

Uncaught Error: Class 'ProgramParameter' not found , UpdateParameterValues does not work

Issue

At one time, the XML Toolkt API included the UpdateParameterValues method. This method was later removed by the community that maintains this open source API. However, Zend continued to use it in a demo program and in the 5250 Bridge demo. If a customer uses this method, and used an include to the ProgramParameter.php script in 5250 Bridge, this function will no longer work in newer versions of Zend Server that do not include the 5250 demo.

Environment

Zend Server 2020 or newer, running on any supported version of IBM i

Resolution

To restore the ability to run UpdateParameterValues, you will need to use the ProgramParameter.php script from a different source. Please copy the ProgramParameter.php script into your application directory, and change the include in your failing script to find it there. There are three ways you can find it.

If the XML Toolkit sample application is installed, you should be able to find it here: `/usr/local/zendphp74/var/apps/http/__default__/0/samples/active/Toolkit/ProgramParameter.php` Notice that `/samples` is the name of the application directory defined at deployment, so the name of this directory may be something different on your server.

If you do not have the Sample app, you can download the ProgramParameter.php script here: [ProgramParameter.php](#)

If you have trouble downloading the attachment, here is the source so you can create your own ProgramParameter.php script:

```

<?php
namespace ZendToolkitApi;

/**
 * App api code
 */

class ProgramParameter extends \ToolkitApi\ProgramParameter
{
    /**
     * updates $arrParams, so pass it by reference.
     * $arrParams is an array of parameter arrays or objects.
     *
     * @deprecated Can't find where this function is used.
     *
     * @param $arrParams
     * @param array $arrValues
     */
    static function UpdateParameterValues(&$arrParams, array $arrValues) {
        if (!is_array($arrValues) || !is_array($arrParams)) {
            return false;
        }

        // loop through all values passed in
        foreach($arrValues as $varName => $newData) {
            // for each value, loop through all params at this level to see if the names match.
            // find a param matching value passed in.
            foreach ($arrParams as $single) {
                // if a data structure, get inner array and call self recursively.
                if (is_object($single) && $single->isDS()) {
                    $arr = $single->getParamValue();
                    self::UpdateParameterValues($arr, array ($varName => $newData));
                } else {
                    // regular param, not a ds. could be an array of values, though.
                    $paramName = $single->getParamName();

                    if ($paramName === $varName) {
                        // $single->setParamValue(self::handleParamValue($newData)); //
                        if data is an array; not done right
                        $single->setParamValue($newData);
                        break;
                    }
                }
            }
        }
    }
}

```