

Migration from Millennium & Symphony to KOHA: Khalifa University experience

Nikesh Narayanan¹ Walter Brian Hall² Shaju Badarudeen³

1. Asst. Professor and IT librarian, Zayed University, Dubai.
Nikesh.narayanan@zu.ac.ae
2. Walter Brian Hall, Public Services Librarian, Khalifa University, Abu Dhabi.
Walter.hall@ku.ac.ae
3. Shaju Badarudeen, Senior System Administrator, Khalifa University, Abu Dhabi
Shaju.badarudeen@ku.ac.ae

Abstract

This paper provides the details of the implementation and challenges of the ILS migration project of Khalifa University University, Abu Dhabi. The University is established by merging the three prominent higher education institutions in Abu Dhabi, namely the Khalifa University of Science, Technology and Research (KUSTAR), the Petroleum Institute (PI), and the Masdar Institute of Science and Technology (MI). Before the merger, KUSTAR library used Innovative Interfaces' Millennium ILS, PI used Sirsi-Dynix's Symphony, and MI used the open-source Koha system (with the support of a service provider). After the merger, the new University decided to choose the KOHA ILS system for the campus libraries. This paper examines the objectives, risk factors, migration procedures, implementation tasks, customization, and challenges of ILS migration of Khalifa University.

1. Introduction

Khalifa University (KU) is established by merging the three prominent higher education institutions in Abu Dhabi, namely the Khalifa University of Science, Technology and Research (KUSTAR), the Petroleum Institute (PI), and the Masdar Institute of Science and Technology (MI). Before the merger, KUSTAR library used Innovative Interfaces' Millennium ILS, PI used Sirsi-Dynix's Symphony, and MI used the open-source Koha system (with the support of a service provider). After the merger, the new University decided to choose a single ILS system for the campus libraries. A comparison of Millennium, Symphony, and Koha was made by the systems team to examine the functional capabilities against the requirements of the library. Various parameters were identified for the comparisons, and the KOHA was selected. This article examines the objectives, implementation strategy and the challenges of migration from Millennium and Symphony to Koha

2. Key Objectives

Once the choice of ILS had been made, the goal of the library and IT team was to implement Koha with all the existing functionalities that the existing Millennium and Symphony installations had. Additionally, exploring other functionalities and features of Koha were secondary objectives. The team had the following specific goals:

- Implement Koha with all needed settings and required installations within 3 months time

- Integrate SIP2 for use with existing self-check machines.
- Integrate LDAP for single sign-on (SSO) to provide a seamless user experience.
- Integrate acquisition through EDIFACT.

3. Comparison and selection of software

The ILS comparison was made under two broad categories. The first one was centered on end-user features. The second was back-end staff operations. After analyzing the parameters of the chosen options, the team was confident about the capabilities of Koha, and a recommendation was made in favor of it to replace the existing Millennium and Symphony ILS with the following justifications:

- All of the essential functionalities required for the library are available in Koha.
- Because Koha is open-source, it provides easy access to modifications that can be made at the local level.
- Koha is free from dependence on vendor lock-ins.
- Libraries do not have to wait for service from commercial ILS providers help desk, which has frequently been slow.
- No license restrictions on the number of users, number of items, etc.
- Koha has an active user group that freely shares resources and information.
- Koha is rapidly developing, with two free annual upgrades.
- Easy to switch over to other systems in the future, since all data is open and can be retrieved.
- Expert existing library and IT staff to manage the new system.
- Koha's reporting system is more flexible and powerful than the previous systems.
- No need to pay any subscriptions or annual maintenance contracts, which saves money.

It was decided to undertake the Koha implementation project by a team consisting of IT and library personnel. The implementation responsibilities were broadly divided as follows.

- IT department provided all hardware.
- IT installed all the required software, which can be accessible securely by all stakeholders.
- IT would provide a staging area to test changes to the system (e.g., changes to stylesheets or other display code, software upgrades, and patches, etc.).
- IT insured the necessary infrastructure (like network connectivity, unified authentication mechanism, etc.) to support system features
- The library team would be responsible for all internal system set up of Koha and data migration.
- The library and IT team would work together for other integrations (e.g., SIP2, EDIFACT, Z39.50 etc.).

4. Risk factors

As there was no service provider support, the project success was solely based on the expertise of the library and IT staff. The project was also the team's first experience of ILS data migration on this scale (nearly 200,000 records). The biggest challenge was that the three systems store their item records in different sections within the MARC format, so the records from the non-Koha systems were changed with the MarcEdit program and some custom scripts written by the library team.

