is a SHA-1 hash of the certificate subject.

```
$ @SSL111$COM:SSL111$UTILS
$ openssl x509 -hash -in SSL$ROOT:[DEMOCA.CERTS]438F16D6.0
 37d8de08
 -----BEGIN CERTIFICATE-----
 MIIB9zCCAWACCQC1TifkDidaxTANBgkqhkiG9w0BAQUFADBAMQswCQYDVQQGEwJV
 UzELMAkGA1UECgwCSFAxDTALBgNVBAsMBFNUU0QxFTATBgNVBAMMDENBIEF1dGhv
 cml0eTAeFw0xNTExMjYyMTI3NThaFw0yMDExMjQyMTI3NThaMEAxCzAJBgNVBAYT
 AlVTMQswCQYDVQQKDAJIUDENMAsGA1UECwwEU1RTRDEVMBMGA1UEAwwMQ0EgQXV0
 aG9yaXR5MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC3v+0ecrW2nbQ7ASwe
 6hNeCPyixt6FdqnADVTVAws7TG70JFtVPK6pbc81grwJZPbJn1oAxTGMLLiANr/Y
 XPlU73OUG+rrSiirq5fhWjVrD6M+yK9XHo6qnjMVUuwXITc8Sxr1xzDb/nOBX1+L
 qkzGIX/4hvc4ko4OZ8mhKkEauwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAJetkXxW
 YSi/crNHg+vSPiK1QA/KwLKDSNFDNazyvM9toswa9yA6U6ZBal0WCTj9efOi8Rbd
 l1AH7HEUXUTccIrjlzOVsO4safWGt/wpyHNMZGAxA25Dd8fQbf9GpAvooaSPrdJU
 u23fgeoXF3GcLYd/hog/yhpOq1w+BsA+nVi+
 -----END CERTIFICATE-----
$
```

c) You will have to use a certificate file name having "37d8de08" if you wish to use this certificate store with VSI SSL111 V1.1:

```
$ COPY SSL$ROOT:[DEMOCA.CERTS]438F16D6.0 -
       SSL111$ROOT:[DEMOCA.CERTS]37d8de08.0
```

OR

```
$ openssl x509 -hash -in SSL$ROOT:[DEMOCA.CERTS]438F16D6.0 .out
SSL111$ROOT:[DEMOCA.CERTS]37d8de08.0
```

(Here we are assuming that SSL111$ROOT:[DEMOCA.CERTS] is the new certificate store directory used with VSI SSL111 V1.1)

