VSI SSL31 Version 3.1-4 for OpenVMS Installation Guide and Release Notes

March 2024

This document contains hardware and software prerequisites,
installation instructions, post-installation tasks,
instructions for building your application, the VSI SSL31 directory
structure, and release notes for VSI SSL31 Version 3.1-4 for OpenVMS.

The information in this file applies to VSI SSL31 running on
OpenVMS Alpha servers.

VSI SSL31 Version 3.1-4 for OpenVMS is based on Open Source OpenSSL version
3.1.4 from OpenSSL.org.

Vulnerabilities CVE/CAN:

Please refer to 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 SSL31 for OpenVMS kit requires approximately 200,000 blocks of
  working disk space to install. Once installed, the software occupies
  approximately 110,000 blocks of disk space.

- Software Prerequisites

  VSI SSL31 V3.1-4 for OpenVMS requires the following software.

- Operating System

  VSI OpenVMS Alpha server Version V8.4-2L1 or later.

- Account Quotas and System Parameters

  There are no specific requirements for account quotas and system parameters
  for installing or using VSI SSL31 for OpenVMS.

New Features in VSI SSL31 Version 3.1-4 for OpenVMS

VSI SSL31 Version 3.1-4 for OpenVMS is based on Open Source
OpenSSL Version 3.1 stream.

For more information see the changes log: https://www.openssl.org/news/cl31.txt

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

---

VSI SSL3 and VSI SSL31

The SSL31 product name has been introduced 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.

Following is a snapshot of coexistence:

```
$ PROD SHOW PROD SSL*
---------------------------------- ----------- ---------
PRODUCT                            KIT TYPE    STATE
---------------------------------- ----------- ---------
VSI AXPVMS SSL V1.4-503            Full LP     Installed
VSI AXPVMS SSL1 V1.0-2UA           Full LP     Installed
VSI AXPVMS SSL111 V1.1-1M          Full LP     Installed
VSI AXPVMS SSL3 V3.0-12            Full LP     Installed
VSI AXPVMS SSL31 V3.1-4             Full LP     Installed
---------------------------------- ----------- ---------
5 items found
```

Logical names:

All the logical names associated with VSI SSL31 V3.1 are prefixed with SSL31$. The following is a comparison of system-level logical names that are defined for VSI SSL V1.4 and VSI SSL31 V3.1 (a similar comparison can be made between SSL31 and SSL111):

VSI SSL V1.4-503 Logicals	VSI SSL31 V3.1-4 Logicals
"OPENSSL" = "SSL$INCLUDE:"	"OPENSSL" = "SSL31$INCLUDE:"
"SSL$CERT" = "SSL$ROOT:[DEMOCA.CERTS]"	"SSL31$CERT" = "SSL31$ROOT:[DEMOCA.CERTS]"
"SSL$CERTS" = "SSL$ROOT:[DEMOCA.CERTS]"	"SSL31$CERTS" = "SSL31$ROOT:[DEMOCA.CERTS]"
"SSL$COM" = "SSL$ROOT:[COM]"	"SSL31$COM" = "SSL31$ROOT:[COM]"
"SSL$CONF" = "SSL$ROOT:[DEMOCA.CONF]"	"SSL31$CONF" = "SSL31$ROOT:[DEMOCA.CONF]"
"SSL$CRL" = "SSL$ROOT:[DEMOCA.CRL]"	"SSL31$CRL" = "SSL31$ROOT:[DEMOCA.CRL]"
"SSL$EXAMPLES" = "SYS$COMMON:[SYSHLP.EXAMPLES.SSL]"	"SSL31$EXAMPLES" = "SYS$COMMON:[SYSHLP.EXAMPLES.SSL31]"
"SSL$EXE" = "SSL$ROOT:[ALPHA_EXE]"	"SSL31$EXE" = "SSL31$ROOT:[ALPHA_EXE]"
"SSL$INCLUDE" = "SSL$ROOT:[INCLUDE]"	"SSL31$INCLUDE" = "SSL31$ROOT:[INCLUDE]"
"SSL$KEY" = "SSL$ROOT:[DEMOCA.CERTS]"	"SSL31$KEY" = "SSL31$ROOT:[DEMOCA.CERTS]"
"SSL$KEYS" = "SSL$ROOT:[DEMOCA.CERTS]"	"SSL31$KEYS" = "SSL31$ROOT:[DEMOCA.CERTS]"
	"SSL31$LIB" = "SSL31$ROOT:[LIB]"
	"SSL31$MODULES" = "SSL31$ROOT:[MODULES]"
"SSL$PRIVATE" = "SSL$ROOT:[DEMOCA.PRIVATE]"	"SSL31$PRIVATE" = "SSL31$ROOT:[DEMOCA.PRIVATE]"
"SSL$ROOT" = "SYS$SYSDEVICE:[VMS$COMMON.SSL.]"	"SSL31$ROOT" = "SYS$SYSDEVICE:[VMS$COMMON.SSL31.]"

