



## CSWS\_PHP for HP Secure Web Server for OpenVMS Alpha (based on Apache)

### Installation Guide and Release Notes

October 2003

CSWS\_PHP V1.2  
CPQ-AXPVMS-CSWS\_PHP-V0102--1.PCSI-DCX-AXPEXE

CSWS\_PHP V1.1  
CPQ-AXPVMS-CSWS\_PHP-V0101--1.PCSI-DCX-AXPEXE

### Contents:

- what's new in version 1.2
- software prerequisites
- documentation
- downloading the kit
- expanding the kit
- installing CSWS\_PHP for OpenVMS
- removing CSWS\_PHP for OpenVMS
- using extensions
  - ODBC extension
  - OCI8 extension
- PHP sample scripts
  - calendar
  - info
  - ODBC
  - OpenVMS
- release notes
  - PHP\$ logical names changed to APACHE\$ in version 1.2
  - add logical names to PHP\_SETUP.COM for extended file name support
  - configuring CSWS\_PHP not required
  - PHP DNS functions supported only with TCP/IP Services for OpenVMS
  - PHP LINK functions not supported

### What's New in CSWS\_PHP Version 1.2

- **MySQL extension**, which allows you to access MySQL database servers. More information about MySQL can be found at <http://www.mysql.com/>.
- **OpenVMS extension**, which performs several functions such as converting filenames and showing uptime. See the OpenVMS sample script for more information.
- **Updated OpenSSL extension**, based on HP SSL Version 1.1-A for OpenVMS.
- **Based on PHP 4.3.2.** (CSWS\_PHP V1.1 was based on PHP 4.1.1, and CSWS\_PHP V1.0 was based on PHP 4.0.4PL1.)

### Software Prerequisites

The **CSWS\_PHP** kit requires that the following software is installed *before* you install CSWS\_PHP:

OpenVMS Alpha Version 7.2-2 or higher  
HP Secure Web Server Version 1.3 (or higher) for OpenVMS Alpha

Note: CSWS\_PHP on OpenVMS Version 7.3-1 requires the VMS731\_UPDATE or VMS731\_ACRTL patch available from <http://ftp.support.compaq.com/patches/.new/openvms.shtml>

## Documentation

For more information about PHP, see <http://www.php.net/>.

## Downloading the Kit

Download CSWS\_PHP for HP Secure Web Server for OpenVMS Alpha self-extracting file from [http://h71000.www7.hp.com/openvms/products/ips/apache/csws\\_php.html](http://h71000.www7.hp.com/openvms/products/ips/apache/csws_php.html)

Make sure the file type of the saved file is .PCSI-DCX-AXPEXE.

## Expanding the Kit

To expand the CSWS\_PHP for OpenVMS Alpha self-extracting file, enter one of the following commands, depending on the kit you download:

```
$ RUN CPQ-AXPVMS-CSWS_PHP-V0102--1.PCSI-DCX-AXPEXE
$ RUN CPQ-AXPVMS-CSWS_PHP-V0101--1.PCSI-DCX-AXPEXE
```

The system displays information about the file compression version, and help information about the command syntax.

At the `Decompress into (file specification):` prompt, press return. The system expands the file and names the decompressed file CPQ-AXPVMS-CSWS\_PHP-V0102--1.PCSI or CPQ-AXPVMS-CSWS\_PHP-V0101--1.PCSI. Do not rename this file.

## Installing CSWS\_PHP for OpenVMS

**Important:** If you are upgrading from an earlier version of CSWS\_PHP and you modified PHP.INI, copy the file to another directory before you begin the installation. After the installation is complete, compare your old PHP.INI with the newly installed version and modify the new file as necessary.

Before you install CSWS\_PHP (or any optional module), **shut down the Secure Web Server**. You can restart the server when the installation is complete.

To install the CSWS\_PHP for *HP Secure Web Server for OpenVMS Alpha* kit, enter the following command.

**Note:** You must install the CSWS\_PHP kit into the same device and directory where you installed *HP Secure Web Server for OpenVMS*.

