

Zend Server UI pages are partially blank on IBM i

Issue

Several pages in the Zend Server UI may show up as blanks, or tables with headings but no detail. We see this most often on new installations of Zend Server for IBM i, and particularly when installed on smaller developer partitions with very small CPU or Memory Pool allocations.

Note: There may also be a problem with DNS entries that can cause severely slow response in the UI:

[Invalid DNS slows Zend Server on IBM i](#)

Environment

Zend Server for IBM i version 8.5 on any supported version of IBM i operating system. Can be applied to earlier versions 6, 7, or 8, with small changes to the instructions related to differences in the menus.

Resolution

Most pages on the Zend Server UI are loaded by the Zend Server Daemon. If the pages are not loading, the Zend Server Daemon may not be running. If it is running, it may be running PHP processes that time out before they can complete the page load.

The first thing to verify is that the ZSDAEMON jobs are running in the ZENDPHP7/ZENDSVR6 subsystem. Here is a quick way to check whether the Zend Server Daemon is active. From a 5250 command line, logged in with a *SECOFR class user profile:

For versions 9 and higher:

```
go zendphp7/zsvmenu
```

Use option '3. Work with Zend Server subsystems'. Select Subsystem Name ZENDPHP7, and press enter. This will bring up the 'Work with Active Jobs' display. Please verify that the following jobs display:

```
ZSDAEMON QTMHHTTP BCI .0 PGM-watchdog
ZSDAEMON QTMHHTTP BCI 3.8 PGM-zsd
```

The daemon runs at least two jobs, watchdog and zsd. When it is active, it may be running additional jobs for additional processes, but these two should always be present.

If the server daemon is not running, you can start it without having to restart Zend Server:

```
go zenphp7/zsdaemon
```

Use option '1. Start Zend Server Daemon'. Give it a minute to start up, then check the subsystem for the ZSDAEMON jobs to verify the daemon is started.

Note:

If you cannot get the ZSDAEMON jobs started, please do not continue with the next steps.

The next step is to set the max_input_time directive and the max_execution_time directives to wait longer before timing out. Here is how to do it.

First, make sure you can edit files in Navigator, as described in this blog post:

[Edit any file in Navigator](#)

Next, stop Zend Server. From the 5250 command line:

```
go zendphp7/zsmenu
```

Use option '2. End Zend Server Subsystem'.

Now, find file `/usr/local/zendphp7/etc/php.ini` in Navigator. Right click the file and select edit in the context menu. In the editor, select File -> Save As... and save the file with some easy to remember back up name like `php.ini.save`. Please notice that after you save the file, you are no longer editing `php.ini`, you are editing `php.ini.save`, so you should close the editor at this point.

When you close the editor and return to Navigator, please use F5 to refresh the Navigator display. You should now see both `php.ini` and `php.ini.save` in the `/usr/local/zendphp7/etc` directory. So now you have a back up and can continue to edit `php.ini`. Once again, right click `php.ini` and select Edit.

In the editor, select Edit -> Find... and search for `max_execution_time`. You will find it in the Resource limits section, along with `max_input_time`:

```
;;;;;;;;;;
; Resource Limits ;
;;;;;;;;;;
; Maximum execution time of each script, in seconds
; http://php.net/max-execution-time
; Note: This directive is hardcoded to 0 for the CLI SAPI
max_execution_time=30
; Maximum amount of time each script may spend parsing request data. It's a good
; idea to limit this time on productions servers in order to eliminate unexpectedly
; long running scripts.
; Note: This directive is hardcoded to -1 for the CLI SAPI
; Default Value: -1 (Unlimited)
; Development Value: 60 (60 seconds)
; Production Value: 60 (60 seconds)
; http://php.net/max-input-time
max_input_time=60
```

In this example we will set these limits up to 180 and 240 (three minutes and four minutes), which is usually enough to resolve issues with the UI. However, they can be set higher if need be, by just repeating the steps to stop Zend Server, edit the file, and start Zend Server.

Please change `max_execution_time` to 180:

```
max_execution_time=180
```

Please change `max_input_time` to 240:

```
max_input_time=240
```

Please DO NOT make any other changes to this file. Do not add spaces, or press enter, or add a comment. Just change the numbers.

Save the file using File -> Save. You can then close the editor using File -> Exit.

Now, please return to the menu on the 5250 display and use option '1. Start Zend Server Subsystem'.

