How to add Zend Framework 2 to your include_path

Issue

This article tells how to set up the include_path to include Zend framework 2 in your project.

Environment

Zend Server running on all supported OS versions.

Resolution

Zend Framework 2 and other libraries are collections of PHP scripts. They can be included in your application using the PHP directive include_path. This can be set globally in the php.ini file to make the library available to every script running on your site, can be set for a directory to be available to every script in that directory, or can be set within a script using the ini_set() directive.

For Zend Server versions 2020.x and 2021.x and higher

Globally

The first method is to set it globally. You will want to set it globally if every script on your site needs access to Framework 2. Otherwise, use one of the other methods to include Framework 2, so that scripts that do not need it will not search for it. To set it globally, you can go into the Zend Server UI, and navigate to Configurations -> PHP -> Paths and Directories. Find the include_path directive, add a colon to the end, and then add the path after that. For example, you could change this:

```
.:/usr/local/zendphp74/var/libraries/Zend_Framework_1/default/library:/usr/local/zendphp74/share/pear:/usr
/local/zendphp74/var/libraries/PHP_Toolkit_for_IBMI_i/default/library:/usr/local/zendphp74/share/ToolkitApi
```

to this:

```
.:/usr/local/zendphp74/var/libraries/Zend_Framework_1/default/library:/usr/local/zendphp74/share/pear:/usr
/local/zendphp74/var/libraries/PHP_Toolkit_for_IBMI_i/default/library:/usr/local/zendphp74/share/ToolkitApi:
/usr/local/zendphp74/share/ZendFramework2/library
```

Per directory

The second method is to set up a directory so that every script in that directory can use the Framework 2 library. You can create a file called .user.ini in that directory. (Notice the file name starts with a period. You might read that as dot-user-dot-ini.) In that file you can specify the include path:

```
include_path = '.:/usr/local/zendphp74/var/libraries/Zend_Framework_1/default/library:/usr/local/zendphp74
/share/pear:/usr/local/zendphp74/var/libraries/PHP_Toolkit_for_IBMI_i/default/library:/usr/local/zendphp74
/share/ToolkitApi:/usr/local/zendphp74/share/ZendFramework2/library'
```

If you don't need all that other stuff in your include path, you might do it simpler:

```
include_path = '.:/usr/local/zendphp74/share/ZendFramework2/library'
```

Per script

The final method is to use ini_set() to set your include_path in your script. To append ZF 2 to your existing path:

```
ini_set('include_path',ini_get('include_path').':/usr/local/zendphp74/share/ZendFramework2/library');
```

Or if you don't need all the others in your include_path:

```
ini_set('include_path', '/usr/local/zendphp74/share/ZendFramework2/library');
```

Per script with zend_deployment_library_path()

If Zend Deployment is managing your Zend Framework 2 library, there is a function zend_deployment_library_path() that can be used to specify the path. Deployment can manage multiple versions of a library, with one version being the default. The zend_deployment_library_path() function allows you to specify which version of the library you would like to use. You can specify a particular version for testing a new version of the library, or perhaps if you have an application that is only compatible with a prior version. There is a discussion of the zend_deployment_library_path() function usage in the online documentation:

Using Libraries

Here is a short script that demonstrates how this function might be used. The line highlighted with a red font shows how to append the Zend Framework 2 path to the current path.

```
<?php
// Demonstrate usage for zend_deployment_library_path with Zend Framework 2
echo "<pre>"; // I like a monospaced font for stuff like this

// Store the current path
$initial_path = ini_get(include_path);
echo "The current path is:<br>$initial_path<br><br>";

// Test for existence of the Framework 2 default path
$zf2Path = zend_deployment_library_path('Zend Framework 2');

if (!$zf2Path) {
    echo "Zend Framework 2 default path not found.<br><br>";
    die; //nothing more to do if ZF2 is not deployed
} else {
    echo "The default path for Zend Framework 2 is:<br>$zf2Path<br><br>";

    // Here is one method to append the ZF2 path to the initial path, using the functions
    ini_set(include_path, ini_get(include_path) . ':' . zend_deployment_library_path('Zend Framework 2'));
    echo "The new path with the Zend Framework 2 default path appended is:<br>" . ini_get(include_path) .
"<br><br>";

    // We could also just use the variables we have already loaded.
    ini_set(include_path, $initial_path . ':' . $zf2Path);
    echo "The new path with the Zend Framework 2 default path appended is still:<br>" . ini_get(include_path)
. "<br><br>";
}

// Demonstrate that spelling must be exact
$zf2Path = zend_deployment_library_path('Zned Framewark 2');

if (!$zf2Path) {
    echo "Zned Framewark 2 path not found.  Check your spelling.<br><br>";

}

// Get the path for version 2.2.5
$zf2Path = zend_deployment_library_path('Zend Framework 2', '2.2.5');

if (!$zf2Path) {
    echo "Zend Framework 2 Version 2.2.5 path not found.<br><br><br><br>";
} else {
    echo "The path for Zend Framework 2 Version 2.2.5 is:<br>$zf2Path";
}
?>
```

