

# VSI OpenVMS

## VSI Pascal for OpenVMS Installation Guide

Document Number: DO-DPASIG-01A

Publication Date: March 2022

This guide explains how to install the VSI Pascal compiler and related software components on the VSI OpenVMS operating system.

**Revision Update Information:** This is a new manual.

**Operating System and Version:** VSI OpenVMS I64 Version 8.4-1H1  
VSI OpenVMS Alpha Version 8.4-2L1

**Software Version:** VSI Pascal Version 6.2 for OpenVMS I64  
VSI Pascal Version 6.2-125 for OpenVMS Alpha

---

# VSI Pascal for OpenVMS Installation Guide



VMS Software

---

Copyright © 2022 VMS Software, Inc. (VSI), Burlington, Massachusetts, USA

## Legal Notice

Confidential computer software. Valid license from VSI required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for VSI products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. VSI shall not be liable for technical or editorial errors or omissions contained herein.

HPE, HPE Integrity, HPE Alpha, and HPE Proliant are trademarks or registered trademarks of Hewlett Packard Enterprise.

Intel, Itanium and IA64 are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java, the coffee cup logo, and all Java based marks are trademarks or registered trademarks of Oracle Corporation in the United States or other countries.

Kerberos is a trademark of the Massachusetts Institute of Technology.

Microsoft, Windows, Windows-NT and Microsoft XP are U.S. registered trademarks of Microsoft Corporation. Microsoft Vista is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries.

Motif is a registered trademark of The Open Group

UNIX is a registered trademark of The Open Group.

<b>Preface .....</b>	<b>v</b>
1. About VSI .....	v
2. Intended Audience .....	v
3. Document Structure .....	v
4. Related Documents .....	v
5. OpenVMS Documentation .....	v
6. VSI Encourages Your Comments .....	vi
7. Conventions .....	vi
<b>Chapter 1. Preparing to Install VSI Pascal .....</b>	<b>1</b>
1.1. Required Operating System Components .....	1
1.2. License Registration .....	1
1.3. Multiple Version Support .....	2
1.4. Installation Procedure Requirements .....	2
1.4.1. Installation Account Privileges and Disk Space .....	2
1.4.2. System Parameters .....	3
1.4.2.1. Calculating the Values for GBLPAGES and GBLSECTIONS .....	4
1.4.2.2. Changing System Parameter Values with AUTOGEN .....	4
1.4.3. Backing Up Your System Disk .....	5
1.5. Stopping the Installation .....	5
<b>Chapter 2. Installing VSI Pascal .....</b>	<b>7</b>
2.1. Installation Procedure .....	7
2.2. Error Recovery .....	13
<b>Chapter 3. After Installing VSI Pascal .....</b>	<b>15</b>
3.1. Running the Installation Verification Procedure (IVP) Separately .....	15
3.2. User Account Requirements .....	15
3.3. Editing System Files .....	15
3.4. Enhancing VSI Pascal Performance .....	16
3.4.1. Using Multiple Environment Files (OpenVMS VAX Systems Only) .....	16
3.4.2. Installing VSI Pascal as a Shared Image .....	16
3.4.3. Improving VSI Pascal Compilation Performance on OpenVMS I64 and OpenVMS Alpha Systems .....	16
3.4.3.1. Required Virtual Memory .....	17
3.4.3.2. Required Physical Memory .....	18
3.4.3.3. Effects of Physical Memory .....	19



# Preface

This guide describes how to install VSI Pascal on the OpenVMS operating system.

Keep this guide with your distribution kit. You will need it to install maintenance updates or to reinstall VSI Pascal for any other reason.

## 1. About VSI

VMS Software, Inc. (VSI) is an independent software company licensed by Hewlett Packard Enterprise to develop and support the OpenVMS operating system.

VSI seeks to continue the legendary development prowess and customer-first priorities that are so closely associated with the OpenVMS operating system and its original author, Digital Equipment Corporation.

## 2. Intended Audience

This guide is intended for the system manager who installs VSI Pascal for OpenVMS.

## 3. Document Structure

This guide contains the following chapters:

- Chapter 1 describes the hardware and software requirements for VSI Pascal and related procedures that you complete before the installation.
- Chapter 2 describes the VSI Pascal installation procedure.
- Chapter 3 describes the VSI Pascal Installation Verification Procedure (IVP) and any postinstallation procedures you must complete to use VSI Pascal.

## 4. Related Documents

In addition to this guide, the VSI Pascal documentation set includes the following books:

- *VSI Pascal for OpenVMS User Manual*
- *VSI Pascal for OpenVMS Language Reference Manual*

For general information on OpenVMS system management, see the OpenVMS operating system documentation set.

For help understanding OpenVMS system error messages, see the *OpenVMS System Messages and Recovery Procedures Reference Manual* or the online HELP/MESSAGE facility.

## 5. OpenVMS Documentation

The full VSI OpenVMS documentation set can be found on the VMS Software Documentation webpage at <https://docs.vmssoftware.com>

## 6. VSI Encourages Your Comments

You may send comments or suggestions regarding this manual or any VSI document by sending electronic mail to the following Internet address: <docinfo@vmssoftware.com>. Users who have OpenVMS support contracts through VSI can contact <support@vmssoftware.com> for help with this product.

## 7. Conventions

The following product names may appear in this manual:

- OpenVMS Industry Standard 64 for Integrity Servers
- OpenVMS I64
- I64

All three names — the longer form and the two abbreviated forms — refer to the version of the OpenVMS operating system that runs on the Intel® Itanium® architecture.