After the MARC item records were converted, the team began with a test instance of Koha installed on a desktop computer and a limited setup records (i.e., all patron records and 10,000 bibliographic and item records).

This test went exceptionally well, and the confidence gained through this successful implementation of the test system was encouraging and remarkable. The team then worked to perform full export of all data on the staging server and was successful. Regarding the future of Koha, all the developments are dependent on community developers. But considering the past experience of other users, Koha has been performing superbly in updating and bug fixing.

5. Implementation

The implementation schedule was prepared, and a time frame was set for each task. The tasks were broadly divided as follows

- Installation of KOHA on a Linux server with all dependency software (Apache, PERL Modules, MySQL Database, etc.)
- Defining the basic parameters of Koha
- Data migration
- Customization of Koha based on library requirements
- Integration with external systems
- Testing and debugging
- Live Koha

5.1. Installation of Koha and dependency software

The Environment: The application and database servers are virtual servers created in VMWare infrastructure. The servers are scalable and running on high availability mode.

Server Operating System: It was decided to use 64-bit Ubuntu 16.04.4 LTS (Xenial Xerus), including the Mitaka release of OpenStack, with support guaranteed until April 2021. The server was updated with the latest security patches and upgrades.

Database Servers: The newest version (5.7.12) of MySQL Server was installed. Necessary tweaking has been done to disable the strict mode of MySQL.

Web Servers: Newest version of Apache HTTP Server (2.4.18) was installed.

KOHA: The latest version of Koha 18.05 was downloaded and installed.

5.2. Defining basic parameters of Koha

Defining the basic parameters and authorized values in Koha is one of the essential processes that had to be completed before the migration process. The unified library's patron and circulation policies, locations, collections codes, item types, etc. are all dependent on these values. In our case, all these values were different in Millennium and Symphony. We had to define either new values or map the values of Millennium and Symphony systems onto Koha. The following basic parameters are mandatory before data migration.

Libraries and groups: The library code is mandatory in the 952 marc tag of Koha for item circulation: 952\$a is the branch code, and 952\$b is the holding library.

Item types: The item type is the essential parameter for circulation rules in Koha. In Millennium it is more flexible, and you can use a combination of item types and location codes for defining circulation rules. In our library, the circulation rule had been different for different collections. So we had to define many item types to follow the same circulation rules that were set in Millennium and Symphony. The item type information is mandatory in 952\$y field of Koha.

Other authorized values: It is important to define Location (LOC), and Collection (CCODE) authorized values before migration. LOC determines the shelving location, stored in the 952\$c tag. CCODE denotes collection code and is stored in the 952\$8 tag for data migration.

5.3. Data Migration

We had set up a data migration policy to decide the data that are important for the new system. We had set up a priority order of the data to be transferred to KOHA. The following is our data priority table.

Data	Priority
MARC data- Bibliographic records with item details for print books	#1
Patron records with circulation data	#1
MARC data for ebooks	#2
Order records	#3

5.3.1. MARC data

Tools to export the raw MARC data were available in both Millennium and Symphony; hence it was easy to export all MARC data from these systems, including the item level information. We did data cleaning of this exported data before importing to Koha. For example, lost books and withdrawn items were removed from the MARC record file prepared for Koha import. As all the systems use the MARC 21 standard, most of the MARC tags were able to migrate without any modification. One of the major mapping requirements was needed in 942\$c field of Koha, where we added the Biblio type of the document. This is especially important for the eBooks category, which doesn't have any 'item' associated with it. So the only way to get a

result list facet is through the 942 tag. For example, we set the 942 \$c subfield to get an 'Ebook' facet in the search result page.

Another significant mapping and editing task was within the item records. In Koha, item-level details are stored in 952 field. Millennium uses the 945 tag and Symphony the 949 tag to hold item-level data. Hence it was required to map the item level tags of Millennium and Symphony with 952 marc tag of Koha. Some item data elements which are essential to Koha were not included in MARC data of Millennium and Symphony. So we had to add those data elements in each item record before importing to Koha.

Mapping logic is given in the below table.

