The BIC Library Communication Framework

Improving communication between library systems

WELCOME TO THE BIC LCF NEWSLETTER

Issue 2, March 2020

RELEASE 1.2.0 NOW AVAILABLE!

The latest release of the Library Communication Framework was published earlier this year.

Read about the changes and improvements in this issue!

https://bic.org.uk/114/Library-Communications-Framework

Hello again!

It's been some time since Issue 1 but that doesn't mean we haven't been busy!

In many ways LCF (the Library Communication Framework) has been something of a victim of its own success. After being endorsed by the government's Library Taskforce interest in the project grew exponentially both domestically and internationally. For a while requests for enhancements and improvements threatened to overwhelm BIC's resources and efforts to promote LCF to the library community were almost entirely suspended in order to meet the new demands.

So in this issue, we are attempting to do two things.

Firstly to share with you the fruits of our labours. We have news of a major LCF installation - in Australia - and of our work with the US National Information Standards Organization (NISO) as well as details of the latest release (version 1.2.0) which by adding new functionality also demonstrates the greater flexibility of LCF over previous interoperability protocols.

…all this effort would be wasted if we fail to persuade the people responsible for delivering, maintaining and using library systems to use LCF.

But all this effort would be wasted if we fail to persuade the people responsible for delivering, maintaining and using library systems to use LCF and a very recent survey (details in this issue) shows that while we may have been making headway with suppliers we still have much work to do to make users aware of the benefits of buying solutions that support the protocol.

So our second objective is to increase awareness of LCF among library service providers across all sectors.

BIC is often asked "What exactly is LCF?" often swiftly followed by "I'm not a technical person – can you explain it to me in simple terms?" In his valedictory article, Mick Fortune attempts to answer both questions.

The BIC Library Communication Framework

Improving communication between library systems

CONTENTS

WHY LCF?

On reflection there have probably been two reasons for developing LCF, one technical and one economic.

The technical driver was RFID.

Nowadays RFID is so widely used in every aspect of modern life that it's hard to recall that it first appeared in libraries over 20 years ago and was then practically synonymous with self-service.

Self-service machines using simple magnetised strips had first been pioneered in American libraries by the 3M Corporation – who were also the first supplier to realise that connecting their machines to library management systems (LMS) would be both expensive and complicated unless they could persuade developers to use a protocol. Helpfully 3M provided one – the Standard Interchange Protocol, or SIP for short.

SIP performed very well for many years. Third parties – not providing self-service solutions -were allowed to write their own 'extensions' to connect their devices and services – all of these used their own software, values and variables.

RFID brought with it the potential to store more data on library items. The potential for improving speed and quality of service that this offered was immense. The problem was that self-service providers had decided for themselves how to connect their new RFID systems.

And they all did it differently.

So after 10 years of missed opportunities and librarians discovering that it was going to be at best very expensive and at worst impossible to switch providers the industry finally accepted an international standard – ISO 28560 – and the UK data model created from it by BIC.

This new standard at last offered the potential to improve and expand functionality for self-service providers but did nothing for all those third-party providers of booking systems, print service management etc. who had developed SIP based solutions. A different solution would be required to do that.

And so LCF was born.

Originally LCF was designed simply to replace SIP as the primary means of communication between self-service and LMS providers. The recent expansion of services beyond self-service was however always the aim.

From the beginning LCF was developed as an open source product freely available in the public domain. BIC's hope was that this would encourage its adoption across the sector and commercial suppliers have been slowly adapting their systems to use it. Somewhat ironically the only LMS systems providers that have yet to show any interest in the new protocol have been those based on Open Systems so partly to encourage them BIC expanded LCF's deliverables to include a working RESTful implementation that could be used off-the-shelf, free of charge.

From the library services' point of view the adoption of common standards for interoperability is an obvious benefit. As many had discovered during the first ten years of RFID implementation their absence resulted in proprietary solutions that were expensive to replace (if they could be replaced at all!). Without a common standard, libraries remain vulnerable to repeating the errors made when RFID was first introduced.

And avoiding expensive mistakes leads directly to the economic reason for using LCF.