The following typographic conventions may be used in this manual:

Convention	Meaning
Ctrl/ <i>x</i>	A sequence such as Ctrl/ <i>x</i> indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.
PF1 <i>x</i>	A sequence such as PF1 <i>x</i> indicates that you must first press and release the key labeled PF1 and then press and release another key or a pointing device button.
. . .	A horizontal ellipsis in examples indicates one of the following possibilities: <ul style="list-style-type: none"> <li>• Additional optional arguments in a statement have been omitted.</li> <li>• The preceding item or items can be repeated one or more times.</li> </ul> <p style="text-align: center;">Additional parameters, values, or other information can be entered.</p>
. . . .	A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.
( )	In command format descriptions, parentheses indicate that you must enclose choices in parentheses if you specify more than one.
[ ]	In command format descriptions, brackets indicate optional choices. You can choose one or more items or no items. Do not type the brackets on the command line. However, you must include the brackets in the syntax for OpenVMS directory specifications and for a substring specification in an assignment statement.
	In command format descriptions, vertical bars separate choices within brackets or braces. Within brackets, the choices are optional; within braces, at least one choice is required. Do not type the vertical bars on the command line.

---

Convention	Meaning
{ }	In command format descriptions, braces indicate required choices; you must choose at least one of the items listed. Do not type the braces on the command line.
<b>bold type</b>	Bold type represents the name of an argument, an attribute, or a reason.
<i>italic type</i>	Italic type indicates important information, complete titles of manuals, or variables. Variables include information that varies in system output (Internal error <i>number</i> ), in command lines ( <i>/PRODUCER=name</i> ), and in command parameters in text (where <i>dd</i> represents the predefined code for the device type).
UPPERCASE TYPE	Uppercase type indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege.
–	A hyphen at the end of a command format description, command line, or code line indicates that the command or statement continues on the following line.
numbers	All numbers in text are assumed to be decimal unless otherwise noted. Nondecimal radixes – binary, octal, or hexadecimal – are explicitly indicated.





# Chapter 1. Preparing to Install VSI Pascal

This chapter discusses the preparations and requirements necessary for installing VSI Pascal.

Your bill of materials (BOM) and indented bills report (BIL) specify the number and contents of your media. Be sure to verify the contents of your kit with this information. If your kit is damaged or if you find that parts of it are missing, contact your VSI representative.

VSI Pascal provides online release notes, which are copied to the following file during installation:

```
SYS$HELP : PASCAL05n . RELEASE_NOTES
```

VSI recommends that you read the release notes before proceeding with the installation. For information on accessing the online release notes, see Section 2.1.

Your distribution kit includes a “Read First” cover letter. Please read this cover letter for information that you should know before installing VSI Pascal. This information may not be included in this installation guide or in the release notes.

## 1.1. Required Operating System Components

VSI Pascal requires:

- VSI OpenVMS I64 V8.4-1H1 or higher
- VSI OpenVMS Alpha V8.4-2L1 or higher
- OpenVMS VAX Version 5.5 or higher

Future VSI Pascal releases may require higher versions of the OpenVMS operating system, as described in the online release notes or the Read First cover letter for your operating system.

The OpenVMS operating system comes with a variety of support options or components.

## 1.2. License Registration

You must register the VSI Pascal license through the OpenVMS License Management Facility (LMF) in accordance with the license agreement signed by your site. The license registration information you need is contained in the Product Authorization Key (PAK) that is shipped with VSI Pascal if you ordered the license and media together; otherwise, the PAK is shipped separately to a location based on your license order. The PAK is a paper certificate that contains information about the license to run a particular piece of software.

If you are installing VSI Pascal as an update on a node or cluster that is licensed for this software, you have already completed the License PAK registration requirements.

During the installation, the system asks whether you have registered the VSI Pascal license and loaded the appropriate PAK. Before you start the installation, you must register and load your license for VSI Pascal to be able to run the Installation Verification Procedure (IVP) and for users to be able to run VSI Pascal.

To register a license, first log in to the system manager's account (username: SYSTEM). You then have a choice of two ways to perform the registration:

- Invoke the `SYSS$UPDATE:VMSLICENSE.COM` procedure. When it prompts you for information, respond with data from your License PAK.
- At the DCL prompt (`$`), enter the `LICENSE REGISTER` command with the appropriate qualifiers that correspond to License PAK information.

For information on using LMF, see the *VSI OpenVMS License Management Facility (LMF)*.

## 1.3. Multiple Version Support

VSI Pascal provides support for using multiple versions of the compiler on the same OpenVMS system.

During installation, you have the opportunity to save the current version of the compiler for later use. If chosen, the installation copies the current compiler before installing the new compiler.

The kit provides three command files to display and control which VSI Pascal compiler is used.