These logical names get defined by invoking SYS$STARTUP:SSL$STARTUP.COM and SYS$STARTUP:SSL31$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, VSI SSL3 V3.0 and VSI SSL31 V3.1 versions co-exist, the "OPENSSL" logical name will be pointed to the version of the product that was started last.

If there are any custom command procedures on your system using "SSL$...", "SSL1$...", "SSL111$..." or "SSL3$..." logical names, ensure that they are modified to use "SSL31$..." logical names when migrating from VSI SSL V1.4, VSI SSL1, VSI SSL111 V1.1 or VSI SSL3 V3.0 to VSI SSL31 V3.1.

Directory names:

The top level directory structure for VSI SSL31 V3.1 is SYS$SYSDEVICE:[VMS$COMMON.SSL31]. The top level directory structures for VSI SSL V1.4, VSI SSL1, VSI SSL111 V1.1 and VSI SSL3 V3.0 (if installed) remain as SYS$SYSDEVICE:[VMS$COMMON.SSL], SYS$SYSDEVICE:[VMS$COMMON.SSL1], SYS$SYSDEVICE:[VMS$COMMON.SSL111] and SYS$SYSDEVICE:[VMS$COMMON.SSL3], respectively.

VSI SSL31 V3.1 example programs are located in SYS$COMMON:[SYSHLP.EXAMPLES.SSL31] directory.

If there are any custom command procedures on your system referencing
the "[SSL]", "[SSL1]", "[SSL111]" or "[SSL31]" directories, ensure that they are
modified to use the new "[SSL31]" directory when migrating from VSI SSL V1.4,
VSI SSL1, VSI SSL111 V1.1 or VSI SSL3 V3.0 to VSI SSL31 V3.1.

Command procedure names:

The relevant command procedure names are prefixed with "SSL31" for
the VSI SSL31 V3.1 product. For example:

```
SYS$STARTUP:SSL31$STARTUP.COM
SSL31$COM:SSL31$CERT_TOOL.COM
```

Command procedures for VSI SSL V1.4, VSI SSL1, VSI SSL111 V1.1 and VSI SSL3 V3.0
are prefixed with "SSL", "SSL1", "SSL111" and "SSL3", respectively.

If there are any custom command procedures on your system invoking "SSL$...",
"SSL1$...", "SSL111$..." or "SSL3$..." command procedures, ensure that they are
modified to invoke "SSL31$..." command procedures when migrating from
VSI SSL V1.4, VSI SSL1, VSI SSL111 V1.1 or VSI SSL3 V3.0 to VSI SSL31 V3.1.

Library names:

Library names for VSI SSL31 V3.1 are prefixed with SSL31$ as follows:

```
 SYS$SHARE:SSL31$LIBSSL_SHR.EXE
 SYS$SHARE:SSL31$LIBCRYPTO_SHR.EXE
 SYS$SHARE:SSL31$LIBSSL_SHR32.EXE
 SYS$SHARE:SSL31$LIBCRYPTO_SHR32.EXE
```

Library names for VSI SSL V1.4, VSI SSL1 and VSI SSL111 V1.1 and VSI SSL3 V3.0
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

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

Applications that are linked with VSI SSL V1.4, VSI SSL1, VSI SSL111 V1.1 or
VSI SSL3 V3.0 will continue using VSI SSL V1.4, VSI SSL1, VSI SSL111 V1.1 or
VSI SSL3 V3.0 libraries, and applications that are linked with VSI SSL31 V3.1
product will use the new libraries shipped with the VSI SSL31 product.