For example:

```
$ SHOW LOGICAL APACHE$ROOT
  "APACHE$ROOT" = "DISK1:[WEB_SERVER.APACHE.SPECIFIC.hostname.]"
                = "APACHE$COMMON:"
1  "APACHE$COMMON" = "DISK1:[WEB_SERVER.APACHE.]"

$ PRODUCT INSTALL CSWS_PHP/DESTINATION=DISK1:[WEB_SERVER]
```

For a description of the features you can request with the PRODUCT INSTALL command when starting an installation such as running the IVP, purging files, and configuring the installation, see the *POLYCENTER Software Installation Utility User's Guide*.

As the installation procedure for CSWS\_PHP V1.2 progresses, the system displays the following information. A similar log is displayed for CSWS\_PHP V1.1.

The following product has been selected:

```
CPQ AXPVMS CSWS_PHP V1.2          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.

```
CPQ AXPVMS CSWS_PHP V1.2
```

```
    Hewlett-Packard Company & The Apache Software Foundation.
```

\* This product does not have any configuration options.

Execution phase starting ...

The following product will be installed to destination:

```
    CPQ AXPVMS CSWS_PHP V1.2          DISK1:[WEB_SERVER.]
```

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

The following product has been installed:

```
    CPQ AXPVMS CSWS_PHP V1.2          Layered Product
```

```
CPQ AXPVMS CSWS_PHP V1.2
```

Post-installation tasks are required for PHP for OpenVMS.

The release notes give detailed directions.

This information is a brief checklist.

This installation modifies APACHE\$ROOT:[CONF]HTTPD.CONF to enable MOD\_PHP. Check HTTPD.CONF for accuracy. The line "Include /apache\$root/conf/mod\_php.conf" should be the only difference. Also study the MOD\_PHP configuration file (APACHE\$ROOT:[CONF]MOD\_PHP.CONF) for options required for your site.

The Apache server must be shut down and restarted to make these changes to HTTPD.CONF file take place. Test that MOD\_PHP is working by accessing the sample script from a browser:

```
    http://<your web server host>/php/php_rules.php
```

Thank you for using Apache for OpenVMS.

After the installation is complete, start the *Secure Web Server* by entering the following command:

```
    $ @SYS$STARTUP:APACHE$STARTUP
```

### **Removing CSWS\_PHP for HP Secure Web Server for OpenVMS**

You can remove the CSWS\_PHP kit by using the POLYCENTER Software Installation utility. The *HP Secure Web Server* must be shut down before you remove CSWS\_PHP.

To remove CSWS\_PHP, enter the following commands:

```
$ SET DEF APACHE$ROOT: [000000]
$ @SYS$STARTUP:APACHE$SHUTDOWN
$ PRODUCT REMOVE CSWS_PHP
```

The PRODUCT REMOVE command removes all files created by this installation. It also attempts to remove the directories defined by this installation.

## Using Extensions

**CSWS\_PHP V1.2 includes the extensions listed in the PHP.INI file shown below.** There are two ways to load a PHP extension: using the dl() function and using the PHP.INI file. These methods are as follows:

- The dl() function allows the loading of extensions within a PHP script if the extension resides in the default extension\_dir. Extension\_dir is defined as PHP\_ROOT:[EXTENSIONS] directory (APACHE\$COMMON:[PHP.EXTENSIONS]).
- The PHP.INI file provided with the CSWS\_PHP kit resides in the PHP\_ROOT:[000000] directory (APACHE\$COMMON:[PHP]). PHP.INI contains the extension statement to automatically load the extension for every PHP script executed. To enable the loading of the extension for every PHP script, uncomment the "extension=" statement as shown in the following example, and restart the *Secure Web Server*.

### PHP . INI