- `SYSS$SYSTEM:PASCAL$SHOW_VERSIONS.COM` is used to display all VSI Pascal compilers saved on the system and their version numbers. It also shows the current settings for process, job, group, and system-wide compiler settings. This command file has no parameters.
- `SYSS$SYSTEM:PASCAL$SET_VERSION.COM` is used to select a VSI Pascal compiler other than `SYSS$SYSTEM:PASCAL.EXE`. If invoked without parameters, it displays the currently available compilers and asks for a file name to use for the process-wide compiler setting. It has 2 optional parameters. The first parameter can be `PROCESS`, `JOB`, `GROUP`, or `SYSTEM`. This selects the level of visibility for the selected compiler. The second parameter can be a file name to use instead of being prompted for a file. This allows you to place a compiler selection in a `LOGIN.COM` or some other command file. All saved compilers are located in `SYSS$SYSTEM:`.
- `SYSS$SYSTEM:PASCAL$DEFAULT_VERSION.COM` is used to select the default VSI Pascal compiler when `SYSS$SYSTEM:PASCAL$SET_VERSION.COM` has not been used. The installation always provides a new `SYSS$SYSTEM:PASCAL.EXE` and copies existing compilers. You can use this command file to select one of the saved compilers to be `SYSS$SYSTEM:PASCAL.EXE` and rename the existing `SYSS$SYSTEM:PASCAL.EXE` for later use through `SYSS$SYSTEM:PASCAL$SET_VERSION.COM`.

## 1.4. Installation Procedure Requirements

The installation of VSI Pascal takes approximately 5 to 45 minutes, depending on your system configuration.

### 1.4.1. Installation Account Privileges and Disk Space

To install VSI Pascal, you must be logged in to an account that has `SETPRV` or at least the following privileges:

- `CMKRNL`
- `WORLD`
- `SYSPRV`

---

**Note**

VMSINSTAL turns off the BYPASS privilege at the start of the installation.

---

VSI Pascal requirements for free disk storage space are different during installation and after installation. Table 1.1, Table 1.2, and Table 1.3 summarize the storage requirements for the three operating systems.

**Table 1.1. Number of Blocks Required for OpenVMS I64 Systems**

Kit	During Installation	After Installation
VSI Pascal for OpenVMS I64 systems	45,000	40,500
VSI Pascal Run-Time Library for OpenVMS I64 systems <sup>1</sup>	1,100	1,100
VSI Pascal STARLET files	15,000	15,000
VSI Pascal examples	150	150

<sup>1</sup>Only when that kit is newer than the installed version.

**Table 1.2. Number of Blocks Required for OpenVMS Alpha Systems**

Kit	During Installation	After Installation
VSI Pascal for OpenVMS Alpha systems	25,000	21,000
VSI Pascal Run-Time Library for OpenVMS Alpha systems <sup>1</sup>	500	500
VSI Pascal STARLET files	15,000	15,000
VSI Pascal examples	150	150

<sup>1</sup>Only when that kit is newer than the installed version.

**Table 1.3. Number of Blocks Required for OpenVMS VAX Systems**

Kit	During Installation	After Installation
HP Pascal for OpenVMS VAX systems	3,500	3,000
HP Pascal STARLET files	11,500	10,000
HP Pascal examples	150	150

To determine the number of free disk blocks on the current system disk, enter the following DCL command:

```
$ SHOW DEVICE SYS$SYSDEVICE
```

## 1.4.2. System Parameters

Installing VSI Pascal requires certain system parameter settings.

The minimum number of free global pagelets (512-byte subpage unit) and global sections needed for the installation depends on whether SYS\$SYSTEM:PASCAL.EXE was previously installed as a shared image, as follows:

- If not previously installed as a shared image: one global section and a sufficient number of contiguous free global pagelets to contain the file SYS\$LIBRARY:DCLTABLES.EXE

- If previously installed as a shared image or will be subsequently installed with the PASCAL\$STARTUP command file (Section 3.4.2):
  - Global sections: six on OpenVMS I64 systems, three on OpenVMS Alpha systems, and two on OpenVMS VAX systems
  - A sufficient number of contiguous free global pagelets to contain the file SYS\$LIBRARY:DCLTABLES.EXE:
    - 32,000 additional global pagelets on OpenVMS I64 systems
    - 10,000 additional global pagelets on OpenVMS Alpha systems
    - 1,500 additional global pagelets on OpenVMS VAX systems

These values represent the number of free global pagelets and global sections required for the installation, not the total number you need to run your system and other software.

---

## Note

If you do not ensure that your system has the necessary global pagelets and global section SYSGEN parameters for the installation, the DCL tables can become corrupted in some situations.

---

### 1.4.2.1. Calculating the Values for GBLPAGES and GBLSECTIONS

To install and run VSI Pascal, you must have sufficient free contiguous global pagelets and global sections. To determine the number of global pagelets required by SYS\$LIBRARY:DCLTABLES.EXE, enter the following DCL command:

```
$ DIR/SIZE SYS$LIBRARY:DCLTABLES.EXE
```

This command returns the size (in blocks) of SYS\$LIBRARY:DCLTABLES.EXE. As an approximation, use this value as the number of global pagelets for the file.

To find the number of free contiguous global pagelets and free global sections, use the WRITE command with the F\$GETSYI lexical function. The following example shows how to get this information at your terminal (the default for SYS\$OUTPUT):

```
$ WRITE SYS$OUTPUT F$GETSYI ("CONTIG_GBLPAGES")
15848
$ WRITE SYS$OUTPUT F$GETSYI ("FREE_GBLSECTS")
24
```

If the number of free global pagelets or global sections is less than the required value (see Section 1.4.2), you must increase the system parameter setting. Section 1.4.2.2 describes the procedures for increasing these values using AUTOGEN.

