VSI SSL3 for OpenVMS

Release Notes

Publication Date: April 2024

Software Version: VSI SSL3 for OpenVMS Version 3.0-13

Kit Names: AXPVMS-SSL3-V0300-13-1
I64VMS-SSL3-V0300-13-1
X86VMS-SSL3-V0300-13-1

VMS Software, Inc. (VSI)
Boston, Massachusetts, USA

Table of Contents

1. Introduction 3
1.1. Installation Requirements and Prerequisites 3
1.2. OpenSSL Documentation From Open Group 4
2. Coexistence and Differences Between VSI SSL V1.4, VSI SSL1, VSI SSL111, VSI SSL3, and VSI SSL31 (Alpha and IA-64 only) 4
2.1. Logical Names 5
2.2. Directory Names 6
2.3. Command Procedure Names 6
2.4. Library Names 6
2.5. Migrating Certificate Store From VSI SSL V1.4, VSI SSL1 or VSI SSL111 V1.1 to VSI SSL3 V3.0 7
3. Coexistence and Differences Between VSI SSL111, VSI SSL3, and VSI SSL31 (x86-64 only) 8
3.1. Logical Names 9
3.2. Directory Names 9
3.3. Command Procedure Names 10
3.4. Library Names 10
3.5. Migrating Certificate Store From VSI SSL111 V1.1 to VSI SSL3 V3.0 10
3.6. VSI SSL3 APIs Not Backward Compatible 11
4. Release Notes 11
4.1. FIPS Module Support for x86-64 Architecture 11
4.2. Preserve Configuration Files Before Manually Uninstalling VSI SSL3 11
4.3. Configuration Command Procedure Template Files 12
4.4. VSI SSL3 Must Be Installed on System Disk 12
4.5. Shutdown VSI SSL3 Before Installing on Common System Disk 12
4.6. OpenSSL Version Command Displays VSI SSL3 for OpenVMS Version 12
4.7. Certificate Tool Cannot Have Simultaneous Users 13
4.8. Protect Certificates and Keys 13
4.9. Environment Variables 13
4.10. IDEA, RC5, MDC2 Symmetric Cipher Algorithms Not Supported 13
4.11. RAND_egd, RAND_egd_bytes, RAND_query_egd_bytes Not Supported 13
4.12. Documentation from the OpenSSL Website 13
4.13. .PEM Certificate Files 13
5. Installing VSI SSL3 for OpenVMS 14
5.1. Starting the Installation 14
5.2. Stopping the Installation 14
5.3. Post-Installation Configuration 15
5.3.1. Defining Logical Names and Foreign Commands 15
5.3.2. Preserving Customized Command Procedures Files 16
5.3.3. Optional Post-Installation Steps 16
5.4. VSI SSL3 Directory Structure 17
6. Building VSI SSL3 Applications 17
6.1. Building an Application Using 64-Bit APIs 18
6.2. Building an Application Using 32-Bit APIs 18

1. Introduction

VMS Software, Inc. (VSI) is pleased to introduce VSI SSL3 for OpenVMS Version 3.0-13. This document contains the hardware and software prerequisites, installation instructions, post-installation tasks, instructions for building your application, VSI SSL3 directory structure, and release notes for VSI SSL3 for OpenVMS Version 3.0-13. The information in this document applies to VSI SSL3 running on VSI OpenVMS Alpha, VSI OpenVMS IA-64, and VSI OpenVMS x86-64 systems. Note that certain sections of this document only apply to specific architectures.

VSI SSL3 for OpenVMS Version 3.0-13 is based on Open Source OpenSSL version 3.0.13 from https://www.openssl.org/.

For information about SSL3 vulnerabilities, refer to https://www.openssl.org/news/vulnerabilities.html.

Legal Caution

SSL/TLS data transport requires encryption. Many countries, including the United States, have restrictions on the import and export of cryptographic algorithms. Please ensure that your use of VSI SSL3 is in compliance with all national and international laws that apply to you.

1.1. Installation Requirements and Prerequisites

This section lists hardware and software requirements for VSI SSL3 for OpenVMS Version 3.0-13:

Disk Space Requirements

- On VSI OpenVMS x86-64 systems, the VSI SSL3 for OpenVMS Version 3.0-13 kit requires approximately 700,000 blocks of working disk space to install. Once installed, the software occupies approximately 610,000 blocks of disk space.
- On VSI OpenVMS Alpha and VSI OpenVMS IA-64 systems, the VSI SSL3 for OpenVMS Version 3.0-13 kit requires approximately 200,000 blocks of working disk space to install. Once installed, the software occupies approximately 120,000 blocks of disk space.

Operating System Requirements

Depending on the architecture, VSI SSL3 for OpenVMS Version 3.0-13 requires the following operating system versions:

- VSI OpenVMS Alpha V8.4-2L1 or later
- VSI OpenVMS IA-64 V8.4-2L1 or later
- VSI OpenVMS x86-64 V9.2-1 or later

For VSI OpenVMS Alpha and IA-64, RTL ECO kit 6 or later must be installed.