d) Repeat steps b) and c) for all certificates in the certificate store.

```
    e) Certificate verification (using either the "openssl verify" command
           or verifying the certificate using OpenSSL API's) will work with
           VSI SSL111 V1.1, only if (for the above example) the certificate name
           in the certificate store is "37d8de08.0"

    f) Once you have stopped using the VSI SSL V1.4 certificate store you
           can delete the older certificate files having MD-5 hash file names.

 - For more information, see help on

   openssl x509 -hash, -subject, -subject_hash_old, -issuer, -issuer_hash_old
   option - https://www.openssl.org/docs/man1.0.2/apps/x509.html

   openssl verify -CApath option -
   https://www.openssl.org/docs/man1.0.2/apps/verify.html


OpenSSL documentation from the Open Group
-----------------------------------------
Documentation about the OpenSSL project and The Open Group is
available at the following URL:

http://www.openssl.org

The OpenSSL documentation was written for UNIX users. When reading
UNIX-style OpenSSL documentation, note the following differences
between UNIX and OpenVMS:

- File specification format

  The OpenSSL documentation shows example file specifications in UNIX
  format. For example, the UNIX file specification
  /dka100/foo/bar/file.dat is equivalent to DKA100:[FOO.BAR]FILE.DAT on
  OpenVMS.

- Directory format

  Directories (pathnames) that begin with a period (.) on UNIX begin
  with an underscore (_) on OpenVMS. In addition, on UNIX, the tilde (~)
  is an abbreviation for SYS$LOGIN. For example, the UNIX pathname
  ~/.openssl/profile/prefs.js is equivalent to the OpenVMS directory
  [._OPENSSL.PROFILE]PREFS.JS.


Installing VSI SSL111
---------------------
Install the VSI SSL111 V1.1 for OpenVMS kit by entering the following command:

$ PRODUCT INSTALL SSL111

Whereupon you should observe output similar to the following:

Performing product kit validation of signed kits ...

The following product has been selected:
    VSI I64VMS SSL111 V1.1-1I               Layered Product

Do you want to continue? [YES]

Configuration phase starting ...

You will be asked to choose options, if any, for each selected product and for
any products that may be installed to satisfy software dependency requirements.

Configuring VSI I64VMS SSL111 V1.1-1I: SSL111 for OpenVMS I64 V1.1-1I (Based on OpenSSL 1.1.1i)

    Copyright 2019 VMS Software, Inc.

Do you want the defaults for all options? [YES]
```

```
    VSI SSL111 is not Backward Compatible!

    The VSI SSL111 Version 1.1-1I for OpenVMS is based on the 1.1.1i baselevel of
    OpenSSL. Some of the OpenSSL API, data structure, and commands are changed
    from the previous VSI SSL version 1.4 (Based on OpenSSL 0.9.8 versions).

    If your application is dependent on the VSI SSL 1.4 version of
    product, you must recompile and relink your code with the latest SSL111
    header files and shareable images in order to make use of SSL111 functions
    and features.

    VSI SSL111 and VSI SSL product libraries (shareable images) have different
    names, hence both products may co-exist on the same system.
    Applications that are linked with VSI SSL product will
    continue using VSI SSL libraries and Applications that are linked with
    VSI SSL111 product will use the new libraries shipped with VSI SSL111 product.

    The logical "OPENSSL" is used commonly by both VSI SSL111 and VSI SSL product.
    Care should be taken to identify that this logical is defined to the
    appropriate path before rebuilding the application with the correct libraries
    and header files of VSI SSL111. See the VSI SSL111 "Installation Guide and
    Release Notes" for more information on migrating applications to VSI SSL111.


    Do you want to continue? [YES]

Do you want to review the options? [NO]

Execution phase starting ...

The following product will be installed to destination:
    VSI I64VMS SSL111 V1.1-1I               DISK$I64SYS:[VMS$COMMON.]

Minimum OpenVMS IA64 software not found on system, abort installation

This kit requires a minimum OpenVMS IA64 version of V8.4-1H1.


Terminating is strongly recommended.  Do you want to terminate? [YES] no

Portion done: 0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%

The following product has been installed:
    VSI I64VMS SSL111 V1.1-1I               Layered Product

%PCSI-I-IVPEXECUTE, executing test procedure for VSI I64VMS SSL111 V1.1-1I ...
%PCSI-I-IVPSUCCESS, test procedure completed successfully

VSI I64VMS SSL111 V1.1-1I: SSL111 for OpenVMS I64 V1.1-1I (Based on OpenSSL 1.1.1i)

    There are post-installation tasks that you must complete.

    after upgrading from previous SSL111 versions

    Refer to SYS$HELP:SSL101-1I-I64.RELEASE_NOTES for more information.
$


Stopping and restarting the installation:
-----------------------------------------
Use the following procedure to stop and restart the installation:

To stop the procedure at any time, press Ctrl/Y.

Enter the DCL command PRODUCT REMOVE SSL111 to reverse any changes to the
system that occurred during the partial installation. This deletes all
files created up to that point and causes the installation procedure
to exit.
```

To restart the installation, go back to the beginning of the installation
procedure.

Post-installation Tasks

After the installation is complete, perform the steps in one of the
following sections:

- Ensuring SSL111 startup, shutdown, and logical name creation files are executed

  Add SSL111$STARTUP.COM to SYS$MANAGER:SYSTARTUP_VMS.COM to define SSL111$ logical
  names and install shareable images. If there is already a SSL$STARTUP.COM and/or
  SSL1$STARTUP.COM present in SYSTARTUP_VMS.COM you can either comment these out
  or conditionalize the command procedure as appropriate.