```
;
; Uncomment for the automatic loading of extensions
;
;extension=php_bcmath.exe
;extension=php_bzip2.exe
;extension=php_calendar.exe
;extension=php_ctype.exe
;extension=php_dba.exe
;extension=php_exif.exe
;extension=php_ftp.exe
;extension=php_iconv.exe
;extension=php_ldap.exe
;extension=php_mhash.exe
;extension=php_mysql.exe
;extension=php_oci8.exe
;extension=php_odbc.exe
;extension=php_openssl.exe
;extension=php_openvms.exe
;extension=php_oracle.exe
;extension=php_pcre.exe
;extension=php_posix.exe
;extension=php_session.exe
;extension=php_sockets.exe
;extension=php_xml.exe
;extension=php_zip.exe
;extension=php_zlib.exe
```

## ODBC Extension

The ODBC extension works with any ODBC V2.5 capable server. The ODBC.PHP script works with Attunity Connect "On Platform" Package for OpenVMS Alpha using the RMS demo.

Add the following lines to the beginning of the APACHE\$COMMON:[000000]LOGIN.COM file to enable the script to work with Attunity Connect:

```
$ !
$ ! Run the Attunity login if we find it
$ !
$ IF F$SEARCH ("NAVROOT: [BIN] NAV_LOGIN.COM") .NES. ""
$ THEN
$     @NAVROOT: [BIN] NAV_LOGIN.COM
$     DEFINE APACHE$ODBC_SHR ODNAVSHR
$     DEFINE APACHE$ODBC_PFX NV
$ ENDIF
```

The two logicals required to make the ODBC extension functional are APACHE\$ODBC\_SHR and APACHE\$ODBC\_PFX. These logicals are defined as follows:

- APACHE\$ODBC\_SHR  
This logical defines the ODBC shareable image to be used for the ODBC access.
- APACHE\$ODBC\_PFX  
This logical defines, if needed, any ODBC API prefix.

### OCI8 Extension

The OCI extension works with any Oracle 8 or 9 database server. Add the following lines to the beginning of the file APACHE\$COMMON:[000000]LOGIN.COM:

```
$ !
$ ! Define the OCI extension logicals if we find the OCI client shareable
$ !
$ IF F$SEARCH ("ORA_ROOT: [UTIL] ORACLIENT_V817.EXE") .NES. ""
$ THEN
$     DEFINE APACHE$OCI_SHR ORA_ROOT: [UTIL] ORACLIENT_V817.EXE
$ ENDIF
```

The two logicals required to make the OCI8 extension functional are APACHE\$OCI\_SHR and APACHE\$OCI\_PFX. These logicals are defined as follows:

- APACHE\$OCI\_SHR  
This logical defines the OCI8 shareable image to be used for the OCI8 access.
- APACHE\$OCI\_PFX  
This logical defines, if needed, any OCI8 API prefix.

### Sample PHP Scripts

New PHP sample scripts are included in the CSWS\_PHP V1.2 kit (calendar.php, info.php, odbc.php, and php\_openvms.php). These scripts demonstrate the use of the provided extensions.

---

**PHP\_CALENDAR.PHP** (Useful for Hebrew calendar in Unicode)

```
<?php

#
# Load the calendar extension if needed
#
if (! extension_loaded ("calendar"))
    dl ("php_calendar");
```

```

#
# Display the header
#
echo "    Testing the Calendar extension<br>\n";
#
# Test the calendar functions
#
$m = date("m", time());
$d = date("d", time());
$y = date("Y", time());
$jd = GregorianToJD($m,$d,$y);
echo "Gregorian month (abbr.): " . jdmmonthname($jd, 0) . "<br>\n";
echo "Gregorian month: " . jdmmonthname($jd, 1) . "<br>\n";
echo "Julian month (abbr.): " . jdmmonthname($jd, 2) . "<br>\n";
echo "Julian month: " . jdmmonthname($jd, 3) . "<br>\n";
echo "Jewish month: " . jdmmonthname($jd, 4) . "<br>\n";
$y = 1800;
$jd = GregorianToJD($m,$d,$y);
echo "French month: " . jdmmonthname($jd, 5) . "<br>\n";

?>

```

---

#### PHP\_INFO.PHP

```

<?php

#
# Display the header
#
echo "    Testing the PHPINFO () function<br>\n";
#
# Test the PHPINFO () function
#
phpinfo (INFO_ALL);