### 1.4.2.2. Changing System Parameter Values with AUTOGEN

If you do not have enough free global pagelets or free global sections to install VSI Pascal, you can do either of the following:

- Delete an existing known image using the OpenVMS Install Utility.
- Use the AUTOGEN command procedure to increase the GBLPAGES or GBLSECTIONS system parameters.

AUTOGEN automatically adjusts values for parameters that are associated with the values you reset manually. To change system parameters with AUTOGEN, edit the following file:

```
SYS$SYSTEM:MODPARAMS.DAT
```

To change a parameter value listed in this file, delete the current value associated with that parameter and enter the new value.

To add a new parameter, insert a line in the file that includes both the name of the parameter and its value. For example:

```
WSMAX = 1024
```

To modify incremental parameters such as GBLPAGES and GBLSECTIONS, use ADD\_. The following example increases the global pagelet setting by 2000:

```
ADD_GBLPAGES = 2000
```

---

## Note

When you set the page file quota, do *not* use a value that exceeds the amount of page file space available on the system.

---

After you make all your changes, exit from the editor and then execute the AUTOGEN procedure to recalculate your system parameters. Enter the following command at the DCL prompt to recalculate your system parameters and to reboot the system:

```
$ @SYS$UPDATE:AUTOGEN GETDATA REBOOT
```

When you specify REBOOT, AUTOGEN does an automatic system shutdown and then reboots the system. Any users logged in to the system are immediately disconnected during the shutdown. The automatic reboot puts the new parameter values into effect.

AUTOGEN automatically adjusts some of the SYSGEN parameters, based on the consumption of resources since the last reboot. If you do not want to take advantage of this automatic adjustment, include the /NOFEEDBACK qualifier on the AUTOGEN command line.

For more information about using AUTOGEN, see the OpenVMS System Management Subkit.

## 1.4.3. Backing Up Your System Disk

At the beginning of the installation, VMSINSTAL asks if you have backed up your system disk. VSI recommends that you do a system disk backup before installing VSI Pascal or any other software.

Use the backup procedures that are established at your site. For details on performing a system disk backup, see the section on the Backup Utility in the OpenVMS system management documentation.

## 1.5. Stopping the Installation

To stop the installation procedure at any time, use Ctrl/C. However, files created up to that point are not deleted. You must then delete these files interactively. Chapter 2 lists the directories and files created during the VSI Pascal installation.



# Chapter 2. Installing VSI Pascal

This chapter explains how to install VSI Pascal:

- Section 2.1 contains a step-by-step description of the installation procedure.
- Section 2.2 describes error recovery considerations.

## 2.1. Installation Procedure

The VSI Pascal installation procedure consists of a series of questions and informational messages.

If you have not already done so, insert the Compact Disc (CD) or tape media into the CD drive or tape drive.

To abort the installation procedure at any time, press Ctrl/Y. When you press Ctrl/Y, the installation procedure deletes all files it has created up to that point and exits. You can then start the installation again.

When the system prompts you with a question during the installation procedure, the default answer, if any, is enclosed in brackets ([ ]).

To start the installation:

### 1. Invoke the VMSINSTAL command procedure.

You should invoke the VMSINSTAL command procedure from a privileged account, such as the SYSTEM account. The VMSINSTAL procedure is in the SYS\$UPDATE directory. For example:

```
Username: SYSTEM Password:  
  
$ SET DEFAULT SYS$UPDATE  
$ @SYS$UPDATE:VMSINSTAL (save-set-name) (device-name) [option-list]
```

The following list defines the elements of the VMSINSTAL procedure:

#### **save-set-name**

The installation name for the component. For VSI Pascal, use the following installation name:

```
PASCAL05n
```

To obtain the directory location of the VSI Pascal save set on the distribution media, see the documentation that accompanies the media. Or, you can do the following before starting the installation:

- To determine whether the CD drive is already mounted, enter the following command:

```
$ SHOW DEVICE DKA400:
```

If it is not mounted, enter the appropriate MOUNT command to mount the CD (omit the /FOREIGN qualifier).

- To obtain the directory name, enter the following command:

```
$ DIRECTORY DKA400:[000000]*pascal*.dir  
Directory DKA400:[000000]
```

```
PASCAL05n.DIR;1
```

- To obtain the save set name, use the `DIRECTORY` command specifying the directory file from the previous command:

```
$ DIRECTORY DKA400:[PASCAL05n.KIT]*.A
```

### **device-name**

The name of the device on which you plan to mount the media and the directory containing the VSI Pascal save set. In the following example, the device name is `DKA400:` and the directory is `[PASCAL05n.KIT]`. For tape installations, just enter the name of the tape device.

To specify the device name on the `@VMSINSTAL` command line, enter:

```
DKA400:[PASCAL05n.KIT]
```

`DKA400:` is the name of the CD drive used in the rest of the examples in this guide.

### **option-list**

The word `OPTIONS` followed by one or more of the following option designators: `A`, `L`, `N`, `R`. Their meanings are as follows:

- Auto\_answer option (`A`)

Initially creates a file that contains your answers to `VMSINSTAL` questions and prompts. You can then use the option (and the answer file) to save time during a reinstallation (typically after upgrading your system).

- File log option (`L`)

Logs all activity to the terminal during installation.

- Display or print the release notes (`N`)

Indicates that you want to see the installation question on release notes. If you do not include the `OPTIONS N` parameter, `VMSINSTAL` does not ask you about the release notes. You should review the release notes before proceeding with the installation in case they contain additional information about the installation. If you are restarting the installation and have already reviewed the release notes, you do not need to specify `OPTIONS N`.