  For example:

```
$ if f$search("sys$startup:ssl$startup.com") .nes. ""
$ then
$    @sys$startup:ssl$startup.com
$ endif
$ if f$search("sys$startup:ssl1$startup.com") .nes. ""
$ then
$    @sys$startup:ssl1$startup.com
$ endif
$ if f$search("sys$startup:ssl111$startup.com") .nes. ""
$ then
$    @sys$startup:ssl111$startup.com
$ endif
```

  The SSL111$STARTUP.COM, SSL1$STARTUP.COM and SSL$STARTUP.COM startup command
  procedures in the above example will automatically define the SSL111$, SSL1,
  and SSL$ executive-mode logical names in the SYSTEM logical name table and will
  install into memory the SSL111, SSL1, and SSL 1.4 shareable images that
  reside in the [SYSLIB] directory.

  Ensure that the SSL111$STARTUP.COM command procedure is invoked after
  invoking SSL$STARTUP.COM or SSL1$STARTUP.COM. The command procedures define
  a common logical "OPENSSL" that points to the include (header) file directory
  used when building applications using OpenSSL. Invoking SSL111$STARTUP.COM
  last ensures that the logical is defined to correctly point to the latest
  VSI SSL111 1.1 header files.

  Also, add SSL111$SHUTDOWN.COM to SYS$MANAGER:SYSHUTDWN.COM to remove installed
  images and deassign the SSL111$ logical names at system shutdown. If there is a
  SSL$SHUTDOWN.COM and/or SSL1$SHUTDOWN.COM already present in
  SYS$MANAGER:SYSHUTDWN.COM, conditionalize the script as appropriate.

  For example:

```
$ if f$search("sys$startup:ssl$shutdown.com") .nes. ""
$ then
$    @sys$startup:ssl$shutdown.com
$ endif
$ if f$search("sys$startup:ssl1$shutdown.com") .nes. ""
$ then
$    @sys$startup:ssl1$shutdown.com
$ endif
$ if f$search("sys$startup:ssl111$shutdown.com") .nes. ""
$ then
$    @sys$startup:ssl111$shutdown.com
$ endif
```

  Please refer to "Logical names" under section "Coexistence and major changes
  between VSI SSL V1.4, VSI SSL1, and VSI SSL111 V1.1" in this document.

- Define the foreign commands that use the OpenSSL utility OPENSSL.EXE

such as openssl, ca, enc, req, and X509, by entering the following command:

```
$ @SSL111$COM:SSL111$UTILS
```

- Updated VSI SSL111 files requiring attention

Systems with custom changes to SSL or SSL1 command procedures may need to replicate those changes in the SSL111 command procedures. If so, perform the following actions where appropriate:

- Copy any manual changes done to the site-specific startup command procedures SSL$COM:SSL$SYSTARTUP.COM or SSL1$COM:SSL1$SYSTARTUP.COM to SSL111$COM:SSL111$SYSTARTUP.COM.

- If SYS$STARTUP:SSL$STARTUP.COM or SYS$STARTUP:SSL1$STARTUP.COM have any manual changes, ensure that these changes are copied to the site-specific startup command procedure SSL111$COM:SSL111$SYSTARTUP.COM. This command procedure will be invoked by SYS$STARTUP:SSL111$STARTUP.COM.

- Copy any manual changes done to the site-specific shutdown command procedures SSL$COM:SSL$SYSHUTDOWN.COM or SSL1$COM:SSL1$SYSHUTDOWN.COM to SSL111$COM:SSL111$SYSHUTDOWN.COM.

