**System Documentation for QESS Project (No. T01/QESS/2013)**

ILL Locate Self-developed Program for Aleph Technical Details

Dated: 15 Sep 2016

**Revision Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Author** | **Content** | **Section** |
| 1 Sep 2016 | William NG / Ching Ling CHAN | First version | - |
| 15 Sep 2016 | William NG | Change content | III.1.ii.3 & III.3.v.4 |

**General Remark:**

This document describes programming details and the setup of ILL Locate Self-developed Program for Aleph. This is the program serves as a middle man for Aleph ILL module to search remotely if an ILL supplier holds an available title for ILL request. The remote search action normally executes by ILL operator before the start of an ILL process.

1. **The Program Sources**
   1. The most updated version of the program sources can be found at <https://github.com/HKSFILS/Library-Record-Locate>.
2. **The Main Structure of the Programs**
   1. The program divides into two parts:
      1. Front-end disguised Aleph X-Service
      2. Back-end querying modules
   2. (i) is written in Jsp and works as a service for Aleph ILL Module ILL Locate function. The file:

<https://github.com/HKSFILS/Library-Record-Locate/blob/master/src/X>

* 1. (ii) are written in Java and work as a query agent to suppliers’ library systems. The files are under the folder:

<https://github.com/HKSFILS/Library-Record-Locate/tree/master/src/hk/edu/csids>

* 1. The structure of (ii) is recorded in its Javadoc:

<https://github.com/HKSFILS/Library-Record-Locate/tree/master/doc>

* 1. For querying ILL suppliers’ item availability, there are two optional ways:
     1. Query Primo X-Service first, then ILS’s Item Availability API. Achieved by:
        1. “PrimoQueryByISBN.java”: for query by ISBN. Ssource:

<https://github.com/HKSFILS/Library-Record-Locate/blob/master/src/hk/edu/csids/bookquery/PrimoQueryByISBN.java> and

* + - 1. PrimoQueryByNonISBN.java, source:

<https://github.com/HKSFILS/Library-Record-Locate/blob/master/src/hk/edu/csids/bookquery/PrimoQueryByNonISBN.java>

* + 1. Query Z39.50 in OPAC format. Achieved by:
       1. Z3950QueryByISBN.java, source:

<https://github.com/HKSFILS/Library-Record-Locate/blob/master/src/hk/edu/csids/bookquery/Z3950QueryByISBN.java> and

* + - 1. Z3950QueryByNonISBN.java, source:

<https://github.com/HKSFILS/Library-Record-Locate/blob/master/src/hk/edu/csids/bookquery/Z3950QueryByNonISBN.java>

* 1. For querying using Z39.50, the self-developed programs relies on independent tools; therefore, the following tools are needed to setup:
     1. Yaz4j: A Java library works as an interface to query the Z39.50 client.

(<http://www.indexdata.com/yaz4j>)

* + 1. Yaz: A C++ written library Z39.50 client, being called by Java via JNI.

(<http://www.indexdata.com/yaz>)

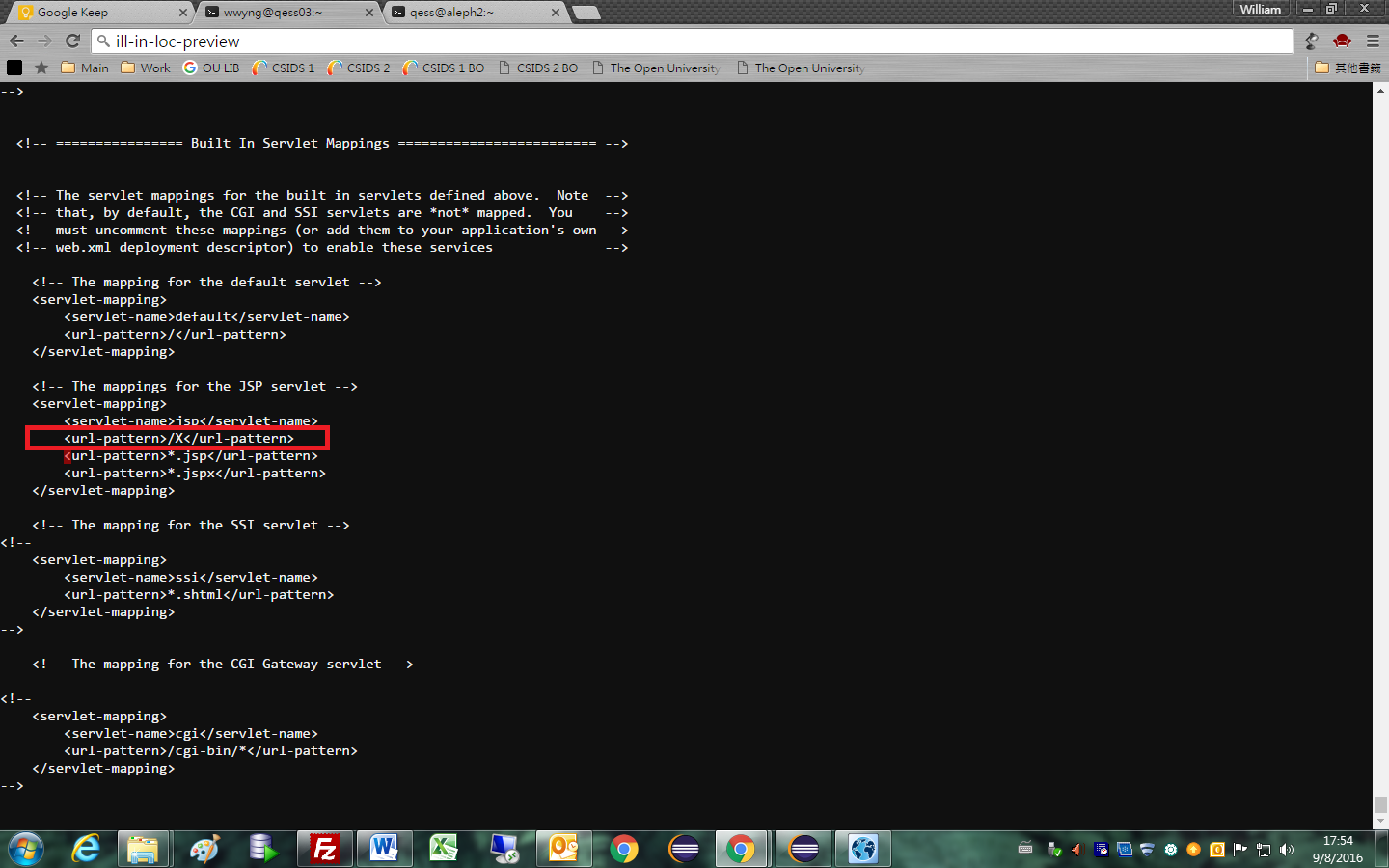
* 1. In order to use the Yaz libraries, which is written in C++ and is OS dependent, setups are needed, which will be described in the next section.
  2. Other third party Java libraries are also used, all obtainable from <https://github.com/HKSFILS/Library-Record-Locate/tree/master/jar>, includes
     1. ZHConverter.jar: for converting Traditional Chinese into Simplified Chinese. Obtainable from:

