PHP 7 support

Source file: doc/eng-us/bc/5.90/php7.md Last changed 2026-11-04
https://doc.exponential.earth/exphelp/viewer/bc/5.90/php7.md

PHP 7.0 support

For the 2017.10 release, Exponential received changes all over the code base, switching object constructor methods to PHP 5 style.

The reason is to achieve full PHP 7.0 support by avoiding the deprecation warnings when using PHP 4 style constructors.

Care has been taken to keep around compatibility functions in all known cases to avoid fatal errors for custom extensions, however to avoid warnings you might need to adapt your code as well.

Common cases are classes extending `eZPersistentObject` or `eZDataType`.

Here is an example of how you should adapt your code for the constructor change:

Diff

```
diff --git a/classes/ezfindresultnode.php b/classes/ezfindresultnode.php
index fafca310..b6462159 100644
--- a/classes/ezfindresultnode.php
+++ b/classes/ezfindresultnode.php
@@ -12,7 +12,7 @@ class eZFindResultNode extends eZContentObjectTreeNode
     */
     function eZFindResultNode( $rows = array() )
     {
-        $this->eZContentObjectTreeNode( $rows );
+        parent::__construct( $rows );
         if ( isset( $rows['id'] ) )
         {
             $this->ContentObjectID = $rows['id'];
```

For best results you should also consider changing your own code to use PHP 5 style constructors. In the example above that would mean renaming `function eZFindResultNode` to `function __construct` and, if you think that other code might exist which extends the `eZFindResultNode` class, add back a courtesy `eZFindResultNode` function that does nothing but call `__construct`

Further reading:

- http://php.net/manual/en/language.oop5.decon.php
- https://www.php.net/manual/en/migration70.incompatible.php
- https://www.php.net/manual/en/migration71.incompatible.php

PHP 7.2 support

Starting with the 2019.03 release, issues happening on PHP 7.2 and PHP 7.3 have been fixed, but in your own code you'll also need to handle some of those.

Most notable is the `Warn when counting non-countable types` change, added in PHP 7.2.

To handle this across all supported PHP versions, we introduced use of symfony/polyfill-php73 package, witch backports PHP 7.3's function is_countable.

Here is an example of changes you might need to apply in your own code to work around that:

Diff

```
diff --git a/kernel/common/eztemplatedesignresource.php b/kernel/common/eztemplatedesignresource.
php
index b0fc28faa9a..9b8ca2a8d94 100644
--- a/kernel/common/eztemplatedesignresource.php
+++ b/kernel/common/eztemplatedesignresource.php
@@ -86,7 +86,7 @@ function templateNodeTransformation( $functionName, &$node,
                 $matchCount = 0;
                 foreach ( $customMatchList as $customMatch )
                 {
-                    $matchConditionCount = count( $customMatch['conditions'] );
+                    $matchConditionCount = is_countable( $customMatch['conditions'] ) ? count( $
customMatch['conditions'] ) : 0;
                     $code = '';
                     if ( $matchCount > 0 )
                     {
```

Further reading:

- https://www.php.net/manual/en/migration72.incompatible.php
- https://www.php.net/manual/en/migration73.incompatible.php

Copyright (C) 1999 - 2014 eZ Systems AS and contributors
This page was inherited from the documentation of the original software and is not covered by the GNU Free Documentation License. It is distributed under the GNU General Public License, version 2 (https://www.gnu.org/licenses/old-licenses/gpl-2.0.html).