



VMS Software

VSI VMS/XDE

Installation and Configuration Guide

Publication Date: July 2026

Software Version: VMS/XDE V2026.07

Table of Contents

1. Introduction	3
2. System Requirements	3
3. Licensing	3
3.1. Preparing Your License for VMS/XDE	3
4. Using VMS/XDE	4
4.1. Installing VMS/XDE	4
4.1.1. Registering License After Installation	5
4.2. Starting VMS/XDE	5
4.3. Shutting Down VMS/XDE	6
4.4. Storage Emulation Directories in This Package	6
Appendix A. Optional Configuration Steps and Reference Information	7
A.1. Understanding Storage Emulation Directories	7
A.1.1. Creating Storage Emulation Directories	7
A.2. Applications Included in the Current Release	7
A.3. Configuring DECwindows	9
A.4. SYSGEN Limitations	9
A.5. Optional Parameters for ./run.sh	10

1. Introduction

VMS Cross-Development Environment (VMS/XDE) is a set of tools for developing OpenVMS x86-64 applications on Linux operating systems. For a large part based on OpenVMS itself, it offers an accurate OpenVMS-like user environment suitable for OpenVMS development and debugging, using the exact same toolchain (compiler, librarian, linker, debugger) that is available on OpenVMS x86-64. Running on Linux, it also provides access to Linux-based tools, like editors and source-control.

For information about new features, bug fixes, and known issues in this release of VMS/XDE, refer to the [Release Notes](#).

2. System Requirements

- **Host operating system:** An x86-64 Linux distribution with a minimum kernel version of 5.9
- **CPU:** Intel Skylake-based (6xxx) or AMD 1st generation Zen-based (Ryzen-1xxx)
- **RAM:** 8 GB
- **Libraries:** GLIBC 2.28 and GLIBCXX 3.4.26
- **File system:** A file system that supports extended attributes (such as ext3 or ext4)

Warning

VSI does not guarantee that VMS/XDE will function correctly when installed directly on NTFS-based partitions or if NTFS-based partitions are used for the DKA0 or DKA1 storage devices.¹

3. Licensing

VMS/XDE requires an OpenVMS license. Additionally, some of the components included in VMS/XDE (such as compilers and certain layered products) require their own separate licenses. To purchase a license or request an evaluation license, visit <https://vmssoftware.com/products/licenses/> or contact VSI Sales at <sales@vmssoftware.com>.

3.1. Preparing Your License for VMS/XDE

VMS/XDE accepts licenses only in the form of command procedure files (.COM). Once you have obtained your license, create a command procedure file (e.g., LICENSE.COM) on the host system, copy the text of your license into the file, and add the following line at the end:

```
$ LICENSE LOAD
```

Your license file should now look similar to this:

```
$ LICENSE REGISTER license-name -  
/ISSUER=your-value -  
/AUTHORIZATION=your-value -  
/PRODUCER=your-value -  
/UNITS=your-value -  
/TERMINATION_DATE=your-value -
```

¹For more information on storage emulation directories, see *Section A.1, "Understanding Storage Emulation Directories"*.

```
/OPTIONS=your-value -  
/TOKEN=your-value -  
/CHECKSUM=your-value -  
$ LICENSE LOAD
```

To update an existing license, add these two lines at the beginning of the license file:

```
$ LICENSE UNLOAD license-name  
$ LICENSE DELETE license-name
```

Important Considerations

- If you create your .COM license file on a non-Linux operating system, make sure *not* to use the carriage return symbol at the ends of lines. In other words, all lines must end in LF, never CR LF.
 - The .COM license file must not call any other command procedure files.
-

4. Using VMS/XDE

Before you start using VMS/XDE, familiarize yourself with the current limitations of the product. Refer to *Section 3, "Known Issues and Limitations"* of the [Release Notes](#).

4.1. Installing VMS/XDE

Warning

VMS/XDE can only be installed interactively (i.e. the installation procedure requires user input). For this reason, VMS/XDE installation cannot be automated as part of a pipeline, script, etc.

