Upgrade XMLSERVICE on the IBM i from the open source package

Issue

The XMLSERVICE distributed with Zend Server for IBM i is the server side component of the XMLSERVICE Toolkit used by IBM i customers to access native mode objects from PHP scripts, to do things like call RPG, CL, or COBOL programs, or run CL commands. The XMLSERVICE is an open source project hosted on the Young i Professionals web site. There are often newer versions available from the open source project than the version distributed with Zend Server, especially if the customer is running an older version of Zend Server. This article tells how to upgrade the XMLSERVICE and run it in a special library, allowing for a quick reversion to the older XMLSERVICE.

Environment

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

Resolution

The latest release of the XMLSERVICE package, as well as prior versions, is available for download here:

YiPs Wiki - XML Service

and installation instructions can be found here:

YiPs Wiki - XML Service Install

We recommend that you install into the XMLSERVICE Library and then set the Toolkit.ini to use the XMLSERVICE library. This allows the existing version to remain in the ZENDPHP7 (ZENDSVR6 for versions 6- 8.5.X) library, in case you decide to go back to that version for whatever reason. Here are some notes that can help:

Download the current version, or one of the archived versions. The version number is always in the name of the file. For example, at the time this article was written the current version was 1.9.2, and the file containing this version is this:

http://youngiprofessionals.com/wiki/uploads/XMLService/xmlservice-rpg-1.9.2.zip

To get started with the installation, please do the first 4 steps on this page:

YiPs Wiki - XML Service Install

At this point, you should have just restored the XMLSERVICE library from the QGPL/XMLSERVICE save file.

When you get to step 5 to create xmlservice RPG programs in your choice of test and/or production libraries, do this to install into the XMLSERVICE library:

```
irst, add XMLSERVICE to library list:
> ADDLIBLE XMLSERVICE
XMLSERVICE: test library XMLSERVICE, good for trying new versions
> CRTCLPGM PGM(XMLSERVICE/CRTXML) SRCFILE(XMLSERVICE/QCLSRC)
> call crtxml -- XMLSERVICE library only
```

To control whether PHP uses the old XMLSERVICE in library ZENDPHP74 (or ZENDPHP7 for versions 9.1.x, 2018.x - 2019.x or ZENDSVR6 for versions 6 - 8.5.x), or the new XMLSERVICE in library XMLSERVICE, you need to edit this file (**make a back up before changing this file!**):

Versions 2020.x, 2021.x and higher

/usr/local/zendphp74/share/ToolkitApi/toolkit.ini

OR

/usr/local/zendphp74/var/libraries/PHP_Toolkit_for_IBMI_i/default/library/toolkit.ini

> **Note:** Please read the following article to understand why the location may be one or the other, and make sure you are set up correctly to use the newer location:
>
> PHP Toolkit remains at 1.5.0 even though the default version is newer

Near the top of the file, in the [system] section, you will see these lines:

```
; set library where XTOOLKIT lives, most likely XMLSERVICE (testing) or ZENDPHP74 (production)
XMLServiceLib = 'ZENDPHP74' ; production
;XMLServiceLib = 'XMLSERVICE' ; for testing new XMLSERVICE packages
```

In this file, a semicolon at the start of a line marks it as a comment. So, in the example above, the XMLServiceLib is set to ZENDPHP74. We could change it to XMLSERVICE, by putting a semicolon in front of that line, and removing the semicolon from the next line, like this:

```
; set library where XTOOLKIT lives, most likely XMLSERVICE (testing) or ZENDPHP74 (production)
;XMLServiceLib = 'ZENDPHP74' ; production
XMLServiceLib = 'XMLSERVICE' ; for testing new XMLSERVICE packages
```

Save the file to make the change. This change takes effect immediately, although you may need to clear the OPcache. You can also change it back to ZENDPHP74 by simply reversing the change, if the newer version of XMLSERVICE shows some undesirable behavior, or just to compare the results using the different versions. The expected behavior is for both versions to be functionally the same when running existing PHP code.

