IBM i Compatibility Wrapper tips

Issue

A customer may want to use the Compatibility Wrapper functions to ease the transition from the Easycom toolkit to the free XMLSERVICE toolkit provided in all current versions of Zend Server for IBM i. These notes can help you get started with the Compatibility Wrapper.

Environment

Any supported version of Zend Server for IBM i running on any supported version of IBM i.

Resolution

The Compatibility Wrapper is just some PHP code that rewrites many of the i5_() functions to use the new XMLSERVICE toolkit API instead of the old Easycom (I5_COMD) service.

Compatibility Wrapper does not support data base functions

Many of your existing scripts may run with very minimal changes, but if you have any database functions, those are the ones that are not available in the CW. To quickly check your script for this kind of database access, look for the i5_open() and i5_query() functions. If neither of these are found, you can probably use the Compatibility Wrapper.

if you have any i5_query() or i5_open() functions, the database portions of the script will need to be redone using the ibm_db2 extension.

IBM DB2 PHP documentation

Enable Compatibility Wrapper in your scripts

The Compatibility Wrapper is available to use on any version of Zend Server for IBM i since 5.6. It is not an extension, so it does not need to be enabled. It is just a matter of including it in your script:

```
// Bring in the Compatibility Wrapper
require_once 'CW/cw.php';
```

This will work as long as the XMLSERVICE Toolkit API is in your path, which it is by default in Zend Server. If cw.php is not found, you can add the Compatibility Wrapper to your path using this code:

```
ini_set('include_path', ini_get('include_path') . ":/usr/local/zendsvr/share/ToolkitAPI");
```

The cw.php script has functions for all of the CW functions. Instead of requiring the file in each script, you can also use the auto_prepend_file directive in a .user.ini file to automatically require cw.php in every script in a directory. This can save some typing, but can also put added overhead if many scripts in the directory do not need cw.php.

Additional code changes

You may need some additional changes to your scripts. The most common change is setting the scope for return variables from certain functions that are themselves called from within a function:

Calls to i5_program_call(), i5_userspace_get(), and i5_command() can provide output by creating PHP variables. The i5 Toolkit seemingly exported these variables into local scope (outside the i5_ function, of course), while the Toolkit Compatibility Wrapper beta exports such variables into global scope. Therefore, if these functions are used within a class or a function call, please add the line

```
if (function_exists('i5_output')) extract(i5_output());
```

after the function call.

When invoking i5_program_call(), i5_userspace_get(), and i5_command() in global space (directly in the called script, not from inside a function), no changes are required.

Another less common change involves some of the functions returning objects rather than resources:

Many i5 Toolkit functions, such as i5_connect() and i5_program_prepare(), are returned as “Resource” objects. The Toolkit Compatibility Wrapper will instead return native PHP objects. Any code that checks for is_resource() should use is_object() instead, or simply check that the object is !== false.

Documentation

You can learn more about the Compatibility Wrapper here: Toolkit Compatibility Wrapper

This documentation talks a lot about how to run a separate Apache instance or disable Easycom before using the Compatibility Wrapper. Usually, none of that is needed, because Easycom is not found in any current or even most older versions of Zend Server. The exception would be if the customer has purchased and installed the Easycom standalone into Zend Server from the company that makes Easycom. So you can usually just skip over the Method 1 and Method 2 sections, as they likely do not apply to you.

You can also see the old documentation for the i5_() functions, annotated to tell whether they are available in the CW, and also to let you know about other differences you might need to be aware of:

Introduction to the PHP Toolkit for IBM i

This documentation might also be useful in understanding the old code, if you want to modernize to the XMLSERVICE Toolkit API.

You can learn more about the new XMLSERVICE Toolkit here:

Toolkit Service Class