OpenLogic

WHITE PAPER

Advantages of Open Source on IBM i

Executive Summary

In May of 2018, IBM released its support for the open source package management (rpm/yum) ecosystem on IBM i. Since then, the availability and adoption of open source on the platform has been growing at a healthy clip.

This white paper explores the source of this growth and examines why now may be the time to consider bringing open source solutions to your IBM i platform. In addition, this white paper will provide a review of the advantages of open source, as well as provide an understanding of the open source development and support paradigms.

Contents

Is Open Source Free?

An often-voiced advantage of open source is its low cost of acquisition. While on the surface this is true (some estimates put the yearly savings upwards of $60 billion) one should keep in mind that the cost calculation here is only considering the cost of acquisition. One needs to keep in mind the total cost of ownership which includes acquisition, installation, configuration, maintenance, and support.

Open source follows a community approach to both development as well as support and as more organizations are adopting an open source strategy, commercial solutions to support open source have started to emerge, including IBM's Technical Support Services (TSS) as well as OpenLogic by Perforce.

While adding cost, these and other commercial open source support offerings address the preference to have the same type of support for open source solutions as that found for commercial products. They also address security requirements and corporate and/or government regulations requiring a formalized support mechanism for implemented solutions.

While adopting a commercial support vehicle for open source solutions does negate the "free" nature of open source solutions, the overall cost is still less than that of commercial solutions. One final thought on cost/savings — while important it should not be the driver for choosing open source. Innovation should be the driver.

Open Source Drives Innovation

The level of innovation in the open source space is nothing short of breathtaking. Virtually any "cutting-edge" initiative that comes to mind has various open source projects associated with it. Everything from machine-learning, to artificial intelligence, to enterprise

messaging, enjoys the collective thought of hundreds and sometimes thousands of talented designers and developers — which is something that commercial endeavors with their limited staffs simply can't match.

Additionally, the engineers that work on open source products tend to do so from a passion for the technology rather than the necessity of producing a commercial product within a given set of constraints.

What Drives Reliability in Open Source

The reliability of open source can not be underestimated. At first look, one might think that the reliability and security of open source would be suspect due to the nature of the code being open to everyone — but it is actually that openness that ensures that the code is both reliable and secure. Consider the following diagram:
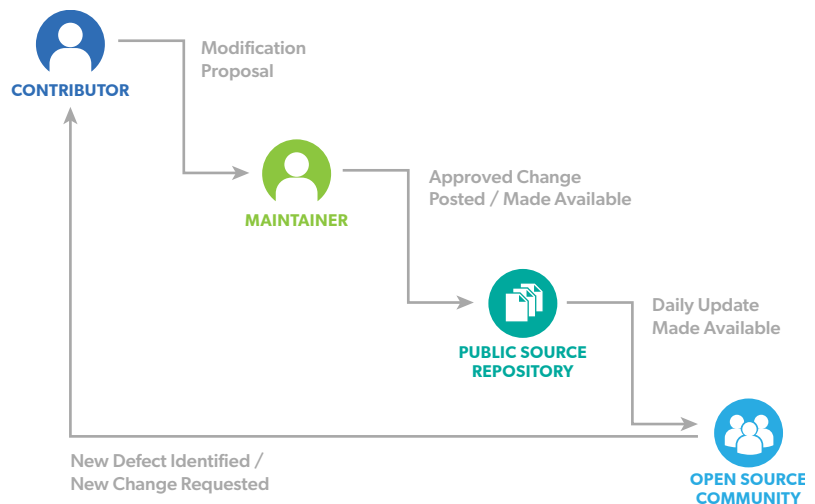
When first looking at the open source movement many think that it represents the "wild-wild-West" of code development and that there are no controls over what makes its way into open source-based solutions. The reality is quite different, open source fosters a more collaborative effort that results in very robust and secure code.

To gain an understanding of the open source ecosystem one first must understand the role of contributors and maintainers. A contributor is anyone who writes or modifies code in an open source project while a maintainer is a person (or persons for larger projects) that essentially holds the ownership of the project and "controls" for lack of a better word when code is committed to the open source code repositories.

So while anyone has access to the code of open source projects and can make modifications to that code it is the maintainer(s) that is the gatekeeper or arbiter that decides if/when that code is publicly available through the repositories.

What drives quality into open source solutions is the peer-review process and the deep understanding of the solution that the reviewers tend to have through the study of the code (after all it is freely available) and the free exchange of ideas with peers. The open-source developer uses this knowledge to propose changes or enhancements to a development effort.