Account Quotas and System Parameters

There are no specific requirements for account quotas and system parameters for installing or using VSI SSL3 for OpenVMS.

1.2. OpenSSL Documentation From Open Group

Documentation for the OpenSSL project and the Open Group is available at http://www.openssl.org.

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

File specification format

The OpenSSL documentation shows example file specifications in the UNIX format, which is different from the OpenVMS format. For example, a UNIX file specification "/dka100/foo/bar/file.dat" would be 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 character (~) is an abbreviation for SYS$LOGIN. For example, if your default directory is SYS$LOGIN, the UNIX pathname "~/.openssl/profile/prefs.js" is going to be equivalent to [._OPENSSL.PROFILE]PREFS.JS on OpenVMS.

2. Coexistence and Differences Between VSI SSL V1.4, VSI SSL1, VSI SSL111, VSI SSL3, and VSI SSL31 (Alpha and IA-64 only)

Warning

The information in this section applies to VSI SSL running on VSI OpenVMS Alpha systems and VSI OpenVMS IA-64 systems.

The SSL product name has been changed to SSL3 to allow VSI SSL V1.4, (based on OpenSSL 0.9.8 stream), SSL1 (based on OpenSSL 1.0.2 stream), VSI SSL111 V1.1 (based on OpenSSL 1.1.1 stream), VSI SSL3 V3.0 (based on OpenSSL 3.0 stream), and VSI SSL31 V3.1 (based on OpenSSL 3.1 stream) to coexist on the same system.

VSI recommends that the VSI SSL3 V3.0, VSI SSL111 V1.1, VSI SSL1, and VSI SSL V1.4 products remain installed until any applications dependent on VSI SSL have been recompiled and relinked against VSI SSL3.

Once all the dependent products/components have been successfully migrated to VSI SSL3 V3.0, the earlier VSI SSL V1.4, SSL1, and SSL111 V1.1 kits can be removed.

Below is an example snapshot of coexistence.

```
$ 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 V1.1-1WA         Full LP     Installed
VSI I64VMS SSL3 V3.0-12            Full LP     Installed
```

```
VSI I64VMS SSL31 V3.1-4              Full LP     Installed
---------------------------------- ----------- ---------
5 items found
```

2.1. Logical Names

All logical names associated with VSI SSL3 V3.0 are prefixed with "SSL3$". The following is a comparison of system-level logical names that are defined for VSI SSL V1.4 and VSI SSL3 V3.0 (a similar comparison can be made between SSL3 and SSL111):

VSI SSL V1.4-503 Logical Names	**VSI SSL3 V3.0-13 Logical Names**
OPENSSL = SSL$INCLUDE:	OPENSSL = SSL3$INCLUDE:
SSL$CERT = SSL$ROOT:[DEMOCA.CERTS]	SSL3$CERT = SSL3$ROOT:[DEMOCA.CERTS]
SSL$CERTS = SSL$ROOT:[DEMOCA.CERTS]	SSL3$CERTS = SSL3$ROOT:[DEMOCA.CERTS]
SSL$COM = SSL$ROOT:[COM]	SSL3$COM = SSL3$ROOT:[COM]
SSL$CONF = SSL$ROOT:[DEMOCA.CONF]	SSL3$CONF = SSL3$ROOT:[DEMOCA.CONF]
SSL$CRL = SSL$ROOT:[DEMOCA.CRL]	SSL3$CRL = SSL3$ROOT:[DEMOCA.CRL]
SSL$EXAMPLES = SYS$COMMON:[SYSHLP.EXAMPLES.SSL]	SSL3$EXAMPLES = SYS$COMMON:[SYSHLP.EXAMPLES.SSL3]
On VSI OpenVMS Alpha: SSL$EXE = SSL$ROOT:[ALPHA_EXE] On VSI OpenVMS IA-64: SSL$EXE = SSL$ROOT:[IA64_EXE]	On VSI OpenVMS Alpha: SSL3$EXE = SSL3$ROOT:[ALPHA_EXE] On VSI OpenVMS IA-64: SSL3$EXE = SSL3$ROOT:[IA64_EXE]
SSL$INCLUDE = SSL$ROOT:[INCLUDE]	SSL3$INCLUDE = SSL3$ROOT:[INCLUDE]
SSL$KEY = SSL$ROOT:[DEMOCA.CERTS]	SSL3$KEY = SSL3$ROOT:[DEMOCA.CERTS]
SSL$KEYS = SSL$ROOT:[DEMOCA.CERTS]	SSL3$KEYS = SSL3$ROOT:[DEMOCA.CERTS]
SSL3$LIB = SSL3$ROOT:[LIB]	SSL3$MODULES = SSL3$ROOT:[MODULES]
SSL$PRIVATE = SSL$ROOT:[DEMOCA.PRIVATE]	SSL3$PRIVATE = SSL3$ROOT:[DEMOCA.PRIVATE]
SSL$ROOT = SYS$SYSDEVICE:[VMS$COMMON.SSL.]	SSL3$ROOT = SYS$SYSDEVICE:[VMS$COMMON.SSL3.]