<https://github.com/ChanningBJ/java-zhconverter>

* + 1. commons-lang3-3.4.jar and commons-validator-1.5.1.jar: for misc string conversion and validation.. Obtainable from:

<http://commons.apache.org/>

1. **The Setup**
   1. Tomcat Server Setup
      1. A Tomcat server will host all the self-developed programs. The placing of the programs:
         1. The JSP “X” (Front-end Aleph Service in disguise): must be placed at the root of a Tomcat root application (i.e. webapps/ROOT/).
         2. In order for “X” to run as a Jsp, the Tomcat configuration file “[tomcat]/conf/web.xml” must be added a line, as:



* + - 1. The Java classes (Back-end query modules): must be placed at the Java class folder of the Tomcat root application (i.e. webapps/ROOT/WEB-INF/classes)
      2. Yaz4j.jar (an interface part of Z39.50 client) and other third parties Java libraries in Jar format: must be placed at the Java library folder of the Tomcat root application (i.e. webapps/ROOT/WEB-INF/lib)
  1. Z395.50 Client Library Setup
     1. The Z39.50 Client Library is OS dependent and must be installed first for making the self-developed programs work. The client library is obtainable from (<http://www.indexdata.com/yaz>)
     2. After installation, two files “libyaz4j.so” and “yaz4j.dll” will be obtainable from the “bin” subfolder under the Yaz program folder. (e.g. In Windows: “Program Files\YAZ\bin”)
     3. OS’s environment variables for loading library (Windows, DLL library; Linux, libxxx.so) must be added to include the path of “libyaz4j.so” or “yaz4j.dll” depending on the OS. (for example, under Linux context the commend:

“export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/usr/local/lib/”

Would need to be run where the “libyaz4j.so” is in “/usr/local/lib/”)

* 1. Supplier List Setup
     1. ILL suppliers’ library system’s addresses and other parameters can be configured under the package name space “hk.edu.csids.bookquery”. An sample of the configuration file:

<https://github.com/HKSFILS/Library-Record-Locate/blob/master/src/hk/edu/csids/bookquery/config.txt>

* + 1. In the configuration file setup take the form (a space is as a separator):
       1. [Configuration Code] [Value]
    2. General Configuration code setup:
       1. INST\_CODE: the institution code current query would be made to.
    3. Configuration codes setup for Primo-ILS query:
       1. SOURCE\_ID\_[institution code]: Primo record source id of an specified institute.
       2. LOCAL\_SCOPE\_[institution code]: Primo search scope of a specified institute.
       3. PRIMO\_BASE: Primo X-Service server address for query.
       4. ILS\_AVA\_BASE\_[institution code]: ILS availability API server address and the API path of a specific institute.
       5. PRIMO\_SEARCHFIELD\_PUBLISHER: Corresponding Primo field for publisher searching.
       6. PRIMO\_SEARCHFIELD\_EDITION: Corresponding Primo field for edition searching.
       7. PRIMO\_SEARCHFIELD\_AUTHOR: Corresponding Primo field for author searching.
    4. Configuration codes setup for Z39.50 query:
       1. Z3950\_PORT\_[institution code]: TPC port of Z39.50 query of a specific institute.
       2. Z3950\_BASE\_[institution code]: Z39.50 query database name of a specific institute.
       3. Z3950\_SERVER\_[institution code]: Z39.50 query server address of a specific institute.
       4. EACC\_SYSTEM\_[institution code]: Can be true or false, indicates if the Z39.50 query server ILS of a specific institute uses “EACC/CCCI” for encoding. The program will translate EACC/CCCII from and to Unicode if that is the case.