To install and configure VMS/XDE, perform the following steps:

1. Make sure that your current host system user is *not* a root user or a member of the root group. Such users cannot install or run VMS/XDE.
2. VMS/XDE is provided to users in the form of a self-extracting archive. Use the following commands to unpack the archive:

```
$ chmod +x ./archive-name  
$ ./archive-name
```

3. You will be prompted for a license. If you have already prepared a .COM license file as described in *Section 3.1, "Preparing Your License for VMS/XDE"*, answer **yes** and, when prompted, specify the full path² to your .COM license file. When the license installation procedure finishes, you will be prompted to review a log file. VSI recommends that you do.

If you do not have a license yet or would like to register it later, answer **no** to continue installation.

During installation, the VMS/XDE archive will be extracted into the directory called VMS-XDE in the current host system directory, and an OpenVMS user will automatically be created within VMS/XDE. The name of that OpenVMS user will be the same as that of the current host system user, with the following modifications:

²Do *not* use the tilde character (~) in this path.

- Names longer than 12 characters will be truncated.
- Any special characters will be removed.

The resulting OpenVMS user name will be displayed on your screen during the installation.

4.1.1. Registering License After Installation

If you have not provided a valid OpenVMS license during installation, you must do so now. Otherwise, skip to the next section.

Prepare a .COM file as described in *Section 3.1, "Preparing Your License for VMS/XDE"* and enter the following command:

```
$ ./run.sh --license=path-to-your-COM-file
```

When the license installation procedure finishes, you will be prompted to review a log file. VSI recommends that you do.

Warning

If you choose not to include the **LICENSE LOAD** command in your .COM file, you must restart VMS/XDE after running the command above. To do so, run the following commands:

```
$ ./shutdown.sh  
$ ./run.sh
```

4.2. Starting VMS/XDE

To run VMS/XDE, run the following command from the VMS/XDE installation directory:

```
$ ./run.sh
```

You will be prompted for a username. Enter the name of the user that was created automatically during installation³. No password is required.

Usage Restrictions

When using VMS/XDE, do not alter the TCP/IP configuration, do not change the OpenVMS timezone settings, and do not use the SYSMAN utility. For more information, refer to *Section 3, "Known Issues and Limitations"* of the [Release Notes](#).

³If you are planning to create more VMS/XDE users, note that the name, user ID, and group ID of the current VMS/XDE user must match those of the current host system user.

4.3. Shutting Down VMS/XDE

When you exit a VMS/XDE interactive session, some of OpenVMS processes will keep running. To gracefully shut down VMS/XDE and all OpenVMS processes, first log out of OpenVMS via the command **LOGOUT**, then enter the following command:

```
$ ./shutdown.sh
```

4.4. Storage Emulation Directories in This Package

The following storage emulation directories⁴ are included in this package and guarantee that VMS/XDE functions correctly:

Directory	Purpose
vms_os/V9.2/DKA0	Emulates the system storage device.
vms_os/V9.2/DKA1	Emulates a device that contains the OpenVMS user home directory.
vms_os/V9.2/DKC0	Emulate devices that do not support the OpenVMS file versioning feature.
vms_os/V9.2/DKC1	
vms_os/V9.2/LDM1	Emulates an OpenVMS memory disk. Do <i>not</i> change anything about this directory, otherwise VMS/XDE may not function correctly.

⁴For more information on storage emulation directories, see *Section A.1, "Understanding Storage Emulation Directories"*.

A. Optional Configuration Steps and Reference Information

A.1. Understanding Storage Emulation Directories

Storage emulation directories are regular host system directories that VMS/XDE uses to emulate OpenVMS storage devices. All data within VMS/XDE is stored on these emulated devices. VMS/XDE supports two types of devices:

Devices that support the OpenVMS file versioning feature.

The directories used for emulating these devices, and everything stored in them, must follow OpenVMS naming conventions and include the version number suffix ;**n**. For more information on OpenVMS file versions, refer to [VSI OpenVMS User's Manual](#).

Devices that do not support the OpenVMS file versioning feature.

The names of these devices always start with the string DKC. The directories used for emulating these devices, and everything stored in them, must follow the host operating system naming conventions.

A.1.1. Creating Storage Emulation Directories

New storage emulation directories must be created in the directory `~/VMS-XDE/vms_os/V9.2`. Before creating a storage emulation directory, make sure to gracefully shut down VMS/XDE as described in *Section 4.3, "Shutting Down VMS/XDE"*. Mounting emulated OpenVMS devices while VMS/XDE is running is currently not possible because the MOUNT utility is not supported.