The logical name "OPENSSL" is used commonly by VSI SSL31 V3.1, VSI SSL3 V3.0,
VSI SSL111 V1.1, VSI SSL1, and VSI SSL V1.4. Care must be taken to identify
that this logical name is defined to the appropriate path
(SSL31$INCLUDE:, SSL3$INCLUDE:, SSL111$INCLUDE:, SSL1$INCLUDE: or SSL$INCLUDE:)
before rebuilding applications.

Migrate certificate store from VSI SSL V1.4, VSI SSL1, VSI SSL111 V1.1 or VSI SSL3 V3.0 to VSI SSL31 V3.1:

- The top level directory structure of VSI SSL31 V3.1 is modified to SYS$SYSDEVICE:[VMS$COMMON.SSL31] from SYS$SYSDEVICE:[VMS$COMMON.SSL], SYS$SYSDEVICE:[VMS$COMMON.SSL1], SYS$SYSDEVICE:[VMS$COMMON.SSL111] or SYS$SYSDEVICE:[VMS$COMMON.SSL31] (Which are the top level directories for VSI SSL 1.4, VSI SSL1, VSI SSL111 V1.1 and VSI SSL3 V3.0 respectively).

  In case there is a certificate store manually created in SYS$SYSDEVICE:[VMS$COMMON.SSL.DEMOCA...], SYS$SYSDEVICE:[VMS$COMMON.SSL1.DEMOCA...]. SYS$SYSDEVICE:[VMS$COMMON.SSL111.DEMOCA...] or SYS$SYSDEVICE:[VMS$COMMON.SSL3.DEMOCA...], copy the certificate store to SYS$SYSDEVICE:[VMS$COMMON.SSL31.DEMOCA...].

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$LOGIN. For example, the UNIX pathname ~/.openssl/profile/prefs.js is equivalent to the OpenVMS directory [._OPENSSL.PROFILE]PREFS.JS if DEFAULT is SYS$LOGIN.

Installing VSI SSL31

Install the VSI SSL31 V3.1 for OpenVMS kit by entering the following command:

```
$ PRODUCT INSTALL SSL31
```

Whereupon you should observe output similar to the following:

```
Performing product kit validation of signed kits ...
%PCSI-I-VSIVALPASSED, validation of VSI-AXPVMS-SSL31-V0301-4-1.PCSI$COMPRESSED;1 succeeded

The following product has been selected:
    VSI AXPVMS SSL31 V3.1-4                 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 AXPVMS SSL31 V3.1-4: SSL31 for OpenVMS AXP V3.1-4 (Based on OpenSSL 3.1.4)

    Copyright 2024 VMS Software, Inc.

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

Do you want to review the options? [NO] yes

VSI AXPVMS SSL31 V3.1-4: SSL31 for OpenVMS AXP V3.1-4 (Based on OpenSSL 3.1.4)
    Run the installation verification procedure (IVP)?: YES

Are you satisfied with these options? [YES]

Execution phase starting ...

The following product will be installed to destination:
    VSI AXPVMS SSL31 V3.1-4                 DISK$AXPSYS:[VMS$COMMON.]

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

The following product has been installed:
    VSI AXPVMS SSL31 V3.1-4                 Layered Product

%PCSI-I-IVPEXECUTE, executing test procedure for VSI AXPVMS SSL31 V3.1-4 ...
%PCSI-I-IVPSUCCESS, test procedure completed successfully

VSI AXPVMS SSL31 V3.1-4: SSL31 for OpenVMS AXP V3.1-4 (Based on OpenSSL 3.1.4)

    Insert the following lines in SYS$MANAGER:SYSTARTUP_VMS.COM:
        @SYS$STARTUP:SSL31$STARTUP.COM
    Insert the following lines in SYS$MANAGER:SYSHUTDWN.COM:
        @SYS$STARTUP:SSL31$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:SSL31-4-AXP.RELEASE_NOTES for more information.

    It is recommended to run extended IVP tests as a post-installation step.


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 SSL31 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 SSL31 startup, shutdown, and logical name creation files are executed
```