?>

```

---

#### PHP\_ODBC.PHP

```

<?php

#
# Load the ODBC extension if needed
#
if (! extension_loaded ("odbc"))
    dl ("odbc");

#
# Display the header
#
echo "    Testing the ODBC extension<br>\n";

#
# Test the ODBC functions
#
$ctx = odbc_connect ("NAVDemo", "", "");
$cur = odbc_exec ($ctx, "select c_custkey, c_name from customer");
odbc_result_all ($cur, "border=1 align='center'");
$rc = odbc_free_result ($cur);
odbc_close ($ctx);

?>

```

---

## PHP\_OPENVMS.PHP

```
<?php

#
# Load the OpenVMS extension if needed
#
if (! extension_loaded ("openvms"))
    dl ("php_openvms");

#
# Display the header
#
echo "    Testing the OpenVMS extension<br>\n";

#
# Allow only errors to be reported
#
error_reporting (E_ERROR);

#
# Test the OpenVMS convert filename function
#
# openvms_cvt_filename (func_code, file_name)
#
# func_codes:
#     OPENVMS_CVT_VMS_TO_UNIX          Convert vms filespec to unix filespec
#     OPENVMS_CVT_UNIX_TO_VMS        Convert unix filespec to vms filespec
#
$VmsFn = "PHP_ROOT:[SCRIPTS]PHP_OPENVMS.PHP";
$UnixFn = openvms_cvt_filename (OPENVMS_CVT_VMS_TO_UNIX, $VmsFn);
if ($UnixFn === FALSE)
    echo "openvms_cvt_filename (OPENVMS_CVT_VMS_TO_UNIX, \"$VmsFn\") = " .
openvms_message (openvms_status ()) . "<br>\n";
else
    echo "openvms_cvt_filename (OPENVMS_CVT_VMS_TO_UNIX, \"$VmsFn\") =
$UnixFn<br>\n";

#
# Test the OpenVMS getdvi function
#
# openvms_getdvi (item_code [,device_name])
#
# item_codes:
#     <item_code>                Any Item code supported by F$GETDVI
#     "?"                        List of supported item codes
# device_name:                    Defaults to "TT"
#
$item = "DISPLAY_DEVNAM";
$val = openvms_getdvi ($item);
if ($val === FALSE)
    echo "openvms_getdvi (\"$item\") = " . openvms_message (openvms_status ()) .
"<br>\n";
else
    echo "openvms_getdvi (\"$item\") = $val<br>\n";

#
# Test the OpenVMS getjpi function
#
# openvms_getjpi (item_code [,proc_name][,pid])
#
# item_codes:
#     <item_code>                Any Item code supported by F$GETJPI
#     "?"                        List of supported item codes
# proc_name:                      Any process name
# pid:                            Any process ID or -1 wild card
#
$item = "LAST_LOGIN_I";
$val = openvms_getjpi ($item);
```

```

if ($val === FALSE)
    echo "openvms_getjpi (\">$item\ ") = " . openvms_message (openvms_status ()) .
"<br>\n";
else
    echo "openvms_getjpi (\ \"$item\ ") = $val<br>\n";

#
# Test the OpenVMS getsyi function
#
# openvms_getsyi (item_code [,node_name][,csid])
#
# item_codes:
#     <item_code>           Any Item code supported by F$GETSYI
#     "?"                   List of supported item codes
# node_name:
#     csid:                 Any node name
#                         Any cluster system ID or -1 wild card
#
$item = "BOOTTIME";
$val = openvms_getsyi ($item, "", 0);
if ($val === FALSE)
    echo "openvms_getsyi (\ \"$item\ ") = " . openvms_message (openvms_status ()) .
"<br>\n";
else
    echo "openvms_getsyi (\ \"$item\ ") = $val<br>\n";

#
# Test the OpenVMS time function
#
# openvms_time ([millisecond_time])
#
$val = openvms_time ();
if ($val === FALSE)
    echo "openvms_time () = " . openvms_message (openvms_status ()) . "<br>\n";
else
    echo "openvms_time () = $val<br>\n";