Some would say that the open access to the code makes the quality and security of the code suspect; however, that is simply not the case as it is negated by the aforementioned peer-review process and maintainers' ability to gate the promotion of code into the public repositories.

Here is how it works: A contributor contributes code by submitting the code either via an email list or through a version control system such as git. In either way, individuals interested in the particular open source project are notified of the submission. These individuals can be other developers, security analysts, interested users, etc. Once the contribution has been made peer-review starts to take place.

Unlike the closed-source development paradigms which typically have teams of six (6) or so developers reviewing code, the size of the peer review for open source is only bounded by the popularity of the project. In some cases hundreds if not thousands of interested parties are reviewing the contribution.

The peer review is a very iterative process where suggestions on changes or improvements to the contribution are made and considered. Over time as suggested changes are incorporated the code becomes stable and the maintainer will then decide to commit the changes to the source code repositories. As you can imagine, with such a large peer-review process and the iterative nature of the process it is hard for bad code, whether simple defect or malicious intent, can survive the light of day that the review process provides.

Once the code is in the repositories, the entire open source community has access to the code and can start to test it, incorporate it into their own solutions, etc.

End-users also have a role in the robustness and effectiveness of open source solutions. Consider the following modification to the earlier diagram:
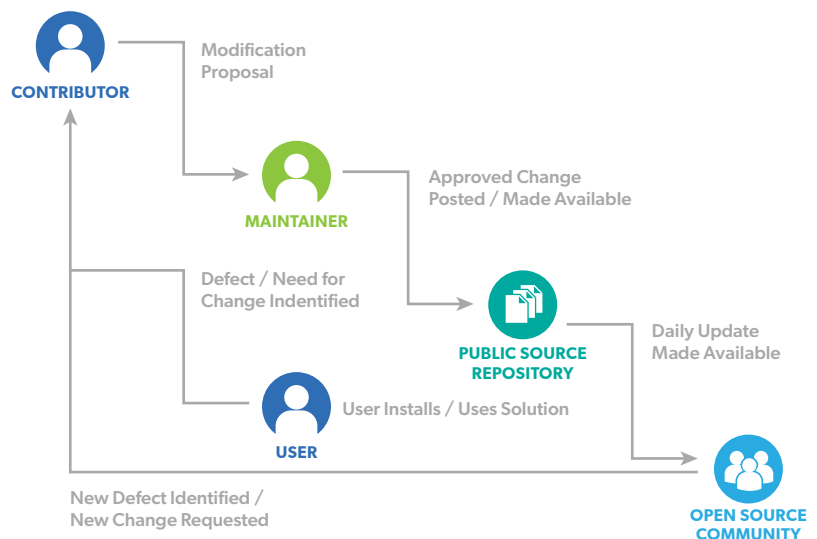
End-users have the capability (and some would say the responsibility) to be active members of the community by reporting defects as well as requesting enhancements to solutions being evaluated/used. Additionally, the user community can help through writing documentation, such as how-to(s) and tips/tricks as well as participating in solution forums.

An Open Source Passion Project

Another key factor that drives quality and innovation in the open source movement is the developers themselves. When developers are allowed to follow their passions, they will be driven toward specific solution-types or technology areas where they will end up doing their best work. When it is no longer compensation or performance measurement that drives them but rather proving to themselves, their peers, and the marketplace -- they can do their best work. The development is no longer work, but rather a pursuit of perfection.

## How IBM i Benefits from the Open Source Movement

The above may sound somewhat altruistic and pie-in-the-sky but the proof is in the technologically advanced solutions that have come out of the open source movement. What does this mean to the IBM i shop? It means that we can benefit from the same technologically superior solutions as other platforms. It means choice — choice in both the types of solutions we deploy as well as choice between solutions/products for satisfying a particular business requirement.

PASE, the Enabler for Open Source

While the above provided a general view of the advantages of open source, let's take a closer look at the advantages that open source brings to IBM i. Before we begin, a basic understanding of how open source is implemented on the platform is in order. Since the convergence of iSeries and pSeries into a unified POWER

system, IBM i has had the ability to run MI-based and AIX-based executables on the same hardware within the same Logical Partition (LPAR).