In the almost never-ending saga that is austerity the need for library service providers to extract maximum benefit from meagre resources has never been more important but as the pace of technological change increases and new applications and services need to be integrated with existing systems some libraries – particularly in the public sector – have struggled to do this effectively as their access to ICT skills has been in decline for many years.

One suggested solution to this problem that has been steadily gaining in popularity is what I might call the 'one system to rule them all' model. In this scenario a number of libraries work together to procure a single system to look after the needs of each member.

One of the difficulties with such a solution is a continuing dependence on an LMS as the central component. Even the most cursory examination of the library market will reveal that the LMS is just one of legions of inter-related (and interoperable) systems in use today. Mobile apps like Solus (of which more later), booking, self-service, e-book and print management systems to name but a few, all play a significant role in the modern library and ensuring their ability not only to work together but to be compatible with future technologies should be key to any financial planning.

Buying a single LMS – or even a single suite of systems – today is no guarantee that they will be 'future proof'. Making sure that all your suppliers use common variables and values – like LCF – significantly reduces that risk. If any of your application and system providers don't support it, ask them why not. It could make all the difference to the future of your service.

GETTING THE MESSAGE OUT

2019 was a relatively quiet year for promotion of LCF. One major event was the annual Plugfest, held in Birmingham in May at which suppliers from the UK's major commercial suppliers of library systems met to test their LCF implementations and determine priorities for expanding the protocol. It became clear after that event that release 1.2 would be of such significance that it would be best to delay a publicity campaign until it had been delivered. This Newsletter is the first salvo of this new attempt to raise awareness of LCF in the wider library community!

And not before time…

Through our work with the Libraries Taskforce we had been grateful for the enthusiastic support shown by other members for our efforts and had formed close links with both Libraries Connected (LC) and the Scottish Library and Information Council (SLIC). So in December 2019 we turned to both of these organisations for help in assessing the current level of awareness of LCF among their respective members.

Between them representing the interests of most public library services in the UK they agreed to circulate a very short questionnaire to their membership on behalf of BIC.

We asked,

» **1. Do you know whether any of your systems providers have deployed LCF in any of your library applications?**

» **2. If so, please supply any information you can about the nature of this deployment where possible including your impressions of how this has affected day to day operations. Could you also advise the name(s) of your providers?**

» **3. Are there any specific areas of library operations, not currently addressed by LCF, that you feel greater systems interoperability might improve?**

» **4. How would you like to see LCF develop in the future (new areas of operation, tackling specific interoperability issues etc.)?**

» **5. Do you foresee any restrictions on the future development of LCF a) by product b) by market sector (e.g. in public libraries)?**

Responses were mixed. Overall, they told a story of a somewhat disappointing lack of awareness of LCF and its purpose among library service providers although many also complained of a negativity by some suppliers to their requests to implement LCF based solutions. Indeed, only one authority reported that they had successfully implemented LCF with 2 out of 3 suppliers – but no-one had yet managed to do this with self-service.

This was something of a surprise since all the major suppliers of self-service systems have been publicly supporting LCF for some years now and have helped to develop the protocol. Bibliotheca have advised us (via Chris Millican) that they will be releasing a new LCF-based version of their software in May of this year. Other suppliers have told us that they are ready to implement LCF but are waiting for clients to request it.

"I have been disappointed by most kiosk vendors lack of any development for LCF so I think inertia is the biggest problem."

One reply summed up the status quo for me. "I have been disappointed by most kiosk vendors lack of any development for LCF so I think inertia is the biggest problem. It would be great to come up with a quick win that could be achieved, over the capability of SIP, that might awaken library purchasers to the need to modernize."

BIC believes Version 1.2 offers support for significant additional functionality that may help establish the 'quick win' identified here but more importantly it is clear that without an increase in demand from clients, suppliers are unlikely to lavish resources on developing it.

Replies also showed that new application providers are tending to use LCF whereas established players – especially those in self-service – are maintaining a watching brief.

There were plenty of suggestions for ways in which LCF could be extended to support new services – evidence perhaps of an underlying demand that is at present being discussed in individual client-supplier concersations rather than industry-wide.

