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

September 2023

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 1.1-1W for OpenVMS.

The information in this file applies to VSI SSL111 running on OpenVMS x86-64 servers.

VSI SSL111 Version 1.1-1W for OpenVMS is based on Open Source OpenSSL version 1.1.1W 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 V1.1-1W for OpenVMS requires the following software.

- Operating System

  VSI OpenVMS x86-64 server Version V9.2-1 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 1.1-1W for OpenVMS

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

Logical names:

All the logical names associated with VSI SSL111 V1.1 are prefixed with SSL111$. The following is a system-level logical names that are defined for VSI SSL111 V1.1:

VSI SSL111 V1.1-1W 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.SSL111]"
"SSL111$EXE" = "SSL111$ROOT:[X86_EXE]"
"SSL111$INCLUDE" = "SSL111$ROOT:[INCLUDE]"
"SSL111$KEY" = "SSL111$ROOT:[DEMOCA.CERTS]"
"SSL111$KEYS" = "SSL111$ROOT:[DEMOCA.CERTS]"
"SSL111$PRIVATE" = "SSL111$ROOT:[DEMOCA.PRIVATE]"
"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 (SSL111$INCLUDE:).

Directory names:

The top level directory structure for VSI SSL111 V1.1 is SYS$SYSDEVICE:[VMS$COMMON.SSL111].

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

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
```

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
```

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 ...
%PCSI-I-VSIVALPASSED, validation of VSI-X86VMS-SSL111-V0101-1W-1.PCSI$COMPRESSED;1 succeeded

The following product has been selected:
    VSI X86VMS SSL111 V1.1-1W              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 X86VMS SSL111 V1.1-1W: SSL111 for OpenVMS X86-64 V1.1-1W (Based on OpenSSL
1.1.1W)

    Copyright 2023 VMS Software, Inc.

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

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

VSI X86VMS SSL111 V1.1-1W: SSL111 for OpenVMS X86-64 V1.1-1W (Based on OpenSSL 1.1.1W)
    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 X86VMS SSL111 V1.1-1W              DISK$X86SYS:[VMS$COMMON.]

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

The following product has been installed:
    VSI X86VMS SSL111 V1.1-1W              Layered Product

%PCSI-I-IVPEXECUTE, executing test procedure for VSI X86VMS SSL111 V1.1-1W ...
%PCSI-I-IVPSUCCESS, test procedure completed successfully

VSI X86VMS SSL111 V1.1-1W: SSL111 for OpenVMS X86 V1.1-1W (Based on OpenSSL 1.1.1W)

    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:SSL111-W-X86.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.

  Also, add SSL111$SHUTDOWN.COM to SYS$MANAGER:SYSHUTDWN.COM to remove installed images and deassign the SSL111$ logical names at system shutdown.

- 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
  ```

- 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.X86_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 no-md2, no-mdc2, no-idea, no-rc5 and no-asm options were used during configuration phase of VSI SSL111 V1.1 building.

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.

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 1.1.1W xx xxx xxxx
SSL111 for OpenVMS V1.1(1W)  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 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 --