Here is the output from this script, running on Zend Server with Zend Framework 2 version 2.3.0 deployed as the default and version 2.2.5 deployed as an earlier version.

```
The current path is:
.:/usr/local/zendphp74/share/ZendFramework/library:/usr/local/zendphp7/share/pear:/usr/local/zendphp7/share
/ToolkitApi

The default path for Zend Framework 2 is:
/usr/local/zendphp74/var/libraries/Zend_Framework_2/2.3.0/library

The new path with the Zend Framework 2 default path appended is:
.:/usr/local/zendphp74/share/ZendFramework/library:/usr/local/zendphp74/share/pear:/usr/local/zendphp74/share
/ToolkitApi:/usr/local/zendphp74/var/libraries/Zend_Framework_2/2.3.0/library

The new path with the Zend Framework 2 default path appended is still:
.:/usr/local/zendphp7/share/ZendFramework/library:/usr/local/zendphp74/share/pear:/usr/local/zendphp74/share
/ToolkitApi:/usr/local/zendphp74/var/libraries/Zend_Framework_2/2.3.0/library

Zned Framewark 2 path not found.  Check your spelling.

The path for Zend Framework 2 Version 2.2.5 is:
/usr/local/zendphp74/var/libraries/Zend_Framework_2/2.2.5/library
```

For Zend Server versions 9.1.x, 2018.x and 2019.0x

Globally

The first method is to set it globally. You will want to set it globally if every script on your site needs access to Framework 2. Otherwise, use one of the other methods to include Framework 2, so that scripts that do not need it will not search for it. To set it globally, you can go into the Zend Server UI, and navigate to Configurations -> PHP -> Paths and Directories. Find the include_path directive, add a colon to the end, and then add the path after that. For example, you could change this:

```
.:/usr/local/zendphp7/var/libraries/Zend_Framework_1/default/library:/usr/local/zendphp7/share/pear:/usr/local
/zendphp7/var/libraries/PHP_Toolkit_for_IBMI_i/default/library:/usr/local/zendphp7/share/ToolkitApi
```

to this:

```
.:/usr/local/zendphp7/var/libraries/Zend_Framework_1/default/library:/usr/local/zendphp7/share/pear:/usr/local
/zendphp7/var/libraries/PHP_Toolkit_for_IBMI_i/default/library:/usr/local/zendphp7/share/ToolkitApi:/usr/local
/zendphp7/share/ZendFramework2/library
```

Per directory

The second method is to set up a directory so that every script in that directory can use the Framework 2 library. You can create a file called .user.ini in that directory. (Notice the file name starts with a period. You might read that as dot-user-dot-ini.) In that file you can specify the include path:

```
include_path = '.:/usr/local/zendphp7/var/libraries/Zend_Framework_1/default/library:/usr/local/zendphp7/share
/pear:/usr/local/zendphp7/var/libraries/PHP_Toolkit_for_IBMI_i/default/library:/usr/local/zendphp7/share
/ToolkitApi:/usr/local/zendphp7/share/ZendFramework2/library'
```

If you don't need all that other stuff in your include path, you might do it simpler:

```
include_path = '.:/usr/local/zendphp7/share/ZendFramework2/library'
```

Per script

The final method is to use ini_set() to set your include_path in your script. To append ZF 2 to your existing path:

```
ini_set('include_path',ini_get('include_path').':/usr/local/zendphp7/share/ZendFramework2/library');
```

Or if you don't need all the others in your include_path:

```
ini_set('include_path', '/usr/local/zendphp7/share/ZendFramework2/library');
```

Per script with zend_deployment_library_path()

If Zend Deployment is managing your Zend Framework 2 library, there is a function zend_deployment_library_path() that can be used to specify the path. Deployment can manage multiple versions of a library, with one version being the default. The zend_deployment_library_path() function allows you to specify which version of the library you would like to use. You can specify a particular version for testing a new version of the library, or perhaps if you have an application that is only compatible with a prior version. There is a discussion of the zend_deployment_library_path() function usage in the online documentation:

Using Libraries

Here is a short script that demonstrates how this function might be used. The line highlighted with a red font shows how to append the Zend Framework 2 path to the current path.

```
<?php
// Demonstrate usage for zend_deployment_library_path with Zend Framework 2
echo "<pre>"; // I like a monospaced font for stuff like this

// Store the current path
$initial_path = ini_get(include_path);
echo "The current path is:<br>$initial_path<br><br>";

// Test for existence of the Framework 2 default path
$zf2Path = zend_deployment_library_path('Zend Framework 2');

if (!$zf2Path) {
    echo "Zend Framework 2 default path not found.<br><br>";
    die; //nothing more to do if ZF2 is not deployed
} else {
    echo "The default path for Zend Framework 2 is:<br>$zf2Path<br><br>";

    // Here is one method to append the ZF2 path to the initial path, using the functions
    ini_set(include_path, ini_get(include_path) . ':' . zend_deployment_library_path('Zend Framework 2'));
    echo "The new path with the Zend Framework 2 default path appended is:<br>" . ini_get(include_path) .
"<br><br>";

    // We could also just use the variables we have already loaded.
    ini_set(include_path, $initial_path . ':' . $zf2Path);
    echo "The new path with the Zend Framework 2 default path appended is still:<br>" . ini_get(include_path)
. "<br><br>";
}

// Demonstrate that spelling must be exact
$zf2Path = zend_deployment_library_path('Zned Framewark 2');

if (!$zf2Path) {
    echo "Zned Framewark 2 path not found.  Check your spelling.<br><br>";

}

// Get the path for version 2.2.5
$zf2Path = zend_deployment_library_path('Zend Framework 2', '2.2.5');

if (!$zf2Path) {
    echo "Zend Framework 2 Version 2.2.5 path not found.<br><br><br><br>";
} else {
    echo "The path for Zend Framework 2 Version 2.2.5 is:<br>$zf2Path";
}
?>
```

Here is the output from this script, running on Zend Server with Zend Framework 2 version 2.3.0 deployed as the default and version 2.2.5 deployed as an earlier version.

```
The current path is:
.:/usr/local/zendphp7/share/ZendFramework/library:/usr/local/zendphp7/share/pear:/usr/local/zendphp7/share
/ToolkitApi

The default path for Zend Framework 2 is:
/usr/local/zendphp7/var/libraries/Zend_Framework_2/2.3.0/library

The new path with the Zend Framework 2 default path appended is:
.:/usr/local/zendphp7/share/ZendFramework/library:/usr/local/zendphp7/share/pear:/usr/local/zendphp7/share
/ToolkitApi:/usr/local/zendphp7/var/libraries/Zend_Framework_2/2.3.0/library

The new path with the Zend Framework 2 default path appended is still:
.:/usr/local/zendphp7/share/ZendFramework/library:/usr/local/zendphp7/share/pear:/usr/local/zendphp7/share
/ToolkitApi:/usr/local/zendphp7/var/libraries/Zend_Framework_2/2.3.0/library

Zned Framewark 2 path not found.  Check your spelling.

The path for Zend Framework 2 Version 2.2.5 is:
/usr/local/zendphp7/var/libraries/Zend_Framework_2/2.2.5/library
```

For Zend Server versions 6 - 8.5.x

Globally

The first method is to set it globally. You will want to set it globally if every script on your site needs access to Framework 2. Otherwise, use one of the other methods to include Framework 2, so that scripts that do not need it will not search for it. To set it globally, you can go into the Zend Server UI, and navigate to Configurations -> PHP -> Paths and Directories. Find the include_path directive, add a colon to the end, and then add the path after that. For example, you could change this:

```
.:/usr/local/zendsvr6/var/libraries/Zend_Framework_1/default/library:/usr/local/zendsvr6/share/pear:/usr/local
/zendsvr6/var/libraries/PHP_Toolkit_for_IBMI_i/default/library:/usr/local/zendsvr6/share/ToolkitApi
```

to this:

```
.:/usr/local/zendsvr6/var/libraries/Zend_Framework_1/default/library:/usr/local/zendsvr6/share/pear:/usr/local
/zendsvr6/var/libraries/PHP_Toolkit_for_IBMI_i/default/library:/usr/local/zendsvr6/share/ToolkitApi:/usr/local
/zendsvr6/share/ZendFramework2/library
```