With this capability IBM implemented the Portable Application Solution Environment (PASE) for running the AIX-based executables. It's important to understand that PASE is not a release of AIX but rather a set of AIX libraries that have been fitted to talk to IBM i rather than directly to the AIX kernel. The following provides a high-level look at the architecture of this environment:
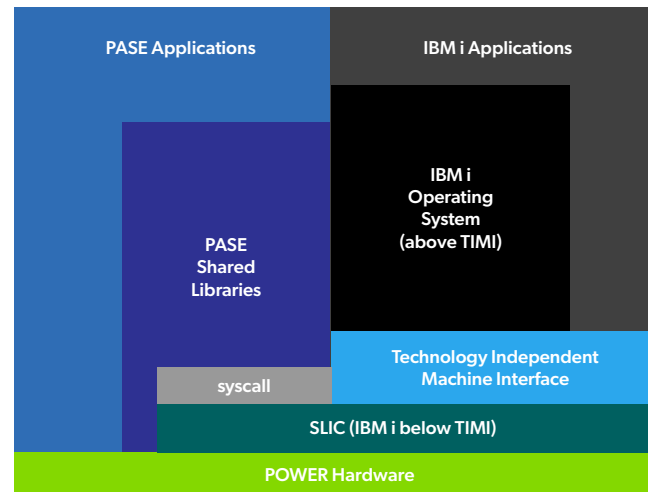
As one can see from the diagram, PASE runs on top of the same hardware as IBM i and programs running within PASE have access to IBM i items such as Db2 resident data, programs (RPG, Cobol, CL), commands, data areas, and data queues (to name a few) via the syscall interface between PASE and the TIMI layer of IBM i.

This enables the implementation of *NIX-like applications within PASE that have native access to IBM i resources while being able to leverage the benefits of POWER Systems including processor allocation strategies, such as shared processors, memory sharing, and I/O performance.

What Can Be Done With Open Source?

Now that the advantages of open source have been outlined, what can you do with open source on IBM i today? First, keep in mind that the availability of open source on IBM i continues to grow at a break-neck pace with over 400 packages available at the time of writing this paper.

In addition to the advantages of innovation, reliability, and quality that open source solutions brings with regard to the IBM i ecosystem, open source solutions also bring choice to the organization.

One example of this would be the myriad of solutions based on the LAMP (Linux, Apache, MariaDB, PHP) stack. The availability of PHP and MariaDB on IBM i enables the entire LAMP-ecosystem of solutions.
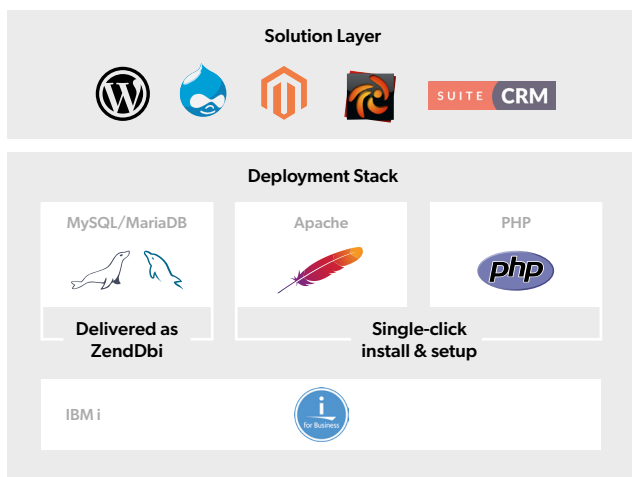
Looking for a Content Management Solution (CMS)— you can check out items such as Joomla, Magento, and WordPress. Need a Customer Relationship Management (CRM) solution? SweetCRM might be a god fit. Eager to embrace eCommerce? Then solutions like ZendCart might be right up your ally. Keep in mind that all of these solutions run unchanged on IBM i!

A simple install of PHP and MariaDB and you have your own variant of the LAMP stack referred to as iAMP — the same Apache Web Server, the same MariaDB open source database, the same PHP scripting language as other platforms — just a better Operating System than any other platform offers and the solutions on top of the stack — again that run unchanged.

But the expanded choices that Open Source provides does not end there. If you are interested in leveraging enterprise messaging — solutions such as Kafka, ActiveMQ, and Apache Camel can now be implemented on the platform.

Benefits of Open Source Languages and Integration with IBM i

If you are interested in developing your own solutions on the platform, the availability of open source languages has grown significantly over the last two years to include popular languages such as PHP, Python, Node.JS, and .NET.

The availability of open source languages brings another benefit above and beyond choice, and that is the ability to attract new talent to the platform. This talent can then use the languages and their integration points (Db2 drivers, toolkit support for XMLService) to develop customized solutions based on language skills readily available in the university and technical school systems.

The integration aspects of the IBM i platform cannot be overstated, as a key benefit is being able to leverage various integration capabilities between open source solutions and IBM i itself.

