VMS Software

PHP Version 8.0-11 for VSI OpenVMS x86-64 and IA-64

Release Notes

Publication Date: November 2026

Operating Systems: VSI OpenVMS x86-64 Version 9.2-3 or higher
VSI OpenVMS IA-64 Version 8.4-2L1 or higher

Software Version: PHP Version 8.0-11 for VSI OpenVMS x86-64 and IA-64

Kit Names: VSI-X86VMS-PHP-V0800-11-1.PCSI
VSI-I64VMS-PHP-V0800-11-1.PCSI

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

Table of Contents

1. Introduction 3
2. Acknowledgments 3
3. What's New in This Release 3
4. Requirements 3
5. Pre-Installation Steps 4
6. Installing the Kit 4
7. Post-Installation Steps 5
8. Using PHP with Apache Web Server 5
9. Included Extensions 6
10. Known Problems and Restrictions 6

Important

Unlike the CSWS_PHP software kit provided by HPE, this PHP kit for VSI OpenVMS does not have any direct or implied association with CSWS (Apache HTTPD for OpenVMS) and it can be installed and used independently of CSWS. The kit does include both MOD_PHP (the PHP module for Apache HTTPD) and support for PHP CGI scripts with CSWS; however, these optional facilities must be configured manually.

1. Introduction

Thank you for your interest in this port of PHP to VSI OpenVMS. The current release of PHP for VSI OpenVMS is based on the PHP 8.0 Open Source distribution.

PHP (http://php.net [https://www.php.net]) is a popular general-purpose Open Source scripting language that is especially suited to web development.

This VSI OpenVMS port of PHP includes the core PHP language interpreter and various extensions (dynamically loadable modules), including extensions for various database client APIs, XML, compression, arbitrary precision arithmetic, web services (SOAP and REST), and graphics. It is anticipated that additional extensions will be included in future releases.

2. Acknowledgments

VMS Software Inc. would like to acknowledge the work of the core PHP development team for their ongoing efforts in developing and supporting this product.

3. What's New in This Release

This release of PHP for VSI OpenVMS x86-64 and IA-64 includes the following updates:

- Fixes for CVE-2026-109618 and CVE-2026-109619 ("PHP2Shell").
- LIBGD$SHR.EXE is now included in the kit, resolving an issue that could prevent the GD and MYSQLND extensions from loading correctly.

For additional information about changes included in PHP 8.0, see https://www.php.net/ChangeLog-8.php#PHP_8_0.

4. Requirements

VSI PHP 8.0-11 can be installed by a suitably privileged user on an OpenVMS system that meets the following software requirements:

- One of the following VSI OpenVMS operating systems:
  - VSI OpenVMS x86-64 Version 9.2-3 or higher
  - VSI OpenVMS IA-64 Version 8.4-2L1 or higher
- PHP must be installed on an ODS-5 enabled disk (the installation will fail if this requirement is not met)

- VSI TCP/IP, HPE TCP/IP Services for OpenVMS, or MultiNet TCP/IP stack for network communication
- VSI Apache Web Server Version 2.4-71 (Optional)

  Apache Web Server is required only if you intend to use PHP with the Apache HTTP Server.

  Previous versions of the Apache Web Server (CSWS) are no longer supported. Remove any existing CSWS installation before installing Apache Web Server. For installation instructions, refer to the *VSI Apache Web Server V2.4-71 Release Notes and Installation Guide* [https://docs.vmssoftware.com/vsi-apache-web-server-based-on-apache-http-server-release-notes-v2-4-71/].
- VSI C Version 7.7-003 for OpenVMS x86-64, or Version 7.4-001 for OpenVMS IA-64 (Optional)

  VSI C is required only if you wish to implement your own extensions.

In addition to the above requirements, it is assumed that the reader has a good knowledge of OpenVMS and of software development in the OpenVMS environment.

5. Pre-Installation Steps

Before installing VSI PHP for OpenVMS, you must manually remove any existing version of HP CSWS_PHP installed on your system. Use PCSI to remove CSWS_PHP by entering the following command:

```
$ PRODUCT REMOVE CSWS_PHP
```

6. Installing the Kit

The kit is provided as an OpenVMS PCSI kit (VSI-X86VMS-PHP-V0800-11-1.PCSI or VSI-I64VMS-PHP-V0800-11-1.PCSI, depending on platform) that can be installed by a suitably privileged user using the following command:

```
$ PRODUCT INSTALL PHP
```

The installation will then proceed as follows (output may differ slightly from that shown, depending on platform and other factors):

```
The following product has been selected:
    VSI X86VMS PHP V8.0-11                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 PHP V8.0-11: PHP for OpenVMS is based on PHP Version 8.0

    (C) Copyright 2026 VMS Software Inc.

    VSI Software Inc.

* This product does not have any configuration options.

Execution phase starting ...
```

```
The following product will be installed to destination:
    VSI X86VMS PHP V8.0-11                DISK$4012SYS:[VMS$COMMON.]

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

The following product has been installed:
    VSI X86VMS PHP V8.0-11                Layered Product

VSI X86VMS PHP V8.0-11: PHP for OpenVMS is based on PHP Version 8.0

    Post-installation tasks are required.

    To define the PHP runtime at system boot time, add the
    following lines to SYS$MANAGER:SYSTARTUP_VMS.COM:

        $ file := SYS$STARTUP:PHP$STARTUP.COM
        $ if f$search("'''file'") .nes. "" then @'file'
```

7. Post-Installation Steps

After the installation has completed successfully, add the commands displayed at the end of the installation procedure to SYSTARTUP_VMS.COM. This ensures that the logical names required for users to access PHP are defined system-wide during system startup.