---

## **Note**

Even if you omit the `N` option when you invoke `VMSINSTAL`, the VSI Pascal release notes are copied automatically to the `SY$HELP` directory.

---

- Alternate root option (`R`)

Allows you to install the product to a system root other than that of the running system.

For detailed information on these `VMSINSTAL` options, see the OpenVMS documentation on software installation in the OpenVMS System Management Subkit.



OPTIONS A,N

When you invoke VMSINSTAL, it checks the following:

- Whether you are logged in to a privileged account

VSI recommends that you install software from the system manager's account with your default device and directory set to SYS\$UPDATE.

- Whether you have sufficient quotas for installation

The following example invokes VMSINSTAL to install VSI Pascal and shows the system response. This example uses the OPTIONS N release note option.

```
$ @SYS$UPDATE:VMSINSTAL PASCAL05n DKA400:[PASCAL05n.kit] OPTIONS N
```

```
OpenVMS Software Product Installation Procedure Vn.n
It is dd-mmm-yyyy at hh:mm.
Enter a question mark (?) at any time for help.
```

If you do not supply the product name or the device name, VMSINSTAL prompts you for this information later in the installation procedure. VMSINSTAL does not prompt you for any options, so be sure to include whatever option you want on the VMSINSTAL command line.

## 2. Confirm system backup.

```
* Are you satisfied with the backup of your system disk [YES]?
```

VMSINSTAL asks if you are satisfied with your system backup. You should always back up your system disk before performing an installation. If you are satisfied with the backup of your system disk, press the Return key. Otherwise, enter NO to discontinue the installation. After you back up your system disk, you can restart the installation.

If you fail to include the device name of the VMSINSTAL command line, the following prompt is displayed:

```
* Where will the distribution volumes be mounted:
```

If this prompt is displayed, type the name of the distribution (media) device and directory that contains the VSI Pascal kit (save set). For example, if the media is contained on a CD located on unit DKA400:, type DKA400:[PASCAL05n.KIT] in reply to this question, as shown in the following example:

```
* Where will the distribution volumes be mounted:DKA400:[PASCAL05n.KIT]
```

## 3. Verify the product being installed.

The following products will be processed:

```
PASCAL V5.n
```

```
Beginning installation of PASCAL V5.n at hh:mm
```

```
%VMSINSTAL-I-RESTORE, Restoring product saveset A...
```

The product name and version are displayed. The first save set (A) is restored to a subdirectory of SYS\$UPDATE.

#### 4. Select a release notes option.

Release notes included with this kit are always copied to SYS\$HELP.

Additional Release Notes Options:

1. Display release notes
2. Print release notes
3. Both 1 and 2
4. None of the above

\* Select option [2]:

If you specified OPTIONS N when you invoked VMSINSTAL, you are now asked to choose one of the four options for reviewing the release notes:

- If you select option 1, VMSINSTAL displays the release notes immediately on the console terminal. You can terminate the display at any time by pressing Ctrl/C.
- If you select option 2, VMSINSTAL prompts you for the name of the print queue that you want to use:

\* Queue name [SYS\$PRINT]:

You can then enter a queue name or press Return to send the file to the default output print device.

- If you select option 3, VMSINSTAL displays the release notes immediately on the console terminal and then prompts you for a queue name for the printed version of the release notes.
- If you choose option 4, the release notes are not displayed or printed but are still copied to the following file in the SYS\$HELP directory:

```
SYS$HELP:PASCAL05n.RELEASE_NOTES
```

You can later use the HELP command to find the location of the release notes in the Help facility:

```
$ HELP PASCAL RELEASE_NOTES
```

Whichever option you choose, VMSINSTAL next asks the following question:

```
* Do you want to continue the installation [NO]?: YES
%VMSINSTAL-I-REMOVED, The product's release notes have been
successfully moved to SYS$HELP.
```

To continue the installation, enter YES. Otherwise, press Return. In either case, the release notes are copied to a file in the SYS\$HELP directory:

```
SYS$HELP:PASCAL059.RELEASE_NOTES
```

---

## Note

The name of the release notes file installed by VMSINSTAL consists of the current product name (in this case, PASCAL) and version number (in this case, 059 for Version 5.9). Do not delete release notes for previous versions of VSI Pascal.

---

#### 5. Install the compiler.

---

\* Do you want to install the VSI Pascal compiler [YES]?

The VSI Pascal kit contains the compiler, message files, command line definition, and help files.

## 6. Install STARLET.

\* Do you want to install the HP Pascal definition files [YES]?

The VSI Pascal kit contains definition files for OpenVMS System Services and several run-time library facility codes. These system definition files are built using information shipped with the OpenVMS operating system.

The installation procedure determines whether VSI Pascal definition files already exist. If the files do not exist, the installation procedure notifies you of this and advises you to create the definition files.

If the VSI Pascal definition files exist, but the system definitions supplied by OpenVMS (which are the basis for the VSI Pascal definition files) have been modified since the last time the VSI Pascal definition files were created, the installation procedure advises you to recreate the files.

If the VSI Pascal definition files exist and if the system definitions supplied by OpenVMS have not been modified since the last time the VSI Pascal definition files were created, the procedure tells you that it is not necessary to recreate the definition files.

## 7. Install the examples.

\* Do you want to install the HP Pascal example programs [YES]?