#
# Test the OpenVMS uptime function
#
# openvms_uptime ()
#
$uptime = openvms_uptime ();
if ($uptime === FALSE)
    echo "openvms_uptime () = " . openvms_message (openvms_status ()) . "<br>\n";
else
    echo "openvms_uptime () = $uptime<br>\n";

echo "<br>\n";

#
# Show the cluster info
#
ShowCluster ();

#
# Show the system info
#
ShowSystem ();

#
# Show Cluster
#
function ShowCluster ()
{

$SystemId = openvms_getsyi ("SCSSYSTEMID");
$NodeName = openvms_getsyi ("NODENAME");
$Time = strtok (openvms_time (), ".");

echo "<pre>\n";
$hdr = "View of Cluster from system ID $SystemId  node: $NodeName";

```



```

$pad = str_repeat (" ", 79 - (strlen ($hdr) + strlen ($Time)));
echo $hdr . $pad . $Time . "\n";
echo "+-----+\n";
echo "|          SYSTEMS          | MEMBERS |\n";
echo "|-----+-----|\n";
echo "|   NODE   | SOFTWARE | STATUS |\n";
echo "|-----+-----+-----|\n";

$ctx = -1;
while (1)
{
    $csid = openvms_getsyi ("NODE_CSID", "", &$ctx);
    if ($csid === FALSE)
    {
        $status = openvms_status ();
        if ($status != 2560)
            echo openvms_message (openvms_status ()) . "<br>\n";
        break;
    }
    $nodeName = str_pad (openvms_getsyi ("NODENAME", "", $csid), 6, " ",
STR_PAD_RIGHT);
    $swtype = openvms_getsyi ("NODE_SWTYPE", "", $csid);
    $swvers = openvms_getsyi ("NODE_SWVERS", "", $csid);
    $software = str_pad ($swtype . $swvers, 8, " ", STR_PAD_RIGHT);
    if (strcasecmp (openvms_getsyi ("CLUSTER_MEMBER", "", $csid), "TRUE") == 0)
        $status = "MEMBER";
    else
        $status = " ";
    echo "| $nodeName | $software | $status |\n";
}

if (openvms_getsyi ("CLUSTER_NODES") == 0)
{
    $nodeName = str_pad (openvms_getsyi ("NODENAME"), 6, " ", STR_PAD_RIGHT);
    $swtype = openvms_getsyi ("NODE_SWTYPE", "", $csid);
    $swvers = openvms_getsyi ("NODE_SWVERS", "", $csid);
    $software = str_pad ($swtype . $swvers, 8, " ", STR_PAD_RIGHT);
    if (strcasecmp (openvms_getsyi ("CLUSTER_MEMBER", "", $csid), "TRUE") == 0)
        $status = "MEMBER";
    else
        $status = " ";
    echo "| $nodeName | $software | $status |\n";
}

echo "+-----+\n";
echo "</pre>\n";

}

#
# Show System (Requires World Privilege)
#
function ShowSystem ()
{
    $VmsVer = trim (openvms_getsyi ("VERSION"));
    $nodeName = openvms_getsyi ("NODENAME");
    $UpTime = trim (openvms_uptime ());
    $Time = openvms_time ();

    echo "<pre>\n";
    echo "OpenVMS $VmsVer on node $nodeName $Time Uptime $UpTime\n";
    echo "  Pid   Process Name   State Pri   I/O     CPU     Page flts  Pages\n";
    $ctx = -1;
    while (1)
    {
        $pid = openvms_getjpi ("PID", "", &$ctx);
        if ($pid === FALSE)
        {
            $status = openvms_status ();
            if ($status != 2472)

```