As an example, there are database drivers for each of the open source languages that provide the ability to develop and implement applications with these languages that are tied to the IBM i data resident in Db2 tables.

Why is this a benefit? The PHP language has an affinity for the development and deployment of web-based applications that are tied to dynamic content. In fact, in many respects, it is PHP that ushered in the growing predominance of dynamic content on the web. The availability of database drivers for PHP tied to Db2 make it is possible to tie PHP applications to IBM i resident data and extend the usefulness of that data to both a wider set of devices (full web-browsers, tablets, mobile devices, etc) as well as a wider set of constituents and stakeholders.

Keep in mind it's not just PHP. Other open source languages such as node.js and Python have drivers tied to Db2 data as well. While access to Db2 will vary with each language the actual process is not difficult.
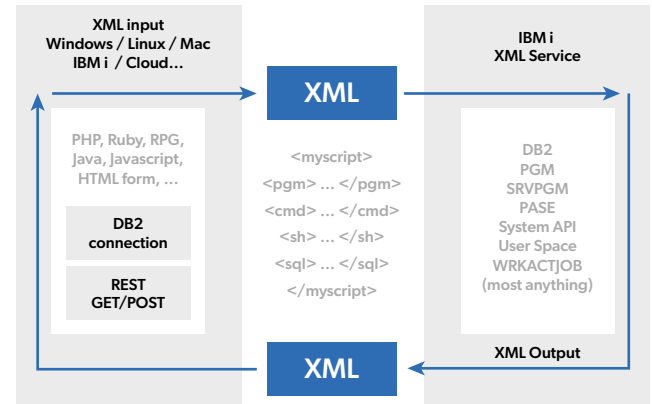
Take the following PHP code as an example:

But what about other IBM i items such as RPG programs, CL commands, data areas, data queues, and the like? The integration answer for that is XMLService and each open source languages' corresponding toolkit. Let's first talk about XMLService. The following diagram provides a high-level view:


```php
<?php
  $conn = db2_connect('*LOCAL', '', '');
  $result = db2_exec($conn, "SELECT * FROM zendphp7.sp_cust");
  $firstRow = db2_fetch_assoc($result);
  echo "<pre>" . print_r($firstRow, true) . "</pre>";
```

The above code is a complete PHP example that uses the Db2 driver (in the case of PHP the driver name is ibm_db2) that selects records from a table and outputs the results.

Obviously, there is more that can be done including error-checking and output formatting; however, the point is that Db2 access from open source languages provides an easy point of integration between open source and IBM i.

Additionally, the Open Database connectivity driver (ODBC) can be used by all of the languages to provide an agnostic way of developing applications that again are tied to Db2 data.

XMLService, developed by IBM and contributed to the open source community, is a collection of RPG programs and stored procedures that provides the ability to invoke programs in the ILE (RPG, Cobol) as well as CL commands by providing an XML payload that "describes" the request and responds with another XML payload that provides the result. By the way, it's not just about calling programs, items such as data areas and data queues are available via XMService as well.

So, what does the XML look like, let's take a look:

```
1
2    Exec start: 2020-05-08 14:37:15
3    Version of toolkit front end: 1.7.0
4    IPC: ''. Control key: *cdata *here
5    Stmt: call zendphp7.iPLUG32K(?,?,?,?) with transport: ibm_db2
6    Input XML: <?xml version="1.0" encoding="ISO-8859-1" ?>
7    <script>
8    <sh rows='on'>/QOpenSys/usr/bin/system "DSPLIBL"</sh>
9    </script>
10   Output XML: <?xml version="1.0" encoding="ISO-8859-1" ?>
11   <script>
12   <sh rows='on'>
13   <row><![CDATA[ 5770SS1 V7R3M0  160422                  Library List
14   <row><![CDATA[                              ASP]]></row>
15   <row><![CDATA[   Library    Type       Device    Text Description]]></row>
16   <row><![CDATA[   QSYS       SYS                  System Library]]></row>
17   <row><![CDATA[   QSYS2      SYS                  System Library for CPI's]]></row>
18   <row><![CDATA[   QHLPSYS    SYS]]></row>
19   <row><![CDATA[   QUSRSYS    SYS                  System Library for Users]]></row>
20   <row><![CDATA[   QGPL       USR                  General Purpose Library]]></row>
21   <row><![CDATA[   QTEMP      USR]]></row>
22   <row><![CDATA[                        * * * * END OF LISTING * * * *]]></row>
23   </sh>
24   </script>
25   Exec end: 2020-05-08 14:37:15. Seconds to execute: 0.12581586837769.
```

The call — lines 7-9

The result
Each CDATA represents a line of output from command execution — lines 11-24

As the diagram shows, one could craft the XML for the call as shown in lines 7-9 of the example (which in this case is calling the DSPLIBL CL command) and send it to XMLService and then dehydrate the XML returned by XMLService (lines 11-24) which shows the output of running the CL command.