- If SYS$STARTUP:SSL$SHUTDOWN.COM or SYS$STARTUP:SSL1$SHUTDOWN.COM have any manual changes, ensure that these changes are copied to the site-specific shutdown command procedure SSL111$COM:SSL111$SYSHUTDOWN.COM. This command procedure will be invoked by SYS$STARTUP:SSL111$SHUTDOWN.COM.

- Copy any manual changes done to the OpenSSL configuration files SSL$ROOT:[000000]OPENSSL.CNF or SSL1$ROOT:[000000]OPENSSL.CNF to SSL111$ROOT:[000000]OPENSSL.CNF.

- Copy any manual changes done to the OpenSSL configuration files SSL$ROOT:[000000]OPENSSL-VMS.CNF or SSL1$ROOT:[000000]OPENSSL-VMS.CNF to SSL111$ROOT:[000000]OPENSSL-VMS.CNF.

- Migrate any SSL certificates store content to VSI SSL111 V1.1 by following the steps highlighted under "Migrate certificate store from VSI SSL V1.4 or VSI SSL1 to VSI SSL111 V1.1" to SSL111 V1.1".

- Migrate any applications built with VSI SSL V1.4 and/or VSI SSL1 to VSI SSL111 V1.1 by rebuilding and relinking the application with the VSI SSL111 V1.1 header files and libraries.

- Migrate any command procedures using VSI SSL V1.4 and/or VSI SSL1 directories, command procedures or logicals to point to VSI SSL111 V1.1 directories, command procedures or logicals. See "Co-existance and major changes between VSI SSL V1.4 and VSI SSL V1.0" section of this document for more information.

- Optionally run the Installation Verification Procedure (IVP) test by entering the following command:

```
$ @SYS$TEST:SSL111$IVP.COM
```

- Optionally start the Certificate Tool by entering the following command:

```
$ @SSL111$COM:SSL111$CERT_TOOL
```

This menu-driven tool allows you to create and view certificates and certificate requests and to sign certificate requests.

VSI SSL111 directory structure

The VSI SSL111 directory structure is as follows:

Root directory: SYS$SYSDEVICE:[VMS$COMMON]

[SSL111] - Top-level directory created by default in SYS$SYSDEVICE:[VMS$COMMON].

One of the following two directories:

```
[SSL111.IA64_EXE]        - Contains images for the Integrity server platform.
[SSL111.COM]             - Contains command procedures.
[SSL111.DEMOCA]          - Contains demos for SSL's CA features
[SSL111.DEMOCA.CERTS]    - Contains certificates and keys.
[SSL111.DEMOCA.CONF]     - Contains configuration files.
[SSL111.DEMOCA.CRL]      - Contains revoked certificates and CRLs.
[SSL111.DEMOCA.PRIVATE]  - Contains private keys and random data.
[SSL111.DOC]             - OpenSSL Group-provided documentation and information.
[SSL111.INCLUDE]         - Contains C header (.H) files.
[SSL111.LIB]             - Contains static libraries (.OLB) files.
[SSL111.TEST]            - Contains files used during the Installation Verification
                           Procedure (IVP).
[SYS$STARTUP]            - Contains startup and shutdown templates and files.
[SYSHLP]                 - Contains release notes.
[SYSHLP.EXAMPLES.SSL111] - Contains SSL crypto and secure session examples.
[SYSLIB]                 - Contains SSL shareable image files.
[SYSTEST]                - Contains SSL111$IVP.COM test files.
```

Note that the VSI SSL111 example programs are located in
SYS$COMMON:[SYSHLP.EXAMPLES.SSL111]. The logical name SSL111$EXAMPLES
points to this directory.

Building a VSI SSL111 application

VSI SSL111 for OpenVMS provides shareable images that contain 64-bit APIs
and shareable images that contain 32-bit APIs. You can choose which API
you wish to use when you compile your application.