Versions 9.1.x, 2018.x and 2019.x

/usr/local/zendphp7/share/ToolkitApi/toolkit.ini

OR

/usr/local/zendphp7/var/libraries/PHP_Toolkit_for_IBMI_i/default/library/toolkit.ini

> **Note:** Please read the following article to understand why the location may be one or the other, and make sure you are set up correctly to use the newer location:
>
> PHP Toolkit remains at 1.5.0 even though the default version is newer

Near the top of the file, in the [system] section, you will see these lines:

```
; set library where XTOOLKIT lives, most likely XMLSERVICE (testing) or ZENDPHP7 (production)
XMLServiceLib = 'ZENDPHP7' ; production
;XMLServiceLib = 'XMLSERVICE' ; for testing new XMLSERVICE packages
```

In this file, a semicolon at the start of a line marks it as a comment. So, in the example above, the XMLServiceLib is set to ZENDPHP7. We could change it to XMLSERVICE, by putting a semicolon in front of that line, and removing the semicolon from the next line, like this:

```
; set library where XTOOLKIT lives, most likely XMLSERVICE (testing) or ZENDPHP7 (production)
;XMLServiceLib = 'ZENDPHP7' ; production
XMLServiceLib = 'XMLSERVICE' ; for testing new XMLSERVICE packages
```

Save the file to make the change. This change takes effect immediately, although you may need to clear the OPcache. You can also change it back to ZENDPHP7 by simply reversing the change, if the newer version of XMLSERVICE shows some undesirable behavior, or just to compare the results using the different versions. The expected behavior is for both versions to be functionally the same when running existing PHP code.

Versions 6 - 8.5.x

/usr/local/zendsvr6/share/ToolkitApi/toolkit.ini

OR

/usr/local/zendsvr6/var/libraries/PHP_Toolkit_for_IBMI_i/default/library/toolkit.ini

> **Note:** Please read the following article to understand why the location may be one or the other, and make sure you are set up correctly to use the newer location:
>
> PHP Toolkit remains at 1.5.0 even though the default version is newer

Near the top of the file, in the [system] section, you will see these lines:

```
; set library where XTOOLKIT lives, most likely XMLSERVICE (testing) or ZENDSVR6 (production)
XMLServiceLib = 'ZENDSVR6' ; production
;XMLServiceLib = 'XMLSERVICE' ; for testing new XMLSERVICE packages
```

In this file, a semicolon at the start of a line marks it as a comment. So, in the example above, the XMLServiceLib is set to ZENDSVR6. We could change it to XMLSERVICE, by putting a semicolon in front of that line, and removing the semicolon from the next line, like this:

```
; set library where XTOOLKIT lives, most likely XMLSERVICE (testing) or ZENDSVR6 (production)
;XMLServiceLib = 'ZENDSVR6' ; production
XMLServiceLib = 'XMLSERVICE' ; for testing new XMLSERVICE packages
```

Save the file to make the change. This change takes effect immediately, although you may need to clear the OPcache. You can also change it back to ZENDSVR6 by simply reversing the change, if the newer version of XMLSERVICE shows some undesirable behavior, or just to compare the results using the different versions. The expected behavior is for both versions to be functionally the same when running existing PHP code.

More

If you are not sure what version of the toolkit is running, the following script can help. Note that the functions to show versions were not available in older versions of the toolkit, so if you get problems running this script, you probably are running an older version. Here is the script:

```
<pre>
<?php
// This program shows version and diagnostic info using built in methods

include_once 'authorization.php';
include_once 'iToolkitService.php';

try {

        $ToolkitServiceObj = ToolkitService::getInstance($db, ", ");
    }
        catch (Exception $e) {
        echo  $e->getMessage(), "\n";
       exit();
    }


$ToolkitServiceObj->setToolkitServiceParams(array('plug'=>"iPLUG1M", 'stateless' => true));

// show front end version
$myVersion = ToolkitService::getFrontEndVersion();
echo "PHP toolkit front-end version: '$myVersion'<br><br>"; // 1.4.0

//show back end version
$myVersion = $ToolkitServiceObj->getBackEndVersion();
echo "back-end (XMLSERVICE) version: '$myVersion'<br><br>";

// show diagnostics
$diagnostics = $ToolkitServiceObj->getDiagnostics();
echo '<PRE>' . print_r($diagnostics, true) . '</PRE>';
// returns array containing [jobinfo][jobnbr], [jobinfo][jobuser], [joblog], [version], more.

$ToolkitServiceObj->disconnect();
?>
```