These logical names get defined by invoking SYS$STARTUP:SSL$STARTUP.COM and SYS$STARTUP:SSL3$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, VSI SSL111 V1.1, and VSI SSL3 V3.0 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 that use the "SSL$...", "SSL1$...", or "SSL111$..." logical names, ensure that they are modified to use the "SSL3$..." logical names when migrating from VSI SSL V1.4, VSI SSL1, or VSI SSL111 V1.1 to VSI SSL3 V3.0.

2.2. Directory Names

The top level directory structure for VSI SSL3 V3.0 is SYS$SYSDEVICE:[VMS$COMMON.SSL3]. The top level directory structures for VSI SSL V1.4, VSI SSL1, and VSI SSL111 V1.1 (if installed) remain as SYS$SYSDEVICE:[VMS$COMMON.SSL], SYS$SYSDEVICE:[VMS$COMMON.SSL1], and SYS$SYSDEVICE:[VMS$COMMON.SSL111] respectively.

VSI SSL3 V3.0 example programs are located in SYS$COMMON:[SYSHLP.EXAMPLES.SSL3] directory.

If there are any custom command procedures on your system that reference the [SSL], [SSL1], or [SSL111] directories, ensure that they are modified to use the new [SSL3] directory when migrating from VSI SSL V1.4, VSI SSL1, or VSI SSL111 V1.1 to VSI SSL3 V3.0.

2.3. Command Procedure Names

The relevant command procedure names are prefixed with SSL3 for the VSI SSL3 V3.0 product. For example:

SYS$STARTUP:SSL3$STARTUP.COM
SSL3$COM:SSL3$CERT_TOOL.COM

Command procedures for VSI SSL V1.4, VSI SSL1 and VSI SSL111 V1.1 are prefixed with SSL, SSL1, and SSL111 respectively.

If there are any custom command procedures on your system invoking the "SSL$...", "SSL1$...", or "SSL111$..." command procedures, ensure that they are modified to invoke "SSL3$..." command procedures when migrating from VSI SSL V1.4, VSI SSL1, or VSI SSL111 V1.1 to VSI SSL3 V3.0.

2.4. Library Names

Library names for VSI SSL3 V3.0 are prefixed with SSL3$ as follows:

SYS$SHARE:SSL3$LIBSSL_SHR.EXE
SYS$SHARE:SSL3$LIBCRYPTO_SHR.EXE
SYS$SHARE:SSL3$LIBSSL_SHR32.EXE
SYS$SHARE:SSL3$LIBCRYPTO_SHR32.EXE

Library names for VSI SSL V1.4, VSI SSL1 and VSI SSL111 V1.1 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

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

Applications that are linked with VSI SSL V1.4, VSI SSL1 or VSI SSL111 V1.1 will continue using VSI SSL V1.4, VSI SSL1 or VSI SSL111 V1.1 libraries and applications that are linked with VSI SSL3 V3.0 product will use the new libraries shipped with VSI SSL3 product.

The logical name "OPENSSL" is used commonly by VSI SSL3 V3.0, 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 (SSL3$INCLUDE:, SSL111$INCLUDE:, SSL1$INCLUDE: or SSL$INCLUDE:) before rebuilding applications.

2.5. Migrating Certificate Store From VSI SSL V1.4, VSI SSL1 or VSI SSL111 V1.1 to VSI SSL3 V3.0

The top level directory structure of VSI SSL3 V3.0 is modified to SYS$SYSDEVICE:[VMS$COMMON.SSL3] from SYS$SYSDEVICE:[VMS$COMMON.SSL], SYS$SYSDEVICE:[VMS$COMMON.SSL1], or SYS$SYSDEVICE:[VMS$COMMON.SSL111], which are the top level directories for VSI SSL 1.4, VSI SSL1, and VSI SSL111 V1.1 respectively.

In case there is a certificate store manually created in SYS$SYSDEVICE:[VMS$COMMON.SSL.DEMOCA...], SYS$SYSDEVICE:[VMS$COMMON.SSL1.DEMOCA...], or SYS$SYSDEVICE:[VMS$COMMON.SSL111.DEMOCA...], copy the certificate store to SYS$SYSDEVICE:[VMS$COMMON.SSL3.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 SSL3 V3.0, this hash is modified from the MD5 to the SHA-1 algorithm. Due to this modification, validation of certificates will fail with SSL3 if we use the same hash names. To avoid this, manually rename the certificate file name to use the new hash.

An example of moving a certificate from VSI SSL V1.4 to VSI SSL3 V3.0 is as follows:

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

2. Assume we have a certificate file 438F16D6.0 in SSL$ROOT:[DEMOCA.CERTS]. The "438F16D6" part of this certificate file name 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-----
$
```

3. After installing VSI SSL3 V3.0, executing the **`openssl x509 -hash`** command from SSL3 will give you "37d8de08" which is a SHA-1 hash of the certificate subject.