Add SSL31$STARTUP.COM to SYS$MANAGER:SYSTARTUP_VMS.COM to define SSL31$ logical names and install shareable images. If there is already a SSL$STARTUP.COM, SSL1$STARTUP.COM, SSL111$STRATUP.COM and/or SSL3$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
$ if f$search("sys$startup:ssl3$startup.com") .nes. ""
$ then
$    @sys$startup:ssl3$startup.com
$ endif
$ if f$search("sys$startup:ssl31$startup.com") .nes. ""
$ then
$    @sys$startup:ssl31$startup.com
$ endif
```

The SSL31$STARTUP.COM, SSL3$STARTUP.COM SSL111$STARTUP.COM, SSL1$STARTUP.COM and SSL$STARTUP.COM startup command procedures in the above example will automatically define the SSL31$, SSL3$, SSL111$, SSL1, and SSL$ executive-mode logical names in the SYSTEM logical name table and will install into memory the SSL31, SSL3, SSL111, SSL1 and SSL 1.4 shareable images that reside in the [SYSLIB] directory.

Ensure that the SSL31$STARTUP.COM command procedure is invoked after invoking SSL$STARTUP.COM, SSL1$STARTUP.COM, SSL111$STARTUP.COM or SSL3$STARTUP.COM. The command procedures define a common logical "OPENSSL" that points to the include (header) file directory used when building applications using OpenSSL. Invoking SSL31$STARTUP.COM last ensures that the logical is defined to correctly point to the latest VSI SSL31 3.1 header files.

Also, add SSL31$SHUTDOWN.COM to SYS$MANAGER:SYSHUTDWN.COM to remove installed images and deassign the SSL31$ logical names at the system shutdown. If there is a SSL$SHUTDOWN.COM, SSL1$SHUTDOWN.COM, SSL111$SHUTDOWN.COM and/or SSL3$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
$ if f$search("sys$startup:ssl3$shutdown.com") .nes. ""
$ then
```

```
$    @sys$startup:ssl3$shutdown.com
$ endif
$ if f$search("sys$startup:ssl31$shutdown.com") .nes. ""
$ then
$    @sys$startup:ssl31$shutdown.com
$ endif
```

Please refer to "Logical names" under the section "Coexistence and major changes between VSI SSL V1.4, VSI SSL1, VSI SSL111, VSI SSL3 and VSI SSL31" 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:

```
$ @SSL31$COM:SSL31$UTILS
```

- Updated VSI SSL31 files requiring attention

Systems with custom changes to SSL, SSL1, SSL111 or SSL3 command procedures may need to replicate those changes in the SSL31 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, SSL1$COM:SSL1$SYSTARTUP.COM, SSL111$COM:SSL111$SYSTARTUP.COM or SSL3$COM:SSL3$SYSTARTUP.COM to SSL31$COM:ssl31$SYSTARTUP.COM

- If SYS$STARTUP:SSL$STARTUP.COM, SYS$STARTUP:SSL1$STARTUP.COM, SYS$STARTUP:SSL111$STARTUP.COM or SYS$STARTUP:SSL3$STARTUP.COM have any manual changes, ensure that these changes are copied to the site-specific startup command procedure SSL31$COM:SSL31$SYSTARTUP.COM. This command procedure will be invoked by SYS$STARTUP:SSL31$STARTUP.COM.

- Copy any manual changes done to the site-specific shutdown command procedures SSL$COM:SSL$SYSHUTDOWN.COM, SSL1$COM:SSL1$SYSHUTDOWN.COM, SSL111$COM:SSL111$SYSHUTDOWN.COM or SSL3$COM:SSL3$SYSHUTDOWN.COM to SSL31$COM:SSL31$SYSHUTDOWN.COM.

- If SYS$STARTUP:SSL$SHUTDOWN.COM, SYS$STARTUP:SSL1$SHUTDOWN.COM, SYS$STARTUP:SSL111$SHUTDOWN.COM or SYS$STARTUP:SSL3$SHUTDOWN.COM have any manual changes, ensure that these changes are copied to the site-specific shutdown command procedure SSL31$COM:SSL31$SYSHUTDOWN.COM. This command procedure will be invoked by SYS$STARTUP:SSL31$SHUTDOWN.COM.