The VSI Pascal kit contains a set of example programs using many features of VSI Pascal and OpenVMS. The examples are installed in the directory `SYS$COMMON:[SYSHLP.EXAMPLES.PASCAL]`. You can access them using the logical `PASCAL$EXAMPLES`, if you run the optional startup file `SYS$STARTUP:PASCAL$STARTUP.COM` (see Section 3.3).

## 8. Save previous compiler.

\* Do you want to save your existing HP Pascal compiler [NO]?

The VSI Pascal kit contains support to save your existing VSI Pascal compiler for access later. The installation provides various command files to display and select the saved VSI Pascal compilers. See Section 1.3 for more information.

## 9. Purge files.

\* Do you want to purge files replaced by this installation [YES]?

You have the option to purge files from previous versions of VSI Pascal that are superseded by this installation. Purging is recommended. If you answered YES to question 8, you can answer YES to this question with no danger of deleting any saved compilers.

## 10. Choose the Installation Verification Procedure (IVP) option.

The installation procedure now asks if you want to run the IVP. The IVP for VSI Pascal checks that the installation is successful. VSI recommends that you run the IVP.

\* Do you want to run the IVP after the installation [YES]?

This kit contains an IVP that is used to verify the correct installation of the VSI Pascal compiler. The IVP is located in:

```
SYS$TEST:PASCAL$IVP.COM
```

After VSI Pascal is installed, you can run the IVP independently to verify that the software is available on your system. You might need to run the IVP after a system failure to make sure that users can access VSI Pascal.

#### 11. Respond to license registration queries.

```
Product:      PASCAL
Producer:     DEC
Version:      V5.n
Release Date: dd-mmm-yyyy
```

\* Does this product have an authorization key registered and loaded?

The installation procedure displays license information about your product and then asks if you have registered and loaded your Product Authorization Key (PAK).

If you have not registered and loaded your PAK, you must answer NO to this question. You must register and load your PAK to successfully complete the installation. If you have not done so, you should stop the installation, register and load your PAK, and then start the installation procedure again.

#### 12. Read informational messages.

At this point, the installation procedure displays a number of informational messages that report on the progress of the installation. There are no further questions. If the installation procedure has been successful up to this point, VMSINSTAL moves the new or modified files to their target directories, updates help files, and updates DCL tables, if necessary. If you chose to have files purged, that work is done now. The following message is displayed:

```
%VMSINSTAL-I-MOVEFILES, files will now be moved to their
      target directories...
```

#### 13. Observe the Installation Verification Procedure (IVP). If you chose to run the IVP, VMSINSTAL runs it now. When the IVP runs successfully, you see the following messages displayed:

```
** Successful Installation of HP Pascal V5.n on 16-JUN-2004 at 13:35 **
      Compiler Version = HP Pascal V5.n
*-----*
* Installation Verification Procedure for      *
* HP Pascal STARLET libraries                 *
*-----*
```

#### 14. End the installation procedure.

```
Installation of HP Pascal V(n.n) completed at hh:mm
VMSINSTAL procedure done at hh:mm
```

These messages indicate that the entire installation procedure is complete.

You can now log out of the privileged account:

```
$ LOGOUT
SYSTEM      logged out at dd-mm-yyy 20:20:02
```

---

## Note

VMSINSTAL deletes or changes entries in the process symbol tables during the installation. Therefore, if you are going to continue using the system manager's account and you want to restore these symbols, you should log out and log in again.

---

## 2.2. Error Recovery

If errors occur during the installation itself or when the Installation Verification Procedure (IVP) is running, VMSINSTAL displays failure messages. If the installation fails, you see the following message:

```
%VMSINSTAL-E-INSFAIL, The installation of HP Pascal V(n.n) has failed.
```

If the IVP fails, you see these messages:

```
The HP Pascal V(n.n) Installation Verification Procedure failed.
%VMSINSTAL-E-IVPFAIL, The IVP for HP Pascal V(n.n) has failed.
```

Errors can occur during the installation if any of the following conditions exist:

- The operating system version is incorrect.
- Quotas necessary for successful installation are insufficient.
- System parameter values for successful installation are insufficient.
- The OpenVMS help library is currently in use.
- The product license has not been registered and loaded.

For descriptions of the error messages generated by these conditions, see the OpenVMS documentation on system messages, recovery procedures, and OpenVMS software installation. If you are notified that any of these conditions exist, you should take the appropriate action as described in the message. For information on installation requirements, see Section 1.4.



# Chapter 3. After Installing VSI Pascal

VSI Pascal is now installed and can be invoked by all users with the PASCAL command. You can perform the following tasks after installing VSI Pascal:

- Running the Installation Verification Procedure (IVP) separately
- Making sure privilege and quota requirements are met
- Editing the system startup and shutdown files
- Enhancing performance
- Reporting problems

The following sections describe these tasks in detail.

## 3.1. Running the Installation Verification Procedure (IVP) Separately

You usually run the Installation Verification Procedure (IVP) during installation. If you want to run the IVP separately to ensure the integrity of installed files should system problems occur, enter the following command procedure:

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

If the IVP fails for any reason, the following message is displayed:

```
HP Pascal VERSION V5.n TEST FAILED
```

## 3.2. User Account Requirements

Several products cannot be used unless certain privilege and quota requirements are met by user accounts.

VSI Pascal does not require any special privileges. There is no benefit from installing the VSI Pascal compiler with privileges.

See Section 3.4 for information about quotas.