There is an easier way! Each open source language provides a "toolkit" for wrapping around XMLService and making it easier to both call the given ILE item and process through the returned result.

Consider the following diagram:
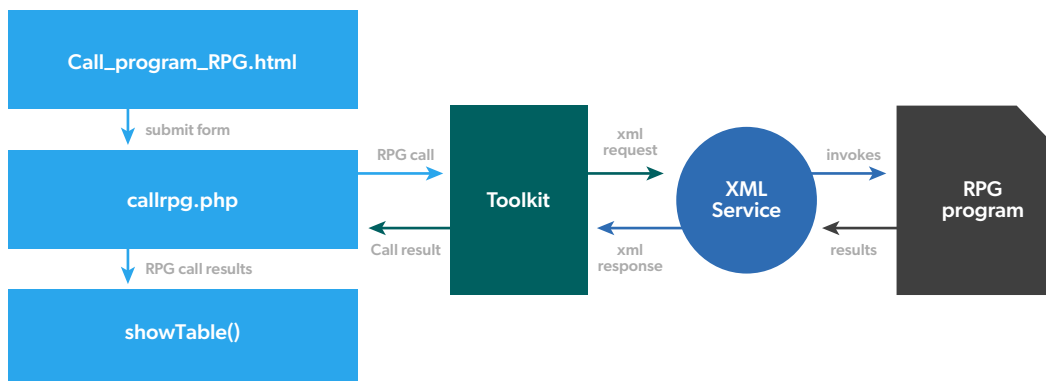
While this is a PHP example, the process holds true for the other languages with their toolkit as well. As you can see the open source program uses the toolkit to specify the item in the ILE to integrate with (in this case an RPG program), the toolkit builds the XMLpayload and then passes that request to XMLService. XMLService uses the information from the XML payload to call the specified RPG program with the specified parameters.

The return path is fairly straight forward. The results from the call to the RPG program are taken by XMLService, an XML payload of the result is built up and provided to the toolkit which then dehydrates the result and provides it to the caller, the open source program, which can then process through the results. Sound difficult? Let's take a look at an actual code example:


```
Call_program_RPG.html
        │ submit form
        ▼
    callrpg.php ──────RPG call──────▶ Toolkit ──xml request──▶ XML Service ──invokes──▶ RPG program
        │        ◀─────Call result───         ◀──xml response──            ◀──results──
        │ RPG call results
        ▼
    showTable()
```

The code to the right is complete, and it uses the toolkit to invoke XMLService to execute the CL DSPLIBL command. The output from the print_r statement is what one would expect, the output of execution of the DSPLIBL command.

```php
<?php
require_once zend_deployment_library_path('PHP Toolkit for IBMI i') .
    '/ToolkitService.php';


$obj = ToolkitService::getInstance();


$rows = $obj->CLInteractiveCommand('DSPLIBL');
echo '<pre>' . print_r($rows, true) . '</pre>';
$obj->disconnect();
```

Again, the point here is that integration between the open source languages and Db2 data as well as ILE programs and artifacts is both easy and powerful.

Closing Thoughts

The advantages of Open Source on IBM i are significant and they include choice of solution, innovation to the platform, and integration with IBM i data, programs, and artifacts to extend their reach and usefulness.

Getting started on open source on IBM i is easy — it just takes a willingness to embrace new methods and new solutions to strengthen an already powerful platform to take it to even higher capabilities.

Get Support for Your Integrated Open Source

No matter which open source software you use on IBM i, OpenLogic can provide dependable, expert-level support for your open source stack. See how we can support your integrated open source software at www.openlogic.com/solutions/support

SEE SUPPORT OPTIONS

openlogic.com/solutions/support

About Perforce

Perforce powers innovation at unrivaled scale. With a portfolio of scalable DevOps solutions, we help modern enterprises overcome complex product development challenges by improving productivity, visibility, and security throughout the product lifecycle. Our portfolio includes solutions for Agile planning & ALM, API management, automated mobile & web testing, embeddable analytics, open source support, repository management, static code analysis, version control, IP lifecycle management, and more. With over 20,000 customers, Perforce is trusted by the world's leading brands to drive their business critical technology development. For more information, visit www.perforce.com.