```

        echo openvms_message (openvms_status ()) . "<br>\n";
    break;
}
$prcpid = str_pad ($pid, 8, " ", STR_PAD_RIGHT);
$prcnam = str_pad (openvms_getjpi ("PRCNAM", "", $pid), 15, " ", STR_PAD_RIGHT);
$state = str_pad (openvms_getjpi ("STATE", "", $pid), 5, " ", STR_PAD_RIGHT);
$pri = str_pad (openvms_getjpi ("PRI", "", $pid), 3, " ", STR_PAD_LEFT);
$io = openvms_getjpi ("DIRIO", "", $pid) + openvms_getjpi ("BUFIO", "", $pid);
$io = str_pad ($io, 9, " ", STR_PAD_LEFT);
$cputim = openvms_time (openvms_getjpi ("CPUTIM", "", $pid));
$pagflts = str_pad (openvms_getjpi ("PAGEFLTS", "", $pid), 9, " ", STR_PAD_LEFT);
$pages = openvms_getjpi ("GPGCNT", "", $pid) + openvms_getjpi ("PPGCNT", "",
$pid);
$pages = $pages / (openvms_getsyi ("PAGE_SIZE") / 512);
$pages = str_pad ($pages, 6, " ", STR_PAD_LEFT);
$multithread = openvms_getjpi ("MULTITHREAD", "", $pid);
$owner = openvms_getjpi ("OWNER", "", $pid);
$mode = openvms_getjpi ("MODE", "", $pid);
if ($multithread >= 1)
    $sts = "M";
else
    $sts = " ";
if ($owner != 0)
    $sts .= "S";
else
if (strcasecmp ($mode, "NETWORK") == 0)
    $sts .= "N";
else
if (strcasecmp ($mode, "BATCH") == 0)
    $sts .= "B";
else
    $sts .= " ";
echo "$prcpid $prcnam $state $pri$io$cputim $pagflts $pages $sts\n";
}
echo "</pre>\n";
}
?>

```

---

## Release Notes

This section contains notes on the current release of CSWS\_PHP.

- PHP\$ logical names changed to APACHE\$ in Version 1.2

In *CSWS\_PHP for HP Secure Web Server* Version 1.2, the prefix of the PHP\$ logical names was changed to APACHE\$. This change allows other *Secure Web Server*-based scripting languages (such as Perl) to use the database connectivity features. The changed logical names are as follows:

| Old Logical Name<br>(V1.0 and V1.1) | New Logical Name<br>(V1.2) |
|-------------------------------------|----------------------------|
| PHP\$ODBC_SHR                       | APACHE\$ODBC_SHR           |
| PHP\$ODBC_PFX                       | APACHE\$ODBC_PFX           |
| PHP\$OCI_SHR                        | APACHE\$OCI_SHR            |
| PHP\$OCI_PFX                        | APACHE\$OCI_PFX            |

- Add logical names to PHP\_SETUP.COM for extended file name support

If you are using CSWS\_PHP alone or with *HP Secure Web Server* Version 1.3, and you have ODS-5 files with extended file names such as grab\_globals.lib.php (which has multiple dots), the files will not be processed correctly.

To solve this problem, add the following logical definitions to the end of PHP\_SETUP.COM located in APACHE\$COMMON:[000000]:

```
$ DEFINE /NoLog DECC$EFS_CASE_PRESERVE ENABLED
$ DEFINE /NoLog DECC$EFS_CASE_SPECIAL ENABLED
$ DEFINE /NoLog DECC$EFS_CHARSET ENABLED
$ DEFINE /NoLog DECC$FILE_SHARING ENABLED
```

Using the commands SET PROCESS/PARSE=EXTENDED and SET PROCESS/PARSE=EXTENDED/CASE=SENSITIVE does not work in this situation.

- Configuring CSWS\_PHP **not** required

During the installation, the file PHP\_SETUP.COM is added to the APACHE\$ROOT directory, and an include for MOD\_PHP.CONF is added to the end of HTTPD.CONF. When you start the *Secure Web Server*, PHP\_SETUP.COM is run and PHP is loaded into the server. You do not need to configure CSWS\_PHP.

- PHP **DNS** functions supported only with TCP/IP Services for OpenVMS

This version of CSWS\_PHP supports the **CHECKDNSRR** and **GETMXRR** functions only on systems using HP TCP/IP Services for OpenVMS. These functions may be supported with other TCP/IP products in a future CSWS\_PHP kit.

- PHP **LINK** functions not supported

This version of CSWS\_PHP does not support the **LINK**, **LINKINFO**, **SYMLINK**, and **READLINK** functions. These functions may be supported in a future CSWS\_PHP kit.

-- end of file --