In the UK LCF is little known outside of the public library sector. BIC must take some responsibility for this lack of awareness. Our work with other members of the Library Taskforce tended to move us away from other library markets and such efforts as we have made to spread the word to the wider community have been few and far between. We still hope for support from bodies like CILIP and its Scottish and Welsh counterparts in getting the message out and will be redoubling our efforts to reach out to the educational and special library communities during 2020.

LCF IN AUSTRALIA

While progress in the UK has been relatively slow LCF has been gaining new supporters elsewhere.

Solus established their LCF credentials working with BIC and the Libraries Taskforce back in 2016 and in 2017 had helped the latter publish its core data set.

As Nick Poole, CILIP Chief Executive, said at the time,

"Reliable, open data about public libraries is the bedrock for good decision making. Having open data that can be examined and laid across other data-sets such as transport links, health information, urban infrastructure, education and socio-economic status will enormously contribute to effectively managing and developing library services."

So when Solus began work in Australia they also became advocates for LCF.

Two of their Australian case studies demonstrate how the LCF philosophy works in practice

Mackay Libraries use a library management system supplied by Aurora IT (AIT). Not members of BIC they were nonetheless able to access all the information they needed

to integrate Solus with their system from BIC's Open Source site. I wrote to their director, Doug Coulson to ask how they found the experience. He replied,

AIT chose the open option of developing our API for the mobile app using the modern, standards based and relatively easy to learn LCF.

"The library industry is struggling to find a common and secure method of communication between LMS software and 3rd party products. Aurora LMS software has been integrated with many 3rd party providers and almost all had different needs and requirements. When AIT decided to integrate with the Solus mobile app, the choices were either to use SIP2, an Aurora proprietary API or the new Library Communications framework (LCF). Having extended knowledge and experience with the SIP2 protocol, we understand its limitations in the transmission, communication, and syntax and language definition. The Aurora API (including Aurora Web Services) was the easy option for us but in addition to our own development, it would also have required considerable and proprietary integration efforts from Solus. AIT chose the open option of developing our API for the mobile app using the modern, standards based and relatively easy to learn LCF.

Although the LCF standard is quite new, we found it fairly mature as it starts where SIP2 ends and it includes the main SIP3 recommendations, so our extensive SIP2 knowledge was not lost. SIP2 provides a protocol for circulation functionality but it is dated and is not fully secure without VPN or encryption. LCF's main advantage therefore is being able to transmit over the internet in a secure way, so it is far superior by operating without the need for a VPN, SSL or TLS encryption software. We found its documentation reasonably detailed and easy to understand and follow. We also found development with the LCF framework to be very straight forward, including searching for titles/manifestations implementing the OpenSearch 1.1 AIT will continue to use LCF in other applications and we've already started to do that."

Melbourne City Libraries are users of Civica – a system widely used in the UK and full members of the LCF Consortium. Civica's SPYDUS system was one of the first to commit to support for LCF. I asked their Senior Product Manager in Australia, Jeremy Langley, how they had found working with LCF.

"At Civica, providing means to connect data is a priority. We want to ensure that we are developing applications and data integration points to enhance the secure exchange and management of information. Having available APIs, help improve service delivery, efficiency and collaboration for libraries and their partners. Our LCF development has allow us integrate with different partners in a secure way, using a standard approach, which reducing risk and costs. Through BIC, we have been able to have input into the various stages, and continuous development of LCF, to ensure it is a standard that works for all libraries."

"feedback has been on the whole incredibly positive"

City of Melbourne Libraries Technology Innovation Coordinator, Ken Harris also told me,

"…(we) worked closely with Solus over the course of a year to provide functional requirements, testing and feedback on their interfacing with the Civica API implementation of the LCF standard. The outcome of this careful testing, communication, flexibility and patience is apparent in the take up of the app across our user base. Since launching 4 months ago we have had 5,400 devices install the app and over 40,000 app visits. The feedback has been on the whole incredibly positive. I look forward to working through some additional features to make the app match and perhaps in some ways exceed the functionality of the online catalogue."

Gratifying as it is to see how easy to implement and rewarding (for client and supplier alike) to use LCF has been in Australia – a country where BIC has spent £0 promoting the protocol – it only emphasises the need for us to find a way to better inform the UK library market.

THE LATEST RELEASE VERSION 1.2.0

LCF is under constant review by the LCF team, who welcome suggestions for future revision to better serve the needs of libraries.