## 3.3. Editing System Files

You can edit the system startup file to provide for automatic startup of VSI Pascal when your system is rebooted. Use the optional startup file to define the PASCAL\$EXAMPLES logical name and to install VSI Pascal as a shared image (see Section 3.4.2).

Edit the system startup file SYS\$STARTUP:SYSTARTUP\_VMS.COM (or SYS\$STARTUP:SYSTARTUP\_V5.COM on OpenVMS VAX V5.n systems) to include the following line:

```
$ @sys$startup:pascal$startup.com
```

## 3.4. Enhancing VSI Pascal Performance

This section includes information on system tuning and explains how to install VSI Pascal as a shared image.

### 3.4.1. Using Multiple Environment Files (OpenVMS VAX Systems Only)

If you intend to use multiple VSI Pascal environment files, you can increase the compilation speed by modifying the FILLM quota of those processes that run the compiler. The VSI Pascal compiler will keep more than one environment file open during the compilation (with a maximum of 50 environment files). If you inherit more than 50 environment files, the compiler will close some of them in order to open other files.

### 3.4.2. Installing VSI Pascal as a Shared Image

If you expect VSI Pascal to be used extensively on your system, you can reduce the system overhead and memory requirements by installing it as a shared image.

To install VSI Pascal as a shared image on a system that is currently running, use the optional VSI Pascal startup file:

```
SYS$STARTUP:PASCAL$STARTUP.COM
```

VSI recommends that you install VSI Pascal as shared on a system that has been rebooted recently because the available space in the global page table is less likely to be fragmented.

The requirements for additional global pages and global sections are listed in Section 1.4.2.

### 3.4.3. Improving VSI Pascal Compilation Performance on OpenVMS I64 and OpenVMS Alpha Systems

For many user applications, the VSI Pascal compiler for OpenVMS I64 and OpenVMS Alpha systems requires significantly more system resources than VSI Pascal for OpenVMS VAX. The compiler will sometimes fail with a “%LIB-E-INSVIRMEM, insufficient virtual memory” error. Also, for very large programs (greater than 10,000 lines), there could be a report of an extremely long compile time.

The following sections explain why VSI Pascal for OpenVMS I64 and OpenVMS Alpha systems requires more memory, and suggests actions to be taken to avoid resource problems.

The Alpha architecture is a RISC (Reduced Instruction Set Computer) architecture. The Itanium architecture, while not technically a RISC architecture, has many of the same details that require additional compiler resources. The VAX architecture is a CISC (Complex Instruction Set Computer) architecture. The main distinguishing characteristic of a RISC machine is that it has few instructions, and each instruction does a small amount of work. A CISC machine generally has many instructions, most of which perform many complicated operations in one step.

By reducing the amount of work that is done in each instruction (and by reducing the number of instructions), the complexity of the hardware is reduced. This reduction in complexity allows the hardware designers to raise the clock speed (and thus the number of instructions per second that can be completed). This results in better overall system performance.



Another characteristic of RISC systems is that compilers for these architectures generally must do a great deal more work than a corresponding compiler for a CISC architecture.

For example, the compiler must compute the best way to use all of the functional units of the processor, and it must determine how to make the best use of registers and on-chip data cache, because reads and writes to main memory are generally slow compared to the speed of the processor.

The VSI Pascal compiler for OpenVMS I64 and OpenVMS Alpha systems is a globally optimizing compiler. The compiler constructs an internal representation of the entire Pascal program, and repeatedly traverses this structure at compile time, so that it can produce the most efficient machine code for the program. This results in high performance code, but the drawback is that the compiler requires more resources than the VSI Pascal for OpenVMS VAX compiler.

The VSI Pascal compiler for OpenVMS I64 and OpenVMS Alpha will require more resources than the VSI Pascal for OpenVMS VAX compiler because it must do more work. However, there are some things that can be done to reduce system resource usage:

- Use the /NOOPTIMIZE qualifier when “checking” programs.

The optimizer is one of the biggest users of system resources in the Pascal compiler and is turned on by default. Also, the higher the optimization level, the more memory required by the compiler.

- Check system tuning.

Since the Pascal compiler often needs a great deal of virtual memory, increase working-set limits for developers that use the compiler. This results in decreased paging and improvements in compile time.

- Check program sizes.

The bigger the program is, the more resources it requires.

Section 3.4.3.1 and Section 3.4.3.2 provide information on memory requirements for VSI Pascal programs on OpenVMS I64 and OpenVMS Alpha systems.

In many tuning exercises, it is instructive to work with a relatively quiet system, submit batch jobs, and retain the log files for later analysis.

---

## Note

The information in the following section is based on a study of a set of large modules. Note that your results may differ.

---

For more information on OpenVMS tuning, see the *VSI OpenVMS System Manager's Manual, Volume 2: Tuning, Monitoring, and Complex Systems*.

### 3.4.3.1. Required Virtual Memory

If there is not enough virtual memory, the compile command may fail with the message “insufficient virtual memory.”

OpenVMS has two parameters that control the amount of virtual memory available to a process:

- The system generation parameter VIRTUALPAGECNT, which sets an upper bound on the number of pagelets of virtual memory for any process in the system.

- The `AUTHORIZE` parameter `PGFLQUOTA`, which determines the number of pagelets a process can reserve in the system's page file(s).