- Copy any manual changes done to the OpenSSL configuration files SSL$ROOT:[000000]OPENSSL.CNF, SSL1$ROOT:[000000]OPENSSL.CNF, SSL111$ROOT:[000000]OPENSSL.CNF or SSL3$ROOT:[000000]OPENSSL.CNF to SSL31$ROOT:[000000]OPENSSL.CNF.

- Copy any manual changes done to the OpenSSL configuration files SSL$ROOT:[000000]OPENSSL-VMS.CNF, SSL1$ROOT:[000000]OPENSSL-VMS.CNF, SSL111$ROOT:[000000]OPENSSL-VMS.CNF or SSL3$ROOT:[000000]OPENSSL-VMS.CNF to SSL31$ROOT:[000000]OPENSSL-VMS.CNF.

- If any other of *.CNF files from previous releases are intended to be used with VSI SSL31 V3.1, insert ".pragma dollarid:on" statement as the first line in order to make the '$' sign without '{}' treated as a usual character (not as substitution template) in VMS paths.

- Migrate any SSL certificates store content to VSI SSL31 V3.1 by following the steps highlighted under "Migrate certificate store from VSI SSL V1.4, VSI SSL1, VSI SSL111 V1.1 or VSI SSL3 V3.0 to VSI SSL31 V3.1".

- Migrate any applications built with VSI SSL V1.4, VSI SSL1, VSI SSL111 V1.1 and/or VSI SSL3 V3.0 to VSI SSL31 V3.1 by rebuilding and relinking the application with the VSI SSL31 V3.1 header files and

libraries.

- Migrate any command procedures using VSI SSL V1.4, VSI SSL1, VSI SSL111 V1.1 and/or VSI SSL3 V3.0 directories, command procedures or logicals to point to VSI SSL31 V3.1 directories, command procedures or logicals. See "Coexistence and major changes between VSI SSL V1.4, VSI SSL1, VSI SSL111, VSI SSL3 V3.0 and VSI SSL31 V3.1" section of this document for more information.

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

```
$ @SYS$TEST:SSL31$IVP.COM
```

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

```
$ @SSL31$COM:SSL31$CERT_TOOL
```

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

VSI SSL31 directory structure

The VSI SSL31 directory structure is as follows:

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

```
[SSL31] - Top-level directory created by default in SYS$SYSDEVICE:[VMS$COMMON].
[SSL31.ALPHA_EXE]       - Contains images for the Alpha server platform.
[SSL31.COM]             - Contains command procedures.
[SSL31.DEMOCA]          - Contains demos for SSL's CA features
[SSL31.DEMOCA.CERTS]    - Contains certificates and keys.
[SSL31.DEMOCA.CONF]     - Contains configuration files.
[SSL31.DEMOCA.CRL]      - Contains revoked certificates and CRLs.
[SSL31.DEMOCA.PRIVATE]  - Contains private keys and random data.
[SSL31.DOC]             - OpenSSL Group-provided documentation and information.
[SSL31.INCLUDE]         - Contains C header (.H) files.
[SSL31.LIB]             - Contains static libraries (.OLB) files.
[SSL31.MODULES]         - Contains dynamically loadable OpenSSL
                          modules (e.g. providers).
[SYS$STARTUP]          - Contains startup and shutdown templates and files.
[SYSHLP]               - Contains release notes.
[SYSHLP.EXAMPLES.SSL31] - Contains SSL crypto and secure session examples.
[SYSLIB]               - Contains SSL shareable image files.
[SYSTEST]              - Contains SSL31$IVP.COM test file.
```

Note that the VSI SSL31 example programs are located in SYS$COMMON:[SYSHLP.EXAMPLES.SSL31]. The logical name SSL31$EXAMPLES points to this directory.