Anyone can contribute to and follow discussions about ideas for improvements and new features in LCF, by visiting the LCF pages on GitHub (github.com/bic-org-uk/bic-lcf). Go to the 'Issues' tab to read current and past issues and to create new issues for discussion. As each issue gets considered by the LCF team, it is tagged in various ways to indicate the nature of the issue, and, if appropriate, in which future version of LCF it is likely to be resolved.

LCF versions

New versions of LCF are released with varying frequency, depending upon demand for new or improved features and the nature of the revision. LCF uses "semantic versioning", with versions falling into three categories: a "patch" version only contains backwards-compatible bug-fixes; a "minor" version contains new functionality added in a backwards-compatible manner; a "major" version is not backwards-compatible with previous versions. The current released version is '1.2.0', indicating that there have been two minor versions released since this "semantic versioning" approach was adopted in April 2018.

LCF development and review process

Most of the technical work to maintain LCF – to draft revisions and to respond to issues posted on GitHub – is done by a team of technical editors, of whom there are three currently. Proposed revisions are discussed in detail by the LCF Technical Panel, comprising the technical editors and experts representing libraries and suppliers. Progress on LCF development more generally, including agreeing the overall timetable for revision, is governed by the LCF Review Group.

Development of a new or improved feature for LCF generally starts with an issue being posted to GitHub. The technical editors review the issue initially and come to a view as to how the issue can best be resolved. Bugs and errors in documentation that prevent or hinder implementation of the current version will be fixed as a matter of urgency. Extensions of

existing features that can be introduced in a backwards-compatible way are likely to find their way into the next minor release within a 12-month period. Major new features that replace existing features of LCF will require more effort on the part of implementers, so will be given more careful consideration and unlikely to be included in a new version until implementers are ready for it.

New features

It is important to remember that when we talk about "features" in the context of LCF we are not talking about functionality. One of the criticisms levelled at LCF in its early days was that it wasn't sufficiently "visible". Another was that it is only of interest to those of a technical bent. It's difficult to defend LCF against these charges because they are essentially true!

But what should matter to consumers is evidence of LCF's ability to respond both to changes in the market and requests from those seeking to improve their products and services. Each of the items in the following list represents a response to a request from somewhere in the library world, and each has a 'business case' associated with it – whether that is to add an additional element to support a product (or products) or data to complete the Taskforce Core Data Set. The details of the business case (or cases) proposed and the discussions that ensued for each may be found on the LCF GitHub site.

More information on Release 1.2.0 is available at https://bic-org-uk.github.io/bic-lcf/ but for the dedicated here are some of the highlights that have been added over the past year,

» **Revised and clarified the documentation for terminal and patron authentication and authorisation**

» **Added support for returning library opening times**

» **Ability to check the delivery status of messages or alerts sent to Patrons**

» **Added the ability to specified ranges (for example data ranges) in selection criteria**

» **Deprecated classification scheme and classification code entities**

LCF now uses semantic versioning (of the form major.minor.patch), where:

» **Patch** updates will resolve errors in documentation or provide clarifications which do not change the underlying technical specification

» **Minor** updates will add functionality in a backwards compatible manner

» **Major** updates will include changes which are not backwards compatible, and as a result will use different namespaces and web-service endpoints

The next substantial version of LCF will consider the following issues:

» **Support for JSON formatted requests and responses as well as the existing XML formats**

» **Support for OpenAPI - a machine-readable documentation language for REST web services which has growing tool support for generating client and server code (https://www.openapis.org/)**

» **Inter-library loan support**

» **A richer selection query language**

» **Ability to expand sub-entities and restricting which fields are returned in responses**

We welcome feedback on these issues as well as any other issues or suggested enhancements. Discussions and proposals can be made via github issues:
https://github.com/bic-org-uk/bic-lcf/issues

LCF Reference Test Server

To assist testing client implementations basic and connectivity and communications, there is a test implementation of LCF 1.2.0 running at https://lcf.ceridwen.com. This implements REST endpoints for all LCF Entities. In addition, it also includes a web-based interface using Swagger which allows exploration of the various REST operations, provides example requests and responses, and provides a basic web user interface to make live requests and responses.

