**Step 1: Building & packaging the SL Translit Engine as a jar**

1. Clone and open the maven project from the IDE (preferably Intellij)   
   <http://www.govforge.icta.lk/gf/project/translit_v2/>
2. Modify the parent pom.xml (\translit-master\pom.xml) with the following  
     
   (***Double click below xml to view the open document***)  
     
   The important changes include the below,  
     
   <**properties**>  
    <**project.build.sourceEncoding**>UTF-8</**project.build.sourceEncoding**>  
   </**properties**>

<**plugins**>  
 <**plugin**>  
 <**groupId**>org.apache.maven.plugins</**groupId**>  
 <**artifactId**>maven-surefire-plugin</**artifactId**>  
 <**version**>2.16</**version**>  
 <**configuration**>  
 <**argLine**>-Dfile.encoding=UTF-8</**argLine**>  
 </**configuration**>  
 </**plugin**>  
 <**plugin**>  
 <**groupId**>org.apache.maven.plugins</**groupId**>  
 <**artifactId**>maven-compiler-plugin</**artifactId**>  
 <**version**>3.8.0</**version**>  
 <**configuration**>  
 <**source**>11</**source**>  
 <**target**>11</**target**>  
 <**encoding**>UTF-8</**encoding**>  
 <**fork**>true</**fork**>  
 <**meminitial**>1024m</**meminitial**>  
 <**maxmem**>2024m</**maxmem**>  
 </**configuration**>  
 </**plugin**>  
 <**plugin**>  
 <**groupId**>org.apache.maven.plugins</**groupId**>  
 <**artifactId**>maven-jar-plugin</**artifactId**>  
 <**version**>2.1</**version**>  
 <**configuration**>  
 <**archive**>  
 <**manifestEntries**>  
 <**Extension-Name**>${project.groupId}</**Extension-Name**>  
 <**Specification-Title**>${project.artifactId}</**Specification-Title**>  
 <**Specification-Vendor**>Asankha Chamath Perera</**Specification-Vendor**>  
 <**Specification-Version**>${project.version}</**Specification-Version**>  
 <**Implementation-Title**>Translit</**Implementation-Title**>  
 <**Implementation-Vendor-Id**>com.asankha</**Implementation-Vendor-Id**>  
 <**Implementation-Vendor**>Asankha Chamath Perera</**Implementation-Vendor**>  
 <**Implementation-Version**>${project.version}</**Implementation-Version**>  
 <**Main-Class**>com.asankha.translit.Transliterate</**Main-Class**>  
 </**manifestEntries**>  
 </**archive**>  
 </**configuration**>  
 </**plugin**>  
</**plugins**>

Make sure to verify the highlighted markup above. Mainly the above xml file was modified to match the maven compiler and JDK versions of the Mosip’s Reg-Client 1.1.3 version.

1. Once the pom is modified accordingly, go ahead and do a maven install from the maven toolbar (or cli) to build and package the jar.  
     
     
   Graphical user interface, text, application, email

   Description automatically generated

**Step 2: Building & packaging the** *kernel-transliteration-translit-1.1.3*

**library as a jar**

1. Create a new maven project from

**Step 3: Integrating the Translit jar to the reg-client source code.**

The steps below were followed to add the above jar as a local dependency to reg-client code.

1. Add the jar (we built in Step 1) to the resources folder. *(\reg-client\registration\registration-client\src\main\resources\translit\_lib\translit-core-1.0.0.jar*)  
   Graphical user interface, text, application, email

   Description automatically generated
2. Modify the pom.xml (registration/registration-client/pom.xml) to add our Jar.  
   <**dependency**>  
    <**groupId**>translit-core-1.0.0</**groupId**>  
    <**artifactId**>translit-core-1.0.0</**artifactId**>  
    <**scope**>system</**scope**>  
    <**version**>1.0.0</**version**>  
    <**systemPath**>${project.basedir}/src/main/resources/translit\_lib/translit-core-1.0.0.jar</**systemPath**>  
   </**dependency**>  
   Graphical user interface, text, application

   Description automatically generated
3. Build the project to include our jar. And you may refer to the jar within the source code of the reg-client by simply adding this java import above any .java file.  
   **import** com.asankha.translit.Transliterate
4. Finally refer the code in branch 1.1.3-translit (<https://github.com/ICTASL/udi-poc-registration/tree/1.1.3-translit>) to understand how the above jar was referenced within the reg-client code.  
   (mainly the [DemographicDetailController.java](https://github.com/ICTASL/udi-poc-registration/blob/1.1.3-translit/registration/registration-client/src/main/java/io/mosip/registration/controller/reg/DemographicDetailController.java), [FXUtils.java](https://github.com/ICTASL/udi-poc-registration/blob/1.1.3-translit/registration/registration-client/src/main/java/io/mosip/registration/controller/FXUtils.java) and [TranslitUtil.java](https://github.com/ICTASL/udi-poc-registration/blob/1.1.3-translit/registration/registration-client/src/main/java/io/mosip/registration/util/common/TranslitUtil.java) classes)

**Step 3: Improving the accuracy of our Translit engine.**

1. The key to improve the accuracy of a certain transliteration between 2 languages lies within the below files. Mainly the rules-\*.txt files. Located in the cloned project in **Step 1. *(\translit-master\modules\core\src\main\resources\com\asankha\translit\resources)***UDI-POC team mainly changed the en-to-si.txt, rules-en.txt and the rules-si.txt files to suite our requirements for the POC stage. Please refer <https://github.com/ICTASL/UDI-poc/tree/language-translation-1.1/master_data_setup/language_translation>  
   For the changes done.
2. Now rebuild the project with **maven install** phase.
3. Finally add the newly built jar to the resources folder (*same as in* ***Step 3.1***) for the changes to reflect in the Reg-Client. *(\reg-client\registration\registration-client\src\main\resources\translit\_lib\translit-core-1.0.0.jar*)