# Set IBM i PASE environment variables for php-cli

## Issue

Sometimes running a PHP script using the php-cli command produces a different result than running the same script in a browser.  Examples include showing different characters in the output or running out of memory.  This can be caused by having different environment variables set for PASE.  Environment variables for PHP scripts running in the browser are set in the FastCGI configuration.  For php-cli, we can set environment variables in the php-cli shell script.  This article tells how to set PASE environment variables in the php-cli shell script.

**Note: These changes assume you run your script using QP2SHELL**

The discussed environment variables apply to PASE.  If you run your php-cli script using QSH or QSHELL, you should instead use QP2SHELL.

## Environment

Any version of Zend Server, running on any supported version of IBM i.

## Resolution

**Note: The php-cli shell script can be replaced by updates, so you may need to redo this change.**

The php-cli shell script is usually updated with each Zend Server upgrade, so it is important to keep track of changes to this file.  You will likely need to reapply these changes every time you upgrade Zend Server.

In this example, we will set the LANG variable to 'C', to match the default setting in the FastCGI configuration as distributed by Zend in the Zend Server for IBM i package.  This was used by a customer in Europe who was seeing the comma used as the decimal separator rather than the period, only when using php-cli to run their script.  Setting LANG=C allowed them to keep using the character set with needed expanded characters (å, ø, and æ in this case) while still using the non-European style decimal separator.

php-cli is a shell script, so we can use the export command in this script to set environment variables.

**Note: This requires editing the shell script, which requires the Unix/Linux style end-of-line.**

Please use a Unix/Linux aware editor like Zend Studio or Notepad ++ to edit the file. **Do not use Notepad**, it will convert the end-of-line to Windows style, and php-cli will stop working, with a 'not found' error.  Also, please **BACK UP THIS FILE** before changing it, so you can restore it if you can't change it back with the editor for any reason.

### Zend Server versions 2020.x, 2021.x

The file to change is: **/usr/local/zendphp74/bin/php-cli**

After you BACK UP this file, open it in a Unix safe editor. It will look like this:

```
#!/bin/sh
 if [ -f /usr/local/zendphp74/etc/zce.rc ];then
 . /usr/local/zendphp74/etc/zce.rc
 else
 echo '/usr/local/zendphp74/etc/zce.rc doesn't exist!'
 exit 1;
 fi
 LIBPATH=$ZCE_PREFIX/lib
 $ZCE_PREFIX/bin/php.bin '$@'
```

Add the export just before the last line, so now it looks like this (red highlight added for emphasis, you don't need to make the line red):

```
#!/bin/sh
 if [ -f /usr/local/zendphp74/etc/zce.rc ];then
 . /usr/local/zendphp74/etc/zce.rc
 else
 echo '/usr/local/zendphp74/etc/zce.rc doesn't exist!'
 exit 1;
 fi
 LIBPATH=$ZCE_PREFIX/lib
 export LANG=C
 $ZCE_PREFIX/bin/php.bin '$@'
```

Save the change.  Since this is php-cli, there is no need to restart Apache or anything.  Just run your script with php-cli to see the changed behavior.

## Zend Server versions 9.1.x, 2018.0.x and 2019.0.x

The file to change is: **/usr/local/zendphp7/bin/php-cli**

After you BACK UP this file, open it in a Unix safe editor. It will look like this:

```
#!/bin/sh
 if [ -f /usr/local/zendphp7/etc/zce.rc ];then
 . /usr/local/zendphp7/etc/zce.rc
 else
 echo '/usr/local/zendphp7/etc/zce.rc doesn't exist!'
 exit 1;
 fi
 LIBPATH=$ZCE_PREFIX/lib
 $ZCE_PREFIX/bin/php.bin '$@'
```

Add the export just before the last line, so now it looks like this (red highlight added for emphasis, you don't need to make the line red):

```
#!/bin/sh
 if [ -f /usr/local/zendphp7/etc/zce.rc ];then
 . /usr/local/zendphp7/etc/zce.rc
 else
 echo '/usr/local/zendphp7/etc/zce.rc doesn't exist!'
 exit 1;
 fi
 LIBPATH=$ZCE_PREFIX/lib
 export LANG=C
 $ZCE_PREFIX/bin/php.bin '$@'
```

Save the change.  Since this is php-cli, there is no need to restart Apache or anything.  Just run your script with php-cli to see the changed behavior.

## Zend Server Versions 6 - 8.5.x

The file to change is:  **/usr/local/zendsvr6/bin/php-cli**

```
#!/bin/sh
 if [ -f /usr/local/zendsvr6/etc/zce.rc ];then
 . /usr/local/zendsvr6/etc/zce.rc
 else
 echo '/usr/local/zendsvr6/etc/zce.rc doesn't exist!'
 exit 1;
 fi
 LIBPATH=$ZCE_PREFIX/lib
 $ZCE_PREFIX/bin/php.bin '$@'
```

Add the export just before the last line, so now it looks like this (red highlight added for emphasis, you don't need to make the line red):

```
#!/bin/sh
if [ -f /usr/local/zendsvr6/etc/zce.rc ];then
. /usr/local/zendsvr6/etc/zce.rc
else
echo '/usr/local/zendsvr6/etc/zce.rc doesn't exist!'
exit 1;
fi
LIBPATH=$ZCE_PREFIX/lib
export LANG=C
$ZCE_PREFIX/bin/php.bin '$@'
```

## Details

Remember to run your php-cli script in PASE in order for these environment variables to work.  You could run them in the PASE terminal QP2TERM, or in batch using QP2SHELL or QP2SHELL2, or in the SSH terminal provided by utilities like PuTTy or Zend Studio.  You should not run them using QSH or STRQSH.  The native mode shell will not recognize the PASE environment variables.

The change made in the above example addresses a change in the language environment variable that is the default we use in the FastCGI configuration.  The LANG=C setting sets PASE to use 7 bit ASCII as the locale, which does not seem to interfere with other encodings, but does seem to help prevent some other unintended locale settings, like using a comma instead of a period for decimal separators.

Here are a couple of other defaults we use in the FastCGI configuration:

CCSID=1208

This one is useful for the ibm_db2 extension.  Using CCSID 1208 (UTF-8) in this environment variable lets ibm_db2 know you would like certain automatic processing to happen to insure better translation from EBCDIC data.  Try this if you are getting unexpected characters showing up when reading or writing data to your DB2 tables.

LDR_CNTRL=MAXDATA=0x40000000

This increases the maximum allowed memory for a PASE process.  If your PHP script runs OK in the browser but runs out of memory in php-cli, you can try exporting this setting.  Valid values for this, and the corresponding memory sizes:

| Value | Added segments | Memory limit |
|---|---|---|
| Unset (default) | 0 | 256 MB |
| LDR_CNTRL=MAXDATA=0x10000000 | 1 | 512 MB |
| LDR_CNTRL=MAXDATA=0x20000000 | 2 | 768 MB |
| LDR_CNTRL=MAXDATA=0x30000000 | 3 | 1 GB |
| LDR_CNTRL=MAXDATA=0x40000000 | 4 | 1.25 GB |
| LDR_CNTRL=MAXDATA=0x50000000 | 5 | 1.5 GB |
| LDR_CNTRL=MAXDATA=0x60000000 | 6 | 1.75 GB |
| LDR_CNTRL=MAXDATA=0x70000000 | 7 | 2 GB |
| LDR_CNTRL=MAXDATA=0x80000000 | 8 | 2.25 GB |

Remember to "export" to set your environment variables.  You could use these, for example:

```
export LDR_CNTRL=MAXDATA=0x40000000

export CCSID=1208

export LANG=C
```