The file names for these shareable images are as follows:

```
SYS$SHARE:SSL111$LIBSSL_SHR.EXE      - 64-bit SSL APIs
SYS$SHARE:SSL111$LIBCRYPTO_SHR.EXE   - 64-bit Crypto APIs
SYS$SHARE:SSL111$LIBSSL_SHR32.EXE    - 32-bit SSL APIs
SYS$SHARE:SSL111$LIBCRYPTO_SHR32.EXE - 32-bit Crypto APIs
```

When you compile your application using VSI C, use the /POINTER_SIZE=64
qualifier to take advantage of the 64-bit APIs. The default value for
the /POINTER_SIZE qualifier is 32.

Linking your application is the same for either 64-bit or 32-bit APIs.
However, the options file used contains either the 64-bit or 32-bit
references to the appropriate shareable image.

Building an application using 64-Bit APIs

To build (compile and link) an example program using the 64-bit APIs,
enter the following commands:

```
$ CC/POINTER_SIZE=64/PREFIX=ALL SAMPLE.C
$ LINK/MAP SAMPLE,LINKER_OPT/OPTIONS
```

In these commands, LINKER_OPT.OPT is a simple text file that contains
the following lines:

```
SYS$SHARE:SSL111$LIBSSL_SHR/SHARE
SYS$SHARE:SSL111$LIBCRYPTO_SHR/SHARE
```

Building an application using 32-Bit APIs

To build (compile and link) an example program using the 32-bit APIs,
enter the following commands:

```
$ CC/PREFIX=ALL SAMPLE.C
$ LINK/MAP SAMPLE,LINKER_OPT/OPTIONS

In these commands, LINKER_OPT.OPT is a simple text file that contains
the following lines:

SYS$SHARE:SSL111$LIBSSL_SHR32/SHARE
SYS$SHARE:SSL111$LIBCRYPTO_SHR32/SHARE


Release Notes
-------------
This section contains notes on the current release of VSI SSL111 for
OpenVMS.

The TLS1_ALLOW_EXPERIMENTAL_CIPHERSUITES experimental ciphers are
enabled in VSI SSL111 V1.1


Legal caution
-------------
SSL/TLS data transport requires encryption. Many governments, including
the United States, have restrictions on the import and export of
cryptographic algorithms. Please ensure that your use of VSI SSL111 is in
compliance with all national and international laws that apply to you.


VSI SSL111 APIs not backward compatible
---------------------------------------
VSI SSL111 V1.1 for OpenVMS is based on the 1.1.1i baselevel of OpenSSL.
Some of the OpenSSL API, data structures, and commands have changed from
the previous VSI SSL V1.4 and VSI SSL1 product versions.

VSI cannot guarantee the backward compatibility of VSI SSL111 V1.1 with
VSI SSL V1.4 or VSI SSL1.

Applications will have to be recompiled and re-linked in order to make
use of the latest VSI SSL111 V1.1 header files and shareable images.

Note that the VSI SSL111 shareable images names are different from
VSI SSL 1.4 or VSI SSL1. Refer to the "Co-existence and major changes
between VSI SSL V1.4, VSI SSL1m and VSI SSL111 V1.1" for details.


Preserve configuration files before manually uninstalling VSI SSL111
--------------------------------------------------------------------
Preserving configuration files is not necessary when you perform a
regular upgrade or reinstallation of VSI SSL111 using the PRODUCT INSTALL
command.

However, if you intend to uninstall VSI SSL111 and wish to preserve
any modifications to the VSI SSL111 configuration files you should back
up these files to a different disk or directory before you  enter the
PRODUCT REMOVE command to remove the VSI SSL111 kit. If you do not take
a backup then any changes you made to OPENSSL-VMS.CNF and OPENSSL.CNF
will be lost when you perform the PRODUCT REMOVE.

When you have completed the reinstallation of VSI SSL111, move the saved
items back into the VSI SSL111 directory structure.


Configuration command procedure template files
----------------------------------------------
The configuration files included in the VSI SSL111 kit are named
OPENSSL.CNF_TEMPLATE and OPENSSL-VMS.CNF_TEMPLATE. This prevents PCSI
from overwriting the .CNF files and allows you to preserve any
modifications you made to OPENSSL.CNF and OPENSSL-VMS.CNF if you
installed a previous release of VSI SSL111 for OpenVMS.
```

If you are upgrading from a previous version of VSI SSL111, after you install the VSI SSL111 kit, compare the new .CNF_TEMPLATE files with your existing .CNF files and add any new information as required.