Users can then invoke the PHP interpreter by defining the following foreign command:

```
$ PHP :== $PHP$ROOT:[BIN]PHP.EXE
```

In general, there are no special quota or privilege requirements for users developing applications with PHP. However, some extensions may have additional requirements. For example, networking extensions may require a higher BYTLM quota.

You may also wish to customize the system-wide PHP initialization file, PHP$ROOT:[ETC]PHP.INI, to specify which PHP extensions are loaded automatically and to define other default settings appropriate for your environment.

Example initialization files for typical development and production environments are provided in PHP$ROOT:[ETC]. These files can be used as the basis for creating a PHP.INI configuration tailored to your site's requirements.

8. Using PHP with Apache Web Server

To use PHP with Apache Web Server to process dynamic web pages using MOD_PHP, perform the following steps. You may also wish to define one or more aliases for directories that will be used to serve PHP scripts.

1. Copy MOD_PHP.EXE into the Apache modules directory by entering the following command:

   ```
   $ COPY/LOG PHP$ROOT:[CSWS]MOD_PHP.EXE APACHE$COMMON:[MODULES]
   ```

2. Edit the Apache configuration file (APACHE$COMMON:[CONF]HTTPD.CONF) and add the following directives. It is recommended that these directives be placed at the end of the file.

   ```
   ## Load PHP module
   LoadModule php_module modules/mod_php.exe
   ```

```
## Define types to be associated with MOD_PHP
AddType application/x-httpd-php .php .phtml
AddType application/x-httpd-php-source .phps
```

3. Stop and restart Apache Web Server to apply the modified configuration by entering the following commands:

```
$ @SYS$STARTUP:APACHE$SHUTDOWN.COM
$ @SYS$STARTUP:APACHE$STARTUP.COM
```

4. To test the PHP module, create a simple script named DEMO.PHP (or use any other name you wish) in APACHE$COMMON:[HTDOCS] with the following contents:

```
<?php
echo "Testing the PHPINFO () function<br>\n";
phpinfo (INFO_ALL);
?>
```

   Then, browse to http://*my.host.name*/demo.php, where *my.host.name* is the name or TCP/IP address of your web server.

   If MOD_PHP is functioning correctly, the PHP information page will be displayed, showing details of your PHP installation, configuration settings, and enabled features.

9. Included Extensions

This release of PHP for OpenVMS includes the core PHP language and a number of commonly used dynamically loadable extensions. The included extension modules are listed below. Note that some extensions are currently available only on IA-64 systems.

BCMATH.EXE	MBSTRING.EXE	REC.EXE
BZ2.EXE	MYSQLI.EXE (IA-64 only)	SHMOP.EXE
CALENDAR.EXE	MYSQLND.EXE (IA-64 only)	SOAP.EXE
CURL.EXE	ODBC.EXE	SOCKETS.EXE
DBA.EXE	OPCACHE.EXE	SYSVSEM.EXE
DLM.EXE	OPENSSL.EXE	SYSVSHM.EXE
EXIF.EXE	OPENVMS.EXE	TIDY.EXE
FTP.EXE	PDO_DBLIB_FREETDS.EXE	XSL.EXE
GD.EXE	PDO_MYSQL.EXE (IA-64 only)	ZIP.EXE
GMP.EXE	RDB.EXE	ZLIB.EXE
LDAP.EXE	READLINE.EXE	

Additional extensions are statically included in the PHP interpreter. To display information about these modules, use the command **`PHP -m`**.

10. Known Problems and Restrictions

- All PHP script files must be Stream_LF.

- Composer Dependency Manager for PHP (https://getcomposer.org) does not currently function correctly on VSI OpenVMS. It is anticipated that an OpenVMS-compatible version of this tool will be included in a future release of PHP for VSI OpenVMS.
- When using PHP with Apache Web Server and MOD_PHP, temporary files are created in SYS$SCRATCH, which for the APACHE$WWW account equates to APACHE$ROOT:[000000]. These files are created with names of the form ZendSem.xxxxx.

  To ensure that these files are deleted when web server worker processes terminate, define the logical name DECC$ALLOW_REMOVE_OPEN_FILES as "TRUE" in APACHE$WWW's LOGIN.COM. Alternatively, implement procedures to periodically remove these files.
- Just-In-Time (JIT) Compilation for PHP is not currently supported on VSI OpenVMS.