```
$ @SSL3$COM:SSL3$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-----
$
```

4. You will have to use a certificate file name having "37d8de08" if you wish to use this certificate store with VSI SSL3 V3.0:

```
$ COPY SSL$ROOT:[DEMOCA.CERTS]438F16D6.0 -
SSL3$ROOT:[DEMOCA.CERTS]37d8de08.0
```

   or

```
$ openssl x509 -hash -in SSL$ROOT:[DEMOCA.CERTS]438F16D6.0 -out
SSL3$ROOT:[DEMOCA.CERTS]37d8
```

   Here, we are assuming that SSL3$ROOT:[DEMOCA.CERTS] is the new certificate store directory used with VSI SSL3 V3.0.

5. Repeat steps 2 and 3 for all certificates in the certificate store.

6. Note that certificate verification (using either the **`openssl verify`** command or the OpenSSL APIs) will work with VSI SSL3 V3.0 (for the above example) only if the certificate name in the certificate store is "37d8de08.0".

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

For more information, refer to https://www.openssl.org/docs/man3.0/man1/x509.html and https://www.openssl.org/docs/man3.0/man1/openssl-verify.html

3. Coexistence and Differences Between VSI SSL111, VSI SSL3, and VSI SSL31 (x86-64 only)

Warning

The information in this section applies to VSI SSL running on VSI OpenVMS x86-64 systems.

The SSL product name has been changed to SSL3 to allow VSI SSL111 V1.1 (based on OpenSSL 1.1.1 stream), VSI SSL3 V3.0 (based on OpenSSL 3.0 stream), and VSI SSL31 V3.1 (based on OpenSSL 3.1 stream) to coexist on the same system.

VSI recommends that the VSI SSL3 V3.0 and VSI SSL111 V1.1 products remain installed until any applications dependent on VSI SSL have been recompiled and relinked against VSI SSL3.

Once all the dependent products/components have been successfully migrated to VSI SSL3 V3.0, the earlier VSI SSL111 V1.1 kit can be removed.

The following is a snapshot of coexistence:

```
$ prod show prod SSL*
---------------------------------- ----------- ---------
PRODUCT                            KIT TYPE    STATE
---------------------------------- ----------- ---------
VSI X86VMS SSL111 V1.1-1U          Full LP     Installed
VSI X86VMS SSL3 V3.0-10            Full LP     Installed
VSI X86VMS SSL31 V3.1-4            Full LP     Installed
---------------------------------- ----------- ---------
3 items found
```

3.1. Logical Names

All logical names associated with VSI SSL3 V3.0 are prefixed with SSL3$. The following is a comparison of system-level logical names that are defined for VSI SSL111 V1.1 and VSI SSL3 V3.0:

VSI SSL111 V1.1.1M Logicals

```
"OPENSSL" = "SSL111$INCLUDE:"
"SSL111$CERT" = "SSL111$ROOT:[DEMOCA.CERTS]"
"SSL111$CERTS" = "SSL111$ROOT:[DEMOCA.CERTS]"
"SSL111$COM" = "SSL111$ROOT:[COM]"
"SSL111$CONF" = "SSL111$ROOT:[DEMOCA.CONF]"
"SSL111$CRL" = "SSL111$ROOT:[DEMOCA.CRL]"
"SSL111$EXAMPLES" =
        "SYS$COMMON:[SYSHLP.EXAMPLES.SSL]"
"SSL111$EXE" = "SSL111$ROOT:[X86_64_EXE]"
"SSL111$INCLUDE" = "SSL111$ROOT:[INCLUDE]"
"SSL111$KEY" = "SSL111$ROOT:[DEMOCA.CERTS]"
"SSL111$KEYS" = "SSL111$ROOT:[DEMOCA.CERTS]"
"SSL111$LIB" = "SSL111$ROOT:[LIB]"

"SSL111$PRIVATE" = "SSL111$ROOT:[DEMOCA.PRIVATE]"
"SSL111$ROOT" = "SYS$SYSDEVICE:[VMS$COMMON.SSL.]"
```

VSI SSL3 V3.0-13 Logicals