KOHA item level Tag/subfield	Data element	Corresponding Millennium Tag/subfield	Corresponding Symphony Tag/subfield	Data newly added
952\$2	Source of classification or shelving scheme		949\$w	lcc –the code for Library of Congress, defined in "classification sources" of cataloging admin
952\$8	Collection code			The values defined in the authorized value field for collection code (CCODE)
952\$a	branchcode	No equivalent subfield in 945 tag of Millennium. The value was taken from biblio location of Millennium. 907\$b	949\$m	The values defined in the basic parameter "Libraries and Groups."
952\$b	branchcode	No equivalent subfield in 945 tag of Millennium. The value was taken from biblio location of Millennium. 907\$b	949\$k	The values defined in the basic parameter "Libraries and Groups."

952\$c	Shelving location code	945\$l	945\$l	The values defined in the authorized value field of KOHA for location (LOC)
952\$g	Purchase price	945\$g		
952\$l	Total Checkouts	945\$u	949\$n	
952\$o	Koha full call number	945\$ab	949\$a	
952\$p	<i>barcode *</i>	<i>945\$i</i>	<i>949\$i</i>	
952\$v	Replacement price		949\$p	We provided a fixed price for all the books
952\$y	<i>Koha item type *</i>	<i>945\$t</i>	949\$t	The values defined in the basic parameter "Item Types." We created additional item types in KOHA.

We used two tools to manipulate the data from the existing MARC records. MarcEdit, a powerful open-source suite of tools for viewing and editing MARC data, was used in mapping MARC fields and also adding additional tags and subfields in MARC files. The library staff also wrote some custom PHP scripts that performed some of the conversions as well.

5.3.2. Patron data

We decided to migrate only active patrons and non-active patrons with checked-out materials. We followed two procedures for patron upload.

5.3.2.1. Patron with checked out items

The following procedures were followed for patrons with at least one checked out item.

- Created a patron reports in Millennium and Symphony for the details of patrons with currently check-out items.
- These patron details were mapped to a tab-delimited format for uploading to Koha. Circulation data were not included at this stage

5.3.2.2. Uploading circulation data

- Circulation data were loaded using the offline circulation feature of Koha
- Since our library doesn't charge for overdue items, fines were not a factor.

5.3.2.3. Other active patrons without any checked-out item.

A list of active employees and students was created from the university Banner system and uploaded to Koha to create a patron list. All patrons' details were overlaid for the previously uploaded patrons (i.e., those with at least one checked item), and new patron records were created for other active patrons.

5.4. Koha Customization

We prepared a document for our customization needs and checked it against the customization options in the Koha global systems preferences, implementing it wherever possible. Customizations were also done in the MARC bibliographic framework, record matching rules, OAI set configuration, setting up of preferred classification source, currencies and exchange rates, budgets, funds, EDI account set up, and setting up of Z39.50 servers, etc. Koha implementation checklists are available at the Koha Community wiki: <http://manual.koha-community.org/3.6/en/implementation.html> (Koha Community, 2020b)

5.5. Koha integration with external systems

Koha can be integrated with external systems using various authentication methods. KU library decided to implement the following integrations in the first phase.

Integration with self-check kiosk: The library has five self-check stations. To connect Koha with the self-check stations, through the SIP2 protocol, two separate user accounts were created in Koha with the issue, return, and renewal permissions for each self-check station. Then the necessary details were added to the /etc/koha/SIPconfig.xml file, as per the instruction given in the Koha Wiki SIP2 server set up page. (Koha Community, 2020b) https://wiki.koha-community.org/wiki/Koha_SIP2_server_setup

LDAP to authenticate patrons: Khalifa University uses the Central Authentication Service (CAS) for authenticating users and providing single sign-on (SSO). The IT department used the existing KU CAS server to integrate Koha with CAS.

Integration with Discovery service: Khalifa University uses EBSCO Discovery Service (EDS). Two options are available to integrate Koha with EDS. The first option is importing all the Koha MARC data into EDS. The second option is to make use of EDS's API to make content discoverable via the Koha interface. Currently, we have adopted the first method. Koha MARC records are exported and uploaded to the EBSCO FTP site for integrating with EDS. We do this biweekly to have the updated Koha records in EDS. While currently, we are doing this importing manually, we plan to implement the OAI harvesting module of Koha so that EDS can harvest MARC records from Koha automatically. EDS provides a link back to Koha using custom links and also displays real-time availability of Koha items in the EDS interface through the Koha Z39.50 search.