Please check to make sure everything is started up normally:

[Verify Zend Server for IBM i has started successfully](#)

After a successful start up, please return to the Zend Server UI and see if you can access the pages that had come up blank previously. Please allow the pages some time to come up, as the timeouts have been set to several minutes.

If you are able to now access the pages in the UI, please navigate to the PHP -> Extensions page. Find the "Resource Limits and Tuning" item in the table, and expand it by single clicking it. Find the `max_execution_time` and `max_input_time` directives and set them to match the entries you made manually in the `php.ini` file. Click Save, and restart to save the changes. This will insure that the values you have stored in the Zend Server data base match the values in the `php.ini` file.

For versions 6 - 8.5:

```
go zendsvr6/zsvmenu
```

Use option '3. Work with Zend Server subsystems'. Select Subsystem Name ZENDSRV6, and press enter. This will bring up the 'Work with Active Jobs' display. Please verify that the following jobs display:

```
ZSDAEMON QTMHHTTP BCI .0 PGM-watchdog
ZSDAEMON QTMHHTTP BCI 3.8 PGM-zsd
```

The daemon runs at least two jobs, watchdog and zsd. When it is active, it may be running additional jobs for additional processes, but these two should always be present.

If the server daemon is not running, you can start it without having to restart Zend Server:

```
go zensvr6/zsdaemon
```

Use option '1. Start Zend Server Daemon'. Give it a minute to start up, then check the subsystem for the ZSDAEMON jobs to verify the daemon is started.

Note:

If you cannot get the ZSDAEMON jobs started, please do not continue with the next steps.

The next step is to set the `max_input_time` directive and the `max_execution_time` directives to wait longer before timing out. Here is how to do it.

First, make sure you can edit files in Navigator, as described in this blog post:

[Edit any file in Navigator](#)

Next, stop Zend Server. From the 5250 command line:

```
go zendsvr6/zsmenu
```

Use option '2. End Zend Server Subsystem'.

Now, find file `/usr/local/zendsvr6/etc/php.ini` in Navigator. Right click the file and select edit in the context menu. In the editor, select File -> Save As... and save the file with some easy to remember back up name like `php.ini.save`. Please notice that after you save the file, you are no longer editing `php.ini`, you are editing `php.ini.save`, so you should close the editor at this point.

When you close the editor and return to Navigator, please use F5 to refresh the Navigator display. You should now see both `php.ini` and `php.ini.save` in the `/usr/local/zendsvr6/etc` directory. So now you have a back up and can continue to edit `php.ini`. Once again, right click `php.ini` and select Edit.

In the editor, select Edit -> Find... and search for `max_execution_time`. You will find it in the Resource limits section, along with `max_input_time`:

```
;;;;;;;;;;;;;
; Resource Limits ;
;;;;;;;;;;;;;
; Maximum execution time of each script, in seconds
; http://php.net/max-execution-time
; Note: This directive is hardcoded to 0 for the CLI SAPI
max_execution_time=30
; Maximum amount of time each script may spend parsing request data. It's a good
; idea to limit this time on productions servers in order to eliminate unexpectedly
; long running scripts.
; Note: This directive is hardcoded to -1 for the CLI SAPI
; Default Value: -1 (Unlimited)
; Development Value: 60 (60 seconds)
; Production Value: 60 (60 seconds)
; http://php.net/max-input-time
max_input_time=60
```

In this example we will set these limits up to 180 and 240 (three minutes and four minutes), which is usually enough to resolve issues with the UI. However, they can be set higher if need be, by just repeating the steps to stop Zend Server, edit the file, and start Zend Server.

Please change `max_execution_time` to 180:

```
max_execution_time=180
```

Please change max_input_time to 240:

```
max_input_time=240
```

Please DO NOT make any other changes to this file. Do not add spaces, or press enter, or add a comment. Just change the numbers.

Save the file using File -> Save. You can then close the editor using File -> Exit.

Now, please return to the menu on the 5250 display and use option '1. Start Zend Server Subsystem'.

Please check to make sure everything is started up normally:

[Verify Zend Server for IBM i has started successfully](#)

After a successful start up, please return to the Zend Server UI and see if you can access the pages that had come up blank previously. Please allow the pages some time to come up, as the timeouts have been set to several minutes.

If you are able to now access the pages in the UI, please navigate to the PHP -> Extensions page. Find the "Resource Limits and Tuning" item in the table, and expand it by single clicking it. Find the max_execution_time and max_input_time directives and set them to match the entries you made manually in the php.ini file. Click Save, and restart to save the changes. This will insure that the values you have stored in the Zend Server data base match the values in the php.ini file.