The names of all storage emulation directories must follow the OpenVMS device naming conventions:

- A directory name must only contain uppercase characters.
- A directory name must begin with the prefix DK, LD, MD, DG, DQ, or DS.
- The initial two-letter prefix must be followed by a single alphabetic character.

Important

Devices named DKC*n* will not support the OpenVMS file versioning feature.

- The first three alphabetic symbols must be followed by a numeric postfix from 0 to 65535.

The names of *all* storage emulation directories, subdirectories, and files stored in them, must use *only* the OpenVMS-supported symbols (ASCII, national code-pages are currently not supported).

A.2. Applications Included in the Current Release

The current version of VMS/XDE includes the following OpenVMS applications:

- Base VSI OpenVMS x86-64 V9.2-3 operating system environment
- The following compilers:

- BASIC
- BLISS
- C
- COBOL
- C++
- Fortran
- LUA
- MACRO-32
- Pascal
- Python
- x86asm
- Linker utility
- Librarian utility
- SDL translator utility
- System Analyzer utility
- Run-time libraries for included compilers and other system libraries, such as C RTL, LIB\$RTL, etc.
- The following layered products:
 - DECset
 - DECforms
 - DECwindows
 - curl
 - Git
 - GNV utilities set
 - LDAP
 - OpenSSH
 - SSL111
 - SSL3
 - SSL31
 - VDD

- VMSI18N

A.3. Configuring DECwindows

Caution

DECwindows support is still under way. VSI tests were performed in a Windows environment using WSL2 and Ubuntu 20.04 with open-source VcXsrv used as the X server.

Using the DECwindows application with VMS/XDE requires additional setup to function properly.

1. Configure an external X server that works via TCP/IP and correctly handles the IPv4 address 127.0.0.1 (localhost) on port 6000.
2. Make sure that VMS/XDE is *not* running, then start the X server.
3. Run VMS/XDE and execute the following DCL commands (or add them to your LOGIN.COM file):

```
$ SET PROCESS/PRIVILEGES=CMEXEC
$ SET DISPLAY/CREATE/TRANSPORT=TCPIP/NODE=127.0.0.1/SERVER=0
$ DEFINE DECW$IGNORE_WORKSTATION "TRUE"
$ @SYS$STARTUP:DECW$STARTUP.COM
```

Note

If you choose to add the commands to your LOGIN.COM, make sure to execute the file after editing it.

A.4. SYSGEN Limitations

When using the SYSGEN utility in VMS/XDE, keep the following limitations in mind:

- Do not change the values of the parameters VCC_FLAGS, MULTITHREAD, MAXPROCESSCNT. Doing so will result in a crash.
- Do not set the values of the parameters listed below lower than the default. Doing so may result in unexpected behavior.

Parameter	Default Value
MAXBUF	32768
DEFMBXBUFQUO	32768
DEFMBXMXMSG	8192

To restore all SYSGEN values to default, run the following command:

```
$ ./run.sh --default-sysgen
```

Starting from VMS/XDE V2026.07, running this command will not replace your X86_64VMSSYS.PAR but instead load the default configuration values from X86_64VMSSYS_DEF.PAR.

A.5. Optional Parameters for ./run.sh

The command `./run.sh` accepts the following optional parameters:

Optional Parameter	Description
<code>--version</code> <code>-v</code>	Show current version.
<code>--help</code> <code>-h</code>	Show help.
<code>--shutdown</code> <code>-s</code>	Gracefully shut down VMS/XDE and all OpenVMS processes.
<code>--input=file-name</code>	Full path to the DCL script that will be executed when VMS/XDE starts, for example <code>--input=DKC0:[MY-DISK]INPUT.COM</code> .
<code>--output=file-name</code>	Full path to the file where you want to write the VMS/XDE output, for example <code>--output=DKC0:[MY-DISK]OUTPUT.LIS</code> .
<code>--error=file-name</code>	Full path to the file where you want to write the error log, for example <code>--error=DKC0:[MY-DISK]ERROR.LIS</code> .
<code>--fgsync</code>	Run in non-interactive mode and wait until finished. This parameter can be used only if the <code>--input</code> and <code>--output</code> parameters are set.
<code>--license=file-name</code>	Path to the license .COM file. Use this parameter to register and load your license after the installation, if you chose not to do that during the installation.
<code>--default-sysgen</code>	Restore the file X86_64VMSSYS.PAR to the default state.