After an “insufficient virtual memory” error, issue the following `DCL` command to see the peak virtual size used by the process, or look at the peak page file size at the end of a batch job's log:

```
SHOW PROCESS/ACCOUNTING
```

If the peak size is at the system generation parameter `VIRTUALPAGECNT`, you will need to raise this value. If the peak size is below `VIRTUALPAGECNT`, and at or above `PGFLQUOTA`, run `AUTHORIZE` to increase `PGFLQUOTA` for the Pascal users. Peak size can exceed `PGFLQUOTA` because some virtual memory, such as read-only image code, is not allocated page file space.

It is difficult to predict precisely how much virtual memory will be required for a compilation, but a starting point for system tuning can be computed by either of the following:

- Multiply 250 times the size of the largest program in disk blocks (including all `INCLUDE` files referenced).
- Multiply 25 times the number of lines in the program (including all `INCLUDE` files).

The resulting figure can then be used as a starting point for the system generation parameter `VIRTUALPAGECNT`. Place a line in the parameter file `SYS$SYSTEM:MODPARAMS.DAT`. For example, if you estimate 370,000 pages, add the following line in `MODPARAMS`, run `AUTOGEN` and reboot.

```
MIN_VIRTUALPAGECNT = 400000
```

If the compilation now completes successfully, use the command `SHOW PROCESS/ACCOUNTING` to determine the Peak Virtual Size; if the actual peak is significantly less than the value computed above, you can reduce `VIRTUALPAGECNT`.

When modifying `VIRTUALPAGECNT` and `PGFLQUOTA`, you may also need to increase the size of the page file.

### 3.4.3.2. Required Physical Memory

When dealing with physical memory you must know:

- If there is enough memory on the system
- If there is enough memory available to the process running the compilation

If the physical memory on the system is too small, the command `LOGOUT/FULL` (which is automatically issued at the end of a batch job) will show a high number of faults, for example, > 100,000 for a single compilation, and an “Elapsed time” that greatly exceeds the “Charged CPU time,” as the system waits for disk I/Os to resolve page faults. In this situation, tuning attempts may be of limited benefit.

If the physical memory on the system is adequate, but the physical memory allotted to the process running the compilation is too small, you may still observe a large number of faults, but elapsed time may remain closer to CPU time. This is because OpenVMS resolves page faults from the page caches (free list, modified list) whenever possible, avoiding the relatively slow disk I/Os. In this situation, basic tuning may also be beneficial.

The amount of physical memory required will vary, but it should be a large fraction of the process peak virtual size—as close to 1.0 as practical. The reason for this is that the compiler makes multiple passes

over the internal representation of the program; therefore a page which falls out of the working set in one pass is probably going to be needed again on the very next pass.

The physical memory present on the system can be determined by the command `SHOW MEMORY/PHYSICAL`. The physical memory used by the compilation is reported as “Peak working set size” by the command `SHOW PROCESS/ACCOUNTING` or at the end of a batch log file.

More physical memory can be made available to a process by minimizing the number of competing processes on the system (for example, by compiling one module at a time, or scheduling large compiles for off hours.)

It can also be made available to a process (if it is present on the machine) by adjusting the system generation parameter `WSMAX` and the corresponding `WSEXTENT` (in `AUTHORIZE`). Such adjustments must be attempted with great caution, as the system may hang if memory is oversubscribed and you create a situation where OpenVMS effectively has no options to reclaim memory.

The following guidelines are useful:

- Set the Pascal users' `WSEXTENT` (in `AUTHORIZE` or `INITIALIZE/QUEUE`) to match `WSMAX`.
- Keep `WSQUOTA` (in `AUTHORIZE` or `INITIALIZE/QUEUE`) low. Make sure that no process or batch queue has a `WSQUOTA` of more than about 20% of physical memory. The difference between `WSEXTENT` and `WSQUOTA` allows OpenVMS to manage memory to meet varying demands.
- Use `AUTOGEN`. `AUTOGEN` will attempt to make a consistent set of changes that do not interfere with each other.

By default, `AUTOGEN` will set the maximum working set (system generation parameter `WSMAX`) to 25% of physical memory. This value is reasonable for a workstation or multiuser system with many active processes.

- `WSMAX` can be increased to a somewhat larger value by editing `MODPARAMS.DAT`. For a system with 64 MB of physical memory, try setting `WSMAX` to no more than about 40% of physical memory, or 52000 pagelets. With 128 MB or more of physical memory, a setting of 50% of physical memory can be attempted.

### 3.4.3.3. Effects of Physical Memory

The effects of physical memory on compilation time were studied for a set of seven large modules. The modules ranged in size from about 1600 to 3300 disk blocks. Your results may differ; but to help give a rough appreciation for the effect of physical memory on compilation time, the following can be noted:

- When the amount of physical memory available to the processes matched the amount of virtual memory, the elapsed times were close to the CPU times.
- As the physical memory was reduced, CPU times rose only slightly, about ten percent.
- As the physical memory was reduced, elapsed times were elongated, at the rate of about 1 hour for each 100 MB of difference between Peak Virtual Size and the actual memory available. For example, when compiling a program that used a Peak Virtual Size of 947760 pagelets, or 463 MB (1 MB = 2048 pagelets), on a system where about 180 MB of physical memory was available to user processes, the compile required about three hours longer than on a 512 MB system.

Your results will be strongly affected by the speed of the devices that are used for paging. Both virtual memory and physical memory requirements can be reduced by breaking large modules up into smaller modules.