```
"OPENSSL" = "SSL3$INCLUDE:"
"SSL3$CERT" = "SSL3$ROOT:[DEMOCA.CERTS]"
"SSL3$CERTS" = "SSL3$ROOT:[DEMOCA.CERTS]"
"SSL3$COM" = "SSL3$ROOT:[COM]"
"SSL3$CONF" = "SSL3$ROOT:[DEMOCA.CONF]"
"SSL3$CRL" = "SSL3$ROOT:[DEMOCA.CRL]"
"SSL3$EXAMPLES" =
        "SYS$COMMON:[SYSHLP.EXAMPLES.SSL3]"
"SSL3$EXE" = "SSL3$ROOT:[X86_64_EXE]"
"SSL3$INCLUDE" = "SSL3$ROOT:[INCLUDE]"
"SSL3$KEY" = "SSL3$ROOT:[DEMOCA.CERTS]"
"SSL3$KEYS" = "SSL3$ROOT:[DEMOCA.CERTS]"
"SSL3$LIB" = "SSL3$ROOT:[LIB]"
"SSL3$MODULES" = "SSL3$ROOT:[MODULES]"
"SSL3$PRIVATE" = "SSL3$ROOT:[DEMOCA.PRIVATE]"
"SSL3$ROOT" = "SYS$SYSDEVICE:[VMS$COMMON.SSL3.]"
```

These logical names get defined by invoking SYS$STARTUP:SSL111$STARTUP.COM and SYS$STARTUP:SSL3$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 SSL3 V3.0 and VSI SSL111 V1.1 versions co-exist, the OPENSSL logical name will point to the version of product that was started last.

If there are any custom command procedures on your system that use the SSL111$... logical names, ensure that they are modified to use the SSL3$... logical names when migrating from VSI SSL111 V1.1 to VSI SSL3 V3.0.

3.2. Directory Names

The top level directory structure for VSI SSL3 V3.0 is SYS$SYSDEVICE:[VMS$COMMON.SSL3]. The top level directory structures for VSI SSL111 V1.1 (if installed) remain as SYS$SYSDEVICE:[VMS$COMMON.SSL111].

VSI SSL3 V3.0 example programs are located in the SYS$COMMON:[SYSHLP.EXAMPLES.SSL3] directory.

If there are any custom command procedure files on your system that reference the [SSL111] directories, ensure that they are modified to use the new [SSL3] directory when migrating from VSI SSL111 V1.1 to VSI SSL3 V3.0.

3.3. Command Procedure Names

The names of the command procedures relevant for VSI SSL3 V3.0 are prefixed with SSL3, for example:

SYS$STARTUP:SSL3$STARTUP.COM
SSL3$COM:SSL3$CERT_TOOL.COM

The names of the command procedures relevant for VSI SSL111 V1.1 are prefixed with SSL111.

If there are any custom command procedure files on your system that invoke the SSL111$... command procedures, ensure that they are modified to invoke the SSL3$... command procedures when migrating from VSI SSL111 V1.1 to VSI SSL3 V3.0.

3.4. Library Names

Library names for VSI SSL3 V3.0 are prefixed with SSL3$ as follows:

SYS$SHARE:SSL3$LIBSSL_SHR.EXE
SYS$SHARE:SSL3$LIBCRYPTO_SHR.EXE
SYS$SHARE:SSL3$LIBSSL_SHR32.EXE
SYS$SHARE:SSL3$LIBCRYPTO_SHR32.EXE

Library names for VSI SSL111 V1.1 remain unchanged:

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

Applications that are linked with VSI SSL111 V1.1 will continue using VSI SSL111 V1.1 libraries and applications that are linked with VSI SSL3 V3.0 product will use the new libraries shipped with VSI SSL3 product.

The logical name OPENSSL is used commonly by VSI SSL3 V3.0 and VSI SSL111 V1.1. Care must therefore be taken to identify that this logical name is defined to the appropriate path (SSL3$INCLUDE: or SSL111$INCLUDE:) before rebuilding applications.

3.5. Migrating Certificate Store From VSI SSL111 V1.1 to VSI SSL3 V3.0

The top level directory structure of VSI SSL3 V3.0 is modified to SYS$SYSDEVICE:[VMS$COMMON.SSL3] from SYS$SYSDEVICE:[VMS$COMMON.SSL111], which is the top level directory for VSI SSL111 V1.1).

In case there is a certificate store manually created in SYS$SYSDEVICE:[VMS$COMMON.SSL111.DEMOCA...], copy the certificate store to SYS$SYSDEVICE:[VMS$COMMON.SSL3.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).

3.6. VSI SSL3 APIs Not Backward Compatible

VSI SSL3 V3.0 for OpenVMS is based on the 3.0.13 base-level of the OpenSSL release. Some of the OpenSSL API, data structures, and commands have changed from the previous VSI SSL111 V1.1 product versions.

VSI cannot guarantee the backward compatibility of VSI SSL3 V3.0 with VSI SSL111 V1.1.

Applications will have to be recompiled and re-linked in order to make use of the latest VSI SSL3 V3.0 header files and shareable images.

Note that the VSI SSL3 shareable images names are different from VSI SSL111 V1.1.

4. Release Notes

This section contains notes on the current release of VSI SSL3 for OpenVMS.

4.1. FIPS Module Support for x86-64 Architecture

VSI SSL 3.0-13 for OpenVMS x86-64 supports a FIPS module based on the Open-Source OpenSSL 3.0 FIPS module – /docs/man3.0/man7/fips_module.html [https://www.openssl.org/docs/man3.0/man7/fips_module.html]. Note that the FIPS module is not enabled by default, so applications using OpenSSL 3.0.13 are not forced to use only FIPS-compliant algorithms, and all applications using SSL 3.0.13 should run without any errors. However, it is possible that some applications may fail if the FIPS module is enabled; any such applications will need to include support for FIPS-compliant algorithms to operate correctly with the FIPS module enabled.

To enable the FIPS module and force all applications to use only FIPS-compliant algorithms, the FIPS module configuration file named SSL3$ROOT:[000000]FIPSMODULE.CNF_AUTOGEN should be generated (as it is not generated automatically during system installation) and the OPENSSL_CONF logical name should be defined to point to a pre-made OpenSSL configuration file named SSL3$ROOT:[000000]OPENSSL-FIPS.CNF. To do so, the following commands can be used:

```
$ MCR SSL3$EXE:OPENSSL fipsinstall -out -
_$ "SSL3$ROOT:[000000]FIPSMODULE.CNF_AUTOGEN" -module SSL3$MODULES:FIPS.EXE
$ DEFINE OPENSSL_CONF SSL3$ROOT:[000000]OPENSSL-FIPS.CNF
```

4.2. Preserve Configuration Files Before Manually Uninstalling VSI SSL3

Preserving configuration files is not necessary when you perform a regular upgrade or reinstallation of VSI SSL3 using the **PRODUCT INSTALL** command. However, if you intend to uninstall VSI SSL3 and wish to preserve any modifications to the VSI SSL3 configuration files you should back up these files to a different disk or directory before you enter **PRODUCT REMOVE** to remove the VSI SSL3 kit. If you do not create a backup, any changes you made to OPENSSL-VMS.CNF and OPENSSL.CNF will be lost.

When you have completed the reinstallation of VSI SSL3, move the saved items back into the VSI SSL3 directory structure.

4.3. Configuration Command Procedure Template Files

The configuration files included in the VSI SSL3 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 SSL3 for OpenVMS.

If you are upgrading from a previous version of VSI SSL3, after you install the VSI SSL3 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 SSL3 for OpenVMS kit, both the .CNF_TEMPLATE and .CNF files are provided.

4.4. VSI SSL3 Must Be Installed on System Disk

The option to install to a location other than the system disk is no longer available. If you download VSI SSL3 and install it as a layered product, it must be installed on the system disk.

4.5. Shutdown VSI SSL3 Before Installing on Common System Disk

Before installing VSI SSL3 to a common system disk in a cluster, you must first shutdown VSI SSL3 by entering the following command on each node in the cluster:

```
$ @SYS$STARTUP:SSL3$SHUTDOWN
```

Shutting down VSI SSL3 deassigns logical names and removes installed shareable images that may interfere with the installation.

After the installation is complete, start VSI SSL3 by entering the following command on each node in the cluster:

```
$ @SYS$STARTUP:SSL3$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$STARTUP]SSL3$SHUTDOWN
$ @device:[directory.SYS$STARTUP]SSL3$STARTUP
```

4.6. OpenSSL Version Command Displays VSI SSL3 for OpenVMS Version

The OpenSSL command line utility command **`version`** includes the VSI SSL3 for OpenVMS version. The OpenSSL **`version`** command displays output similar to the following:

```
OpenSSL> version
OpenSSL 3.0.13 xx xxx xxxx (Library: OpenSSL 3.0.13 xx xxx xxxx)
```

```
SSL3 for OpenVMS V3.0(xx) xxx xx xxxx (Library: SSL3 for OpenVMS V3.0(xx) xxx xx
 xxxx)
```

4.7. 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.

4.8. 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.

4.9. Environment Variables

Generally, the OpenSSL environmental variables can exist in one of two formats: `$var` or `${var}`

In order for these variables to be parsed properly and not be confused with logical names, VSI SSL3 for OpenVMS only accepts the `${var}` format. Additionally, *.CNF files must contain .pragma dollarid:on, which allows using of dollar sign in variable names.

4.10. IDEA, RC5, 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.

4.11. RAND_egd, RAND_egd_bytes, 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.

4.12. 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.

4.13. .PEM Certificate Files

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.

5. Installing VSI SSL3 for OpenVMS

This section describes the process of installing and configuring VSI SSL3 for OpenVMS.

5.1. Starting the Installation

To install the VSI SSL3 V3.0 for OpenVMS kit, enter the following command:

```
$ PRODUCT INSTALL SSL3
```

You will see an output similar to the following:

```
Performing product kit validation of signed kits ...
%PCSI-I-CANNOTVAL, cannot validate $1$DGA85:
[username.ReleaseBuilds.SSL3.3013.X86.X86KIT]VSI-X86VMS-SSL3-V0300-13-1.PCSI$COMPRESSED;1
-PCSI-I-NOTSIGNED, product kit is not signed and therefore has no manifest file
The following product has been selected:
    VSI X86VMS SSL3 V3.0-13               Layered Product [Installed]
Do you want to continue? [YES]
```

Answer `YES` (the default option) to the `Do you want to continue?` prompt to start the installation.

During the installation, you will be asked to choose the installation options for the product. The output that you will see on your screen will be similar to the following:

```
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 X86VMS SSL3 V3.0-13: SSL3 for OpenVMS X86-64 V3.0-13 (Based on
 OpenSSL 3.0.13)
    Copyright 2024 VMS Software, Inc.
Do you want the defaults for all options? [YES]
Do you want to review the options? [NO]
Execution phase starting ...
The following product will be installed to destination:
    VSI X86VMS SSL3 V3.0-13               DISK$V921_EOWYN:[VMS$COMMON.]
Portion done: 0%...30%...60%...70%...80%...90%...100%
The following product has been installed:
    VSI X86VMS SSL3 V3.0-13               Layered Product
%PCSI-I-IVPEXECUTE, executing test procedure for VSI X86VMS SSL3 V3.0-13 ...
%PCSI-I-IVPSUCCESS, test procedure completed successfully
VSI X86VMS SSL3 V3.0-13: SSL3 for OpenVMS X86-64 V3.0-13 (Based on OpenSSL 3.0.13)
    Insert the following lines in SYS$MANAGER:SYSTARTUP_VMS.COM:
        @SYS$STARTUP:SSL3$STARTUP.COM
    Insert the following lines in SYS$MANAGER:SYSHUTDWN.COM:
        @SYS$STARTUP:SSL3$SHUTDOWN.COM
    Review the Installation Guide and Release Notes for post install directions.
    Review the Installation Guide and Release Notes for post upgrade verification
 suggestions.
    Refer to SYS$HELP:SSL30-13-X86.RELEASE_NOTES for more information.
```

5.2. Stopping the Installation

You can stop the installation procedure at any moment by pressing **Ctrl/Y**. Note, however, that before restarting the installation, you will have to manually reverse any changes to the system that occurred during the aborted installation. To do that, enter the following command:

```
$ PRODUCT REMOVE SSL3
```

After all traces of SSL3 have been removed, start the installation as described in Section 5, “Installing VSI SSL3 for OpenVMS”.

5.3. Post-Installation Configuration

After the installation is complete, perform the tasks described in this section to fully configure VSI SSL3.

5.3.1. Defining Logical Names and Foreign Commands

Follow these steps:

1. Define the SSL3$ logical names and install shareable images by adding the line `@SSL3$STARTUP.COM` to the SYS$MANAGER:SYSTARTUP_VMS.COM file.

   If your SYS$MANAGER:SYSTARTUP_VMS.COM file already includes the line `@SSL111$STARTUP.COM`, you can either comment it out or conditionalize the command procedure as appropriate. For example:

   ```
   $ if f$search("sys$startup:ssl111$startup.com") .nes. ""
   $ then
   $    @sys$startup:ssl111$startup.com
   $ endif
   $ if f$search("sys$startup:ssl3$startup.com") .nes. ""
   $ then
   $    @sys$startup:ssl3$startup.com
   $ endif
   ```

   In the code snippet above, the SSL3$STARTUP.COM and SSL111$STARTUP.COM command procedures will automatically define the SSL3$ and SSL111$ executive-mode logical names in the SYSTEM logical name table. They will also install the SSL3 and SSL111 shareable images that reside in the [SYSLIB] directory into memory.

2. Ensure that SSL3$STARTUP.COM is invoked *after* SSL111$STARTUP.COM. Both command procedure files define the OPENSSL logical name that points to the include (header) file directory used when building applications using OpenSSL. Invoking SSL3$STARTUP.COM after its SSL111 counterpart ensures that the OPENSSL logical is defined to correctly point to the latest VSI SSL3 3.0 header files.

3. Add the line `@SSL3$SHUTDOWN.COM` to SYS$MANAGER:SYSHUTDWN.COM to remove installed images and deassign the SSL3$ logical names at system shutdown. If your SYS$MANAGER:SYSHUTDWN.COM already includes the line `SSL111$SHUTDOWN.COM`, conditionalize the script as appropriate. For example:

   ```
   $ if f$search("sys$startup:ssl111$shutdown.com") .nes. ""
   $ then
   $    @sys$startup:ssl111$shutdown.com
   $ endif
   $ if f$search("sys$startup:ssl3$shutdown.com") .nes. ""
   $ then
   $    @sys$startup:ssl3$shutdown.com
   $ endif
   ```

   For more information about SSL-related logical names, refer to Section 3.1, “Logical Names”.

4. Define the foreign commands that use the OPENSSL.EXE utility (such as openssl, ca, enc, req, and X509) by executing the following command:

```
$ @SSL3$COM:SSL3$UTILS
```

5.3.2. Preserving Customized Command Procedures Files

If you have at any point modified any of the SSL111 command procedure files on your system, may need to replicate those changes in the SSL3 command procedure files. Consider the following tips:

- Copy any manual changes from the site-specific SSL111$COM:SSL111$SYSTARTUP.COM to SSL3$COM:SSL3$SYSTARTUP.COM.
- Copy any manual changes from SYS$STARTUP:SSL111$STARTUP.COM to the site-specific SSL3$COM:SSL3$SYSTARTUP.COM. This command procedure will be invoked by SYS$STARTUP:SSL3$STARTUP.COM.
- Copy any manual changes from the site-specific SSL111$COM:SSL111$SYSHUTDOWN.COM to SSL3$COM:SSL3$SYSHUTDOWN.COM.
- Copy any manual changes from SYS$STARTUP:SSL111$SHUTDOWN.COM to the site-specific shutdown command procedure SSL3$COM:SSL3$SYSHUTDOWN.COM. This command procedure will be invoked by SYS$STARTUP:SSL3$SHUTDOWN.COM.
- Copy any manual changes from the OpenSSL configuration file SSL111$ROOT:[000000]OPENSSL.CNF to SSL3$ROOT:[000000]OPENSSL.CNF.
- Copy any manual changes from the OpenSSL configuration file SSL111$ROOT:[000000]OPENSSL-VMS.CNF to SSL3$ROOT:[000000]OPENSSL-VMS.CNF.
- If any other .CNF files from the previous releases are intended to be used with VSI SSL3 V3.0 on your system, insert `.pragma dollarid:on` as the first line to make sure the dollar sign character ($) not shielded by {} will be treated as usual character (not as substitution template) in OpenVMS paths.
- Migrate any SSL certificates store content to VSI SSL3 V3.0. For details, refer to Section 3.5, “Migrating Certificate Store From VSI SSL111 V1.1 to VSI SSL3 V3.0”.
- Migrate any applications built with VSI SSL111 V1.1 to VSI SSL3 V3.0 by rebuilding and relinking the application with the VSI SSL3 V3.0 header files and libraries.
- Migrate any command procedures using the VSI SSL111 V1.1 directories, command procedures, or logical names to point to VSI SSL3 V3.0 directories, command procedures, or logical names. For more information, refer to Section 3, “Coexistence and Differences Between VSI SSL111, VSI SSL3, and VSI SSL31 (x86-64 only)”.

5.3.3. Optional Post-Installation Steps

- Run the base Installation Verification Procedure (IVP) test by entering the following command:

  ```
  $ @SYS$TEST:SSL3$IVP.COM
  ```

- Start the Certificate Tool by entering the following command:

  ```
  $ @SSL3$COM:SSL3$CERT_TOOL
  ```

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

5.4. VSI SSL3 Directory Structure

The VSI SSL3 features the following directory structure:

Directory	Description
SYS$SYSDEVICE:[VMS$COMMON]	Root directory.
[SSL3]	Top-level directory created by default in SYS$SYSDEVICE:[VMS$COMMON].
[SSL3.X86_64_EXE]	Contains images for the x86-64 server platform.
[SSL3.COM]	Contains command procedures.
[SSL3.DEMOCA]	Contains demos for SSL's CA features
[SSL3.DEMOCA.CERTS]	Contains certificates and keys.
[SSL3.DEMOCA.CONF]	Contains configuration files.
[SSL3.DEMOCA.CRL]	Contains revoked certificates and CRLs.
[SSL3.DEMOCA.PRIVATE]	Contains private keys and random data.
[SSL3.DOC]	OpenSSL Group-provided documentation and information.
[SSL3.INCLUDE]	Contains C header (.H) files.
[SSL3.LIB]	Contains static libraries (.OLB) files.
[SSL3.MODULES]	Contains dynamically loadable OpenSSL modules (e.g. providers).
[SYS$STARTUP]	Contains startup and shutdown templates and files.
[SYSHLP]	Contains release notes.
[SYSHLP.EXAMPLES.SSL3]	Contains SSL crypto and secure session examples.
[SYSLIB]	Contains SSL shareable image files.
[SYSTEST]	Contains SSL3$IVP.COM test file.

Note

The VSI SSL3 example programs are located in SYS$COMMON:[SYSHLP.EXAMPLES.SSL3]. The logical name SSL3$EXAMPLES points to this directory.

6. Building VSI SSL3 Applications

VSI SSL3 for OpenVMS provides shareable images that contain 64-bit APIs as well as shareable images that contain 32-bit APIs. You can choose which API you wish to use when you compile your application. The list of these shareable images goes as follows:

SYS$SHARE:SSL3$LIBSSL_SHR.EXE	64-bit SSL APIs
SYS$SHARE:SSL3$LIBCRYPTO_SHR.EXE	64-bit Crypto APIs
SYS$SHARE:SSL3$LIBSSL_SHR32.EXE	32-bit SSL APIs
SYS$SHARE:SSL3$LIBCRYPTO_SHR32.EXE	32-bit Crypto APIs

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

The process of linking an application is the same for the 64-bit and 32-bit APIs. However, the options file would contain either the 64-bit or 32-bit references to the appropriate shareable image.

6.1. 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$SHARE:SSL3$LIBSSL_SHR/SHARE
SYS$SHARE:SSL3$LIBCRYPTO_SHR/SHARE
```

6.2. 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$SHARE:SSL3$LIBSSL_SHR32/SHARE
SYS$SHARE:SSL3$LIBCRYPTO_SHR32/SHARE
```