Per directory

The second method is to set up a directory so that every script in that directory can use the Framework 2 library. You can create a file called .user.ini in that directory. (Notice the file name starts with a period. You might read that as dot-user-dot-ini.) In that file you can specify the include path:

```
include_path = '.:/usr/local/zendsvr6/var/libraries/Zend_Framework_1/default/library:/usr/local/zendsvr6/share
/pear:/usr/local/zendsvr6/var/libraries/PHP_Toolkit_for_IBMI_i/default/library:/usr/local/zendsvr6/share
/ToolkitApi:/usr/local/zendsvr6/share/ZendFramework2/library'
```

If you don't need all that other stuff in your include path, you might do it simpler:

```
include_path = '.:/usr/local/zendsvr6/share/ZendFramework2/library'
```

Per script

The final method is to use ini_set() to set your include_path in your script. To append ZF 2 to your existing path:

```
ini_set('include_path',ini_get('include_path').':/usr/local/zendsvr6/share/ZendFramework2/library');
```

Or if you don't need all the others in your include_path:

```
ini_set('include_path', '/usr/local/zendsvr6/share/ZendFramework2/library');
```

Per script with zend_deployment_library_path()

If Zend Deployment is managing your Zend Framework 2 library, there is a function zend_deployment_library_path() that can be used to specify the path. Deployment can manage multiple versions of a library, with one version being the default. The zend_deployment_library_path() function allows you to specify which version of the library you would like to use. You can specify a particular version for testing a new version of the library, or perhaps if you have an application that is only compatible with a prior version. There is a discussion of the zend_deployment_library_path() function usage in the online documentation:

Using Libraries

Here is a short script that demonstrates how this function might be used. The line highlighted with a red font shows how to append the Zend Framework 2 path to the current path.

```
<?php
// Demonstrate usage for zend_deployment_library_path with Zend Framework 2
echo "<pre>"; // I like a monospaced font for stuff like this

// Store the current path
$initial_path = ini_get(include_path);
echo "The current path is:<br>$initial_path<br><br>";

// Test for existence of the Framework 2 default path
$zf2Path = zend_deployment_library_path('Zend Framework 2');

if (!$zf2Path) {
    echo "Zend Framework 2 default path not found.<br><br>";
    die; //nothing more to do if ZF2 is not deployed
} else {
    echo "The default path for Zend Framework 2 is:<br>$zf2Path<br><br>";

    // Here is one method to append the ZF2 path to the initial path, using the functions
    ini_set(include_path, ini_get(include_path) . ':' . zend_deployment_library_path('Zend Framework 2'));
    echo "The new path with the Zend Framework 2 default path appended is:<br>" . ini_get(include_path) .
"<br><br>";

    // We could also just use the variables we have already loaded.
    ini_set(include_path, $initial_path . ':' . $zf2Path);
    echo "The new path with the Zend Framework 2 default path appended is still:<br>" . ini_get(include_path)
. "<br><br>";
}

// Demonstrate that spelling must be exact
$zf2Path = zend_deployment_library_path('Zned Framewark 2');

if (!$zf2Path) {
    echo "Zned Framewark 2 path not found.  Check your spelling.<br><br>";

}

// Get the path for version 2.2.5
$zf2Path = zend_deployment_library_path('Zend Framework 2', '2.2.5');

if (!$zf2Path) {
    echo "Zend Framework 2 Version 2.2.5 path not found.<br><br><br><br>";
} else {
    echo "The path for Zend Framework 2 Version 2.2.5 is:<br>$zf2Path";
}
?>
```

Here is the output from this script, running on Zend Server with Zend Framework 2 version 2.3.0 deployed as the default and version 2.2.5 deployed as an earlier version.

```
The current path is:
.:/usr/local/zendsvr6/share/ZendFramework/library:/usr/local/zendsvr6/share/pear:/usr/local/zendsvr6/share
/ToolkitApi

The default path for Zend Framework 2 is:
/usr/local/zendsvr6/var/libraries/Zend_Framework_2/2.3.0/library

The new path with the Zend Framework 2 default path appended is:
.:/usr/local/zendsvr6/share/ZendFramework/library:/usr/local/zendsvr6/share/pear:/usr/local/zendsvr6/share
/ToolkitApi:/usr/local/zendsvr6/var/libraries/Zend_Framework_2/2.3.0/library

The new path with the Zend Framework 2 default path appended is still:
.:/usr/local/zendsvr6/share/ZendFramework/library:/usr/local/zendsvr6/share/pear:/usr/local/zendsvr6/share
/ToolkitApi:/usr/local/zendsvr6/var/libraries/Zend_Framework_2/2.3.0/library

Zned Framewark 2 path not found.  Check your spelling.

The path for Zend Framework 2 Version 2.2.5 is:
/usr/local/zendsvr6/var/libraries/Zend_Framework_2/2.2.5/library
```