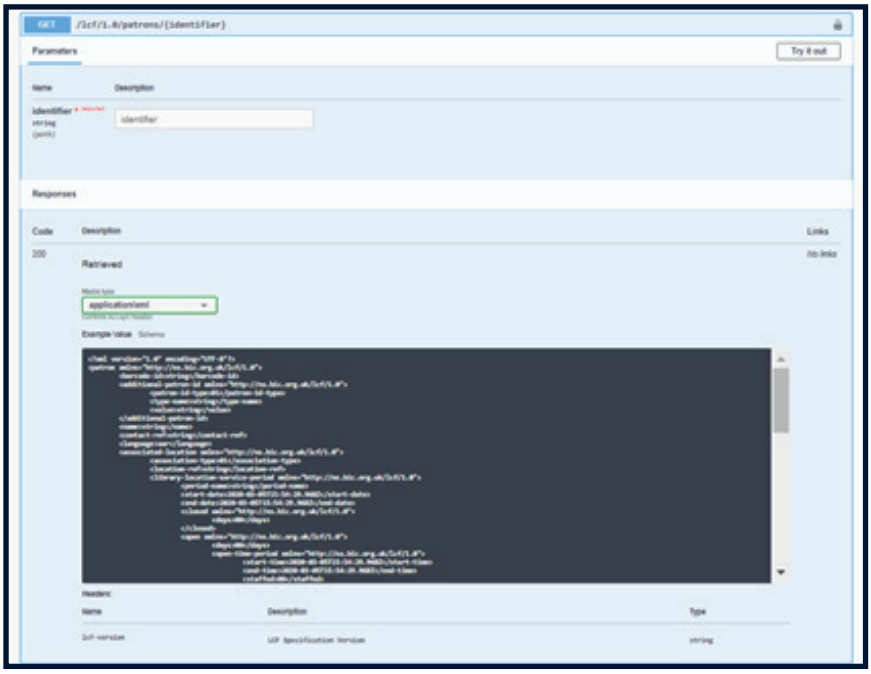

Figure 1- Swagger UI for Patron REST operations

The server includes support for JSON formatted requests and responses which is being considered for the next version of LCF. The server also provides an OpenAPI specification for LCF. OpenAPI (https://www.openapis.org/) is a machine-readable language for REST web-services which can be used for generating client and server code.

The Java source code for the server is open source under the Apache license, and available on github. It uses JAXB and StringTemplate to generate as much of the code as possible from the official LCF XML schema so that it can be quickly updated to support new versions of the LCF standard by recompiling the code with the latest version of the LCF XML Schema.

The server code is modular allowing different backend implementations to be "plugged" to interface with different backend databases, library systems or other implementations. From those using Maven for their development platform, there is also a Maven archetype which will generate template backend code as a basis for developing new server backends.

Future developments will include a more comprehensive test framework as well as client-based tools for testing server implementations.

Move information including links to the REST endpoints, Swagger UI, github repository and Maven instructions can be found at https://lcf.ceridwen.com/

LCF AND NISO – THE FASTEN PROJECT

FASTEN – the **F**lexible **A**PI **St**andard for **E**-content is a **N**ISO project to develop recommended practice for using RESTful web service APIs to serve licensed e-content to library patrons.

Based upon requirements originally specified by Queens Library, New York the project has grown to include national and international participation. One of LCF's Technical Editors, Matthew Dovey, is a member of the FASTEN project and has been instrumental in promoting the aims and capabilities of LCF to NISO members.

The FASTEN recommendations include the LCF message specifications as the underpinning protocol for its communications between library systems and content vendors. This is a different application for the LCF standard as it is used for two-way communications between the library system and content provider systems rather than the more traditional client-server communications in the terminal applications LCF was original designed for.

One of the attractions of LCF to the FASTEN committee was the versatility of its underlying REST model which meant it could be used in new scenarios. Some of the ideas for using LCF in these new scenarios may be applicable to extending LCF to support inter-library loans. In addition, the mechanisms for sending message and alerts to patrons in LCF were influenced by FASTEN's requirements.

The project website has a wealth of information about this exciting project:
https://www.niso.org/standards-committees/fasten