EDI ordering — vendor integration for acquisition: Koha uses the EDIFACT (Electronic Data Interchange for Administration, Commerce, and Transport) standard for the electronic order process. This same method was used in Millennium also. EDI (Electronic Data Interchange) accounts and EAN were created for each vendor by gathering the necessary information following the procedures given in <http://manual.koha-community.org/16.05/en/ediprocess.html> (Koha Community, 2020a)

5.6. Challenges and issues

There were some challenges in data migration and other implementation tasks. The New York University Health Science Library's conversion project from Millennium to KOHA (Walls, 2011) helped us to identify some of the challenges before starting the project.

5.6.1. MySQL strict mode issues

We are using MySQL 5.7, and it didn't work well with Koha. We encountered the following errors:

- PRIMARY KEY NULL Errors during DB Creation
- sample_patrons.sql - ERROR 1048 (23000) at line 1: Column 'address' cannot be null
- authorities_normal_marc21.sql - ERROR 1292 (22007) at line 2: Incorrect DateTime value: '0000-00-00 00:00:00' for column 'timestamp' at row 1
- Error when edit borrowers "There are no patron categories defined," but categories exist

To solve these issues, we have disabled MySQL strict mode. The Koha wiki suggests replacing MySQL with MariaDB (an open-source fork of the MySQL project).

(https://wiki.koha-community.org/wiki/Koha_on_Debian) (Koha Community, 2020)

5.6.2. Determining Circulation rules

Millennium allows item types and item locations along with patron types to frame the loan rules. But Koha uses only the combination of patron categories and item types to determine loan rules. This difference necessitated completely reevaluating our item types, which hadn't previously affected loan rules. We redefined many item types to follow the same loan rules.

5.6.3. 'Create list' feature in Millennium

Millennium has a robust "Create Lists" feature for querying & exporting different types of records (patron, bibliographic, vendor, etc.). Koha doesn't have a similar function but can be queried using powerful SQL queries. However, many of our staff are not familiar with the SQL, so often the systems librarian needs to help other teams to generate various reports. These queries can be saved and used for future use.

Besides, the members of the KOHA community have created a report library (Koha community, 2020) (https://wiki.koha-community.org/wiki/SQL_Reports_Library) with more than 500 prewritten SQL queries. We can use some of those queries with slight modifications based on our requirements. Care must be taken when trying to use these prewritten reports, however, as many are designed for previous versions of Koha, and the database schema has changed significantly over the years.

5.6.4. Recall functionality

There is no recall functionality in Koha, so staff needs to send personal e-mails to patrons, requesting that they return items before the actual return date in the case when required.

5.6.5. Regional language searching

Arabic language searching does not work well in Koha by default. We followed the instruction provided on the Koha community page (https://wiki.koha-community.org/wiki/Correcting_Search_of_Arabic_records) and Arabic searching worked well after making the suggested changes. (Koha Community, 2020c)

5.7. Conclusion

KOHA implementation at Khalifa University is very successful and capable of managing all the expected functionalities of a modern ILS system. The current trend is moving from legacy ILS to cloud-based LSP, a unified resource management system with end-to-end management of print, electronic resources, and digital collections. Hope, KOHA, and other open-source communities will come up with competitive LSP products.

References:

- Koha community (2020) *SQL Reports Library - Koha Wiki*. Available at: https://wiki.koha-community.org/wiki/SQL_Reports_Library (Accessed: 19 June 2020).
- Koha Community (2020a) *10. EDI Process*. Available at: <http://manual.koha-community.org/16.05/en/ediprocess.html> (Accessed: 19 June 2020).
- Koha Community (2020b) *Chapter 13. Implementation Checklist*. Available at: <http://manual.koha-community.org/3.6/en/implementation.html> (Accessed: 19 June 2020).
- Koha Community (2020c) *Correcting Search of Arabic records - Koha Wiki*. Available at: https://wiki.koha-community.org/wiki/Correcting_Search_of_Arabic_records (Accessed: 19 June 2020).
- Koha Community (2020) *Koha on Debian - Koha Wiki*. Available at: https://wiki.koha-community.org/wiki/Koha_on_Debian (Accessed: 19 June 2020).
- Walls, I. (2011) 'Migrating from Innovative Interfaces' Millennium to Koha: The NYU Health Sciences Libraries' experiences', *OCLC Systems and Services*. Emerald Group Publishing Limited, 27(1), pp. 51–56. doi: 10.1108/10650751111106564.