Set error_reporting to control error reporting levels in the php.log file

Issue

The php.log file is filling up with many notices, coding warnings, and deprecated errors. This can make it difficult to find more critical errors, and can cause the log size to grow too quickly.

Environment

This article tells how to set the error_reporting directive using the Zend Server 7 User Interface. The suggested string "E_ALL & ~E_NOTICE & ~E_STRICT & ~E_DEPRECATED" is valid for PHP 5.3 and later.

Resolution

Access the Zend Server User Interface in your web browser at:

http://<your IBM i IP address>:10101

> ⓘ **The User Interface may be at a different port.**
>
> Depending on the version of Zend server and whether it was installed with a different version, your user interface may be at port 10081 or 10091, instead of 10101 as shown in the above example.

Log in if needed. Go to the PHP -> Extensions tab. Expand the 'Error Handling and Logging' category (click on it).

Look at the 'error_reporting' setting. In the default configuration, the value is this:

E_ALL & ~E_DEPRECATED & ~E_STRICT

This will capture all errors to the php.log file, including many warning messages that you may not care about. You can change this value to something more restrictive. One typical string to use would be this:

E_ALL & ~E_NOTICE & ~E_STRICT & ~E_DEPRECATED

After you make the change, the Save button at the top of the table should become enabled. Click the Save button to save your change.

You should see a notification come up in a green box that says the change was successfully changed. This notification goes away after a short time.

Towards the upper right of the page, find the Restart icon. It looks sort of like a circular arrow. It will be orange, indicating a restart is due. You may also see a nearby notification message letting you know the restart is due.

You do not need to restart right away, if the web site is in use. Once there is a time when it will be OK to have the web site down for a couple minutes, clean the Restart icon to restart your web site. Once the restart is complete, the new setting will take effect.

Details

You can see all of the error logging constants with definitions here:

[PHP Error Logging Predefined Constants](#)

Learn more about the error_reporting directive here:

[PHP: Runtime Configuration](#)

As shipped, Zend Server sets error_reporting to E_ALL & ~E_DEPRECATED & ~E_STRICT, which reports everything. This is fine in a development environment, as the developer may be interested in all the notices and warnings about strict coding and deprecated functions. In a production environment, these messages are usually safely ignored, since the developer has had a chance to see them in developing the application and has decided not to address them. In production, they just take up space in the log.

Some customers may also decide not to show warning messages in production. Warning messages show something that is a little bit wrong, but won't usually prevent the script from running. Some people like to see the warnings so they can fix up the scripts, but others don't care so much about them, as long as the application seems to be working OK otherwise. If you do want to suppress warnings, we don't really recommend it, but will show you how:

E_ALL & ~E_NOTICE & ~E_STRICT & ~E_DEPRECATED & ~E_WARNING