Building a VSI SSL31 application

VSI SSL31 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$SHARE:SSL31$LIBSSL_SHR.EXE      - 64-bit SSL APIs
SYS$SHARE:SSL31$LIBCRYPTO_SHR.EXE   - 64-bit Crypto APIs
SYS$SHARE:SSL31$LIBSSL_SHR32.EXE    - 32-bit SSL APIs
SYS$SHARE:SSL31$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$SHARE:SSL31$LIBSSL_SHR/SHARE
SYS$SHARE:SSL31$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$SHARE:SSL31$LIBSSL_SHR32/SHARE
SYS$SHARE:SSL31$LIBCRYPTO_SHR32/SHARE
```

Release Notes

This section contains notes on the current release of VSI SSL31 for OpenVMS.

The no-md2, no-mdc2, no-idea, no-rc5 and no-asm options were used during
configuration phase of VSI SSL31 V3.1 building.

Fixed a memory leak issue when the C standard APIs are used instead of
OpenSSL memory allocator APIs.

The base and legacy providers are included into the kit only.

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 SSL31
complies with all national and international laws that apply to you.

VSI SSL31 APIs are not backward compatible

VSI SSL31 V3.1 for OpenVMS is based on the 3.1.4 baselevel of
OpenSSL. Some of the OpenSSL API, data structures, and commands have changed
from the previous VSI SSL V1.4, VSI SSL1, VSI SSL111 V1.1 and VSI SSL3 V3.0
product versions.

VSI cannot guarantee the backward compatibility of VSI SSL31 V3.1 with VSI SSL V1.4, VSI SSL1, VSI SSL111 V1.1 or VSI SSL3 V3.0.

Applications will have to be recompiled and re-linked in order to make use of the latest VSI SSL31 V3.1 header files and shareable images.

Note that the VSI SSL31 shareable images names are different from VSI SSL 1.4, VSI SSL1, VSI SSL111 V1.1 or VSI SSL3 V3.0. Refer to the "Coexistence and major changes between VSI SSL V1.4, VSI SSL1, VSI SSL111, VSI SSL3 and VSI SSL31" for details.

Preserve configuration files before manually uninstalling VSI SSL31

Preserving configuration files is not necessary when you perform a regular upgrade or reinstallation of VSI SSL31 using the PRODUCT INSTALL command.

However, if you intend to uninstall VSI SSL31 and wish to preserve any modifications to the VSI SSL31 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 SSL31 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 SSL31, move the saved items back into the VSI SSL31 directory structure.

Configuration command procedure template files

The configuration files included in the VSI SSL31 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 SSL31 for OpenVMS.

If you are upgrading from a previous version of VSI SSL31, after you install the VSI SSL31 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 SSL31 for OpenVMS kit, both the .CNF_TEMPLATE and .CNF files are provided.

VSI SSL31 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 SSL31 and install it as a layered product, it must be installed on the system disk.

Shutdown VSI SSL31 before installing it on the common system disk

Before installing VSI SSL31 to a common system disk in a cluster, you must first shutdown VSI SSL31 by entering the following command on each node in the cluster:

```
$ @SYS$STARTUP:SSL31$SHUTDOWN
```

Shutting down VSI SSL31 deassigns logical names and removes installed shareable images that may interfere with the installation.

After the installation is complete, start VSI SSL31 by entering the following command on each node in the cluster:

```
$ @SYS$STARTUP:SSL31$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$STARTUP]SSL31$SHUTDOWN
$ @device:[directory.SYS$STARTUP]SSL31$STARTUP


OpenSSL version command displays VSI SSL31 for OpenVMS version
--------------------------------------------------------------
The OpenSSL command line utility command VERSION includes the
VSI SSL31 for OpenVMS version. The OpenSSL VERSION command displays
output similar to the following:

OpenSSL> version
OpenSSL 3.1.4 xx xxx xxxx (Library: OpenSSL 3.1.4 xx xxx xxxx)
SSL31 for OpenVMS V3.1(xx) xxx xx xxxx (Library: SSL31 for OpenVMS V3.1(xx) xxx xx 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 SSL31 for OpenVMS only accepts the ${var} format.
Additionally, *.CNF files must contain .pragma dollarid:on, which allows using
of the dollar sign in variable names.


IDEA, RC5 and MDC2 symmetric cipher algorithms are 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/. The API and command line documentation
shipped with this kit will likely 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 --