If you did not previously install a VSI SSL111 for OpenVMS kit, both the .CNF_TEMPLATE and .CNF files are provided.

VSI SSL111 requirement to install on system disk

The option to install to a location other than the system disk is no longer available. If you download VSI SSL111 and install it as a layered product, it must be installed on the system disk.

Shutdown VSI SSL111 before installing on common system disk

Before installing VSI SSL111 to a common system disk in a cluster, you must first shutdown VSI SSL111 by entering the following command on each node in the cluster:

```
$ @SYS$STARTUP:SSL111$SHUTDOWN
```

Shutting down VSI SSL111 deassigns logical names and removes installed shareable images that may interfere with the installation.

After the installation is complete, start VSI SSL111 by entering the following command on each node in the cluster:

```
$ @SYS$STARTUP:SSL111$STARTUP
```

Note: If you are installing on a common cluster disk and not a common system disk, omit the SYS$STARTUP logical name and specify the specific startup directory in the shutdown and startup commands. For example:

```
$ @device:[directory.SYS$STARTUP]SSL111$SHUTDOWN
$ @device:[directory.SYS$STARTUP]SSL111$STARTUP
```

OpenSSL version command displays VSI SSL111 for OpenVMS version

The OpenSSL command line utility command VERSION includes the VSI SSL111 for OpenVMS version. The OpenSSL VERSION command displays output similar to the following:

```
OpenSSL> version
OpenSSL 1.1.1i xx xxx xxxx
SSL111 for OpenVMS V1.1  xx xxx xxxx
```

Certificate tool cannot have simultaneous users

Only one user/process should use the Certificate Tool at a time. The tool does not have a locking mechanism to prevent unsynchronized accesses of the database and serial file, which could cause database corruption.

Protect certificates and keys

When you create certificates and keys with the Certificate Tool, take care to ensure that the keys are properly protected to allow only the owner of the keys to use them. A private key should be treated like a password. You can use OpenVMS file protections to protect the key file, or you can use ACLs to protect individual key files within a common directory.

Environment Variables

OpenSSL environmental variables have two formats, as follows:

```
$var
${var}
```

In order for these variables to be parsed properly and not be confused with logical names, VSI SSL111 for OpenVMS only accepts the ${var} format.

IDEA, RC5 and MDC2 symmetric cipher algorithms not supported

The IDEA, RC5 and MDC2 symmetric cipher algorithms are not provided. These algorithms are under copyright protection, and VSI does not have the right to use these algorithms.

APIs RAND_egd, RAND_egd_bytes, and RAND_query_egd_bytes not supported

The RAND_egd(), RAND_egd_bytes(), and RAND_query_egd_bytes() APIs are not available on OpenVMS.

To obtain a secure random seed on OpenVMS, use the RAND_poll() API.

Documentation from the OpenSSL Website

The documentation on the OpenSSL website is located at https://www.openssl.org/docs/. It is likely that the API and command line documentation shipped with this kit will differ from the documentation on the OpenSSL website at some point. If such a situation arises, you should consider the API documentation on the OpenSSL website to have precedence over the documentation included in this kit.

Extra Certificate Files ? *PEM

When you sign a certificate request using either the Certificate Tool or the OpenSSL utility, you may notice that an extra certificate is produced with a name similar to SSL$CRT01.PEM. This certificate is the same as the certificate that you produced with the name you chose. These extra files are the result of the OpenSSL demonstration Certificate Authority (CA) capability, and are used as a CA accounting function. These extra files are kept by the CA and can be used to generate Certificate Revocation Lists (CRLs) if the certificate